

An Application of Number Theory to the Organization of Raster-Graphics Memory

BENNY CHOR, CHARLES E. LEISERSON, AND RONALD L. RIVEST

Massachusetts Institute of Technology, Cambridge, Massachusetts

AND

JAMES B. SHEARER

University of California at Berkeley, Berkeley, California

Abstract. A high-resolution raster-graphics display is usually combined with processing power and a memory organization that facilitates basic graphics operations. For many applications, including interactive text processing, the ability to quickly move or copy small rectangles of pixels is essential. This paper proposes a novel organization of raster-graphics memory that permits all small rectangles to be moved efficiently. The memory organization is based on a doubly periodic assignment of pixels to M memory chips according to a "Fibonacci" lattice. The memory organization guarantees that, if a rectilinearly oriented rectangle contains fewer than $M/\sqrt{5}$ pixels, then all pixels will reside in different memory chips and thus can be accessed simultaneously. Moreover, any M consecutive pixels, arranged either horizontally or vertically, can be accessed simultaneously.

We also define a continuous analog of the problem, which can be posed as: "What is the maximum density of a set of points in the plane such that no two points are contained in the interior of a rectilinearly oriented rectangle of unit area?" We show the existence of such a set with density $1/\sqrt{5}$, and prove this is optimal by giving a matching upper bound.

Categories and Subject Descriptors: B.3.2 [Memory Structures]: Design Styles—*interleaved memories*; I.3.1 [Computer Graphics]: Hardware Architecture—*raster display devices*

General Terms: Design, Performance, Theory

Additional Key Words and Phrases: BITBLT, Fibonacci lattices, golden ratio, interleaving, memory organization, raster graphics, rectangles

1. Introduction

The length of one memory cycle is a bound on how quickly a single pixel (picture element) of a raster-graphics display can be updated. If each pixel in a region is updated individually, the time to update the entire region is unacceptably large for many real-time or interactive environments. A natural way to avoid this problem

This research was supported in part by the Defense Advanced Research Projects Agency under Contract N00014-80-C-0622, by the National Science Foundation under Grant MCS 80-06938, and by the Miller Institute for Basic Research in Science. James Shearer did part of this research while consulting at Bell Laboratories, Murray Hill, N.J. An early version of this paper by Chor, Leiserson, and Rivest appeared in *Proceedings of the 23rd Annual Symposium on the Foundations of Computer Science* (IEEE, New York, 1982, pp. 92-99). © IEEE 1982. The portions of this paper that appeared in the early version are reprinted with permission.

Authors' addresses: B. Chor, C. E. Leiserson, and R. L. Rivest, Laboratory for Computer Science, Massachusetts Institute of Technology, 545 Technology Square, Cambridge, MA 02139; J. B. Shearer, IBM Research, P.O. Box 218, Yorktown Heights, NY 10598.

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.

© 1986 ACM 0004-5411/86/0100-0086 \$00.75

is to access more than a single pixel at a time. Since the memory is typically partitioned among M random-access memory chips, as many as M pixels can be accessed simultaneously, provided that no two pixels reside in the same memory chip.

Figure 1 illustrates a common organization of raster-graphics memory. Each pixel on the screen is assigned to one of M memory chips in row-major order. Thus, in every row, the pixels in column m , $M + m$, $2M + m$, and so forth are stored in the same memory chip m . This organization made a good deal of sense when raster-graphics displays were new and the interface between the raster memory and the cathode-ray tube (CRT) was considered complicated. When the screen is refreshed from memory, the line-by-line horizontal scan accesses M pixels in a row and converts them into an analog video signal. But although the memory system achieves maximal parallelism for the screen refresh operation, it can be remarkably inefficient for other operations. Updating a vertical line of pixels, for example, requires a separate memory access for each pixel.

For arbitrary patterns of access there is no hope of maximal parallelism, since, whatever the organization, an adversary can choose to access all the bits in a single memory chip. The best we can hope for is to achieve high concurrency for a limited set of frequently used operations. And today, since hardware support for screen refresh is relatively well understood, attention focuses on those operations that make the graphics system easier to program.

Many raster-graphics applications rely on the copying or moving of a rectangle of pixels as a basic operation, which is demonstrated by the fact that this operation is implemented in the microcode of most graphics processors. The ability to move small rectangles quickly is especially important in text-oriented applications.

Recently, displays have been developed [3, 5, 9] that are designed to move small squares quickly. Figure 2 shows how pixels are assigned to memory chips in the case of $M = 16$ memory chips. The screen is tiled with \sqrt{M} -by- \sqrt{M} squares, each of which contains a pixel assigned to a different memory. The attraction of this scheme is that any \sqrt{M} -by- \sqrt{M} rectilinearly oriented square, whether aligned on tile boundaries or not, contains pixels assigned to different memories. Thus any square of area M can be accessed in one memory cycle.

Unfortunately, the efficiency of the raster-scan operation is reduced in this scheme compared with the one of Figure 1. The line-by-line scan will only be able to access \sqrt{M} pixels in parallel because every $\sqrt{M} + 1$ -by-one horizontal rectangle contains two pixels in the same memory chip. A possible solution to this problem is to stagger the tiles so that the second column of tiles is shifted vertically by one raster, the third by two rasters, and so on. This ad hoc solution allows simultaneous access of all pixels in any M -by-one rectangle as well as simultaneous access of all pixels in any \sqrt{M} -by- \sqrt{M} square, but it suffers from asymmetry of horizontal and vertical dimensions and introduces a variety of other complications.

This paper asks the question, "How many memory chips M are required to guarantee that all pixels can be accessed simultaneously in an arbitrary rectilinearly oriented rectangle of N pixels?" A naïve organization requires $M = N^2$ memory chips, but we can do much better.

This paper uses techniques from number theory to produce a novel memory organization of $M \approx \sqrt{5}N$ chips that allows all pixels in any rectangle of area N to be simultaneously accessed. The scheme is regular—a doubly periodic function in the plane—and the constant $\sqrt{5}$ is approached from below, so that for small values of N , the constant is less than 2. Furthermore, for the frequently used operation of accessing a horizontal or vertical line, our scheme allows simultaneous access of all M memory chips.

```

1 2 3 4 . . . M 1 2 3 4 . . . M 1 2 3 4 . . . M
1 2 3 4 . . . M 1 2 3 4 . . . M 1 2 3 4 . . . M
1 2 3 4 . . . M 1 2 3 4 . . . M 1 2 3 4 . . . M
1 2 3 4 . . . M 1 2 3 4 . . . M 1 2 3 4 . . . M
1 2 3 4 . . . M 1 2 3 4 . . . M 1 2 3 4 . . . M
1 2 3 4 . . . M 1 2 3 4 . . . M 1 2 3 4 . . . M
1 2 3 4 . . . M 1 2 3 4 . . . M 1 2 3 4 . . . M
1 2 3 4 . . . M 1 2 3 4 . . . M 1 2 3 4 . . . M

```

FIG. 1. A common organization for raster-graphics memory, which is efficient for raster scan, but inefficient for vertical updates.

1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
5	6	7	8	5	6	7	8	5	6	7	8	5	6	7	8
9	10	11	12	9	10	11	12	9	10	11	12	9	10	11	12
13	14	15	16	13	14	15	16	13	14	15	16	13	14	15	16
1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
5	6	7	8	5	6	7	8	5	6	7	8	5	6	7	8
9	10	11	12	9	10	11	12	9	10	11	12	9	10	11	12
13	14	15	16	13	14	15	16	13	14	15	16	13	14	15	16

FIG. 2. The 4-by-4 organization for raster-graphics memory. Every 4-by-4 square contains pixels from distinct memory chips.

The remainder of this paper is organized as follows. Section 2 discusses a continuous model of the problem that prompted our (discrete) solution. Section 3 presents the doubly periodic “Fibonacci” organization of graphics memory, and Section 4 provides the number theoretic analysis necessary to prove that the scheme works. The optimality of the Fibonacci organization is proved in Section 5, and Section 6 discusses the addressing mechanisms needed to make the scheme work in practice. Section 7 contains some concluding remarks.

2. A Continuous Analog

In this section we introduce a continuous analog to the discrete problem. We define a set of *compatible points* as a set of points in the plane such that no two points in the set are contained in the interior of a rectilinearly oriented rectangle of unit area. The question we ask in this section is: “What is the maximum density of a set of compatible points?” We construct a set of compatible points whose density is $1/\sqrt{5}$, and we prove that this density is maximum.

The correspondence between this problem and the discrete problem introduced in the previous section is as follows. First, the continuous problem deals with rectangles of unit area. A closer correspondence to the discrete problem uses rectangles of area N . The set of compatible points then corresponds, in the discrete problem, to the set of pixels that reside in the same memory chip, and the density of points corresponds to the reciprocal of the number M of memory chips. The principal difference in formulation is that, in the continuous model, we no longer require that the “pixels” fall on grid points.

The statement in the continuous problem that the rectangles have unit area instead of area N , however, results in no loss of generality. Any set of points such that no two are contained in a rectangle of area N can be mapped to a set of compatible points by shrinking the coordinates of each point by a factor of \sqrt{N} . Observe, however, that this linear transformation does not work for the discrete case in which all points must have integer coordinates.

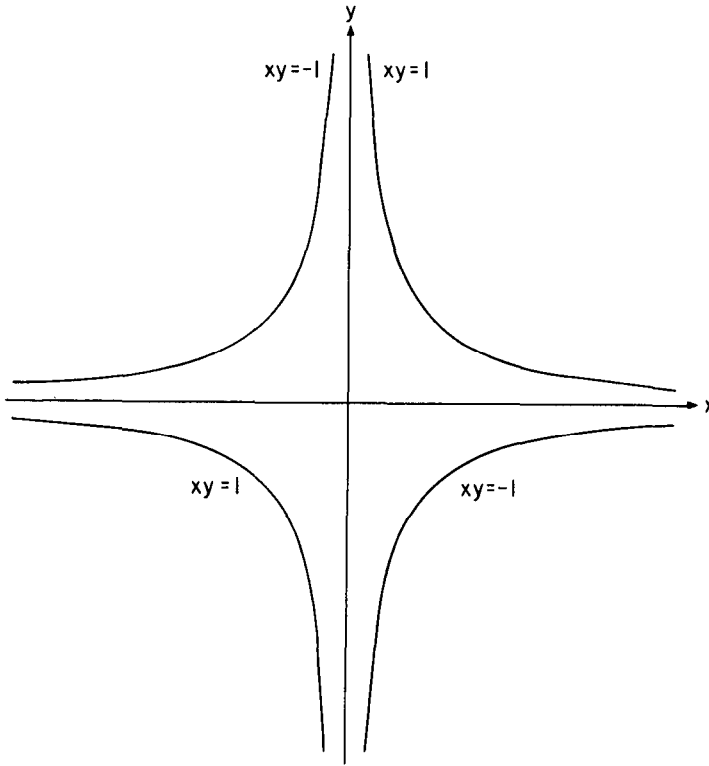


FIG. 3. The forbidden region around a point at the origin. If the origin is in a compatible set, then all the other points in the compatible set must fall outside the region defined by the hyperbolas.

We find it convenient to adopt some standard terminology from geometry of numbers. A *lattice* is a set of points that can be expressed as an integral, linear combination of linearly independent (over \mathbf{R}) *basis vectors*. If there are only two basis vectors, we define the parallelogram with the two basis vectors as sides of the *basic region* of the lattice. The *fundamental lattice* is the lattice generated by the basis vectors $(0, 1)$ and $(1, 0)$, and we call its points *grid points*. Many properties of lattices can be found in [6].

We formally define a set S of points in \mathbf{R}^2 as being a set of *compatible* points if for any pair of points (x_1, y_1) and (x_2, y_2) in the set, we have

$$|(x_1 - x_2)(y_1 - y_2)| \geq 1.$$

Around every point P drawn from a set S of compatible points, there is an infinite-area *forbidden region* bounded by two hyperbolas inside which no other point of S may lie. Figure 3 shows the forbidden region for a point at the origin. The points in that forbidden region satisfy $|xy| < 1$.

In the discrete model, the problem is to minimize the number M of memory chips required to allow simultaneous access of any rectangle of N pixels. For an arbitrary scheme of assigning pixels to memory chips, in a square region of A pixels, there will be some memory chip with the largest number k of pixels in the area. Therefore, the number of memory chips M is at least A/k , or $1/d$ where d is the maximum *density* of pixels from a single memory chip in the square region.

The analog to minimizing the number of memory chips is, in the continuous model, to maximize the density of points in a set of compatible points. Formally, we define the *density* of an arbitrary set of points S as

$$d(S) = \limsup_{r \rightarrow \infty} \frac{|\{p \mid p \in S \cap \mathcal{D}_r\}|}{\pi r^2},$$

where \mathcal{D}_r is a disk centered at the origin with radius r .

We construct a set S of compatible points in the plane whose density is $1/\sqrt{5}$, and then demonstrate the optimality of the construction by proving a matching upper bound on the density of compatible sets.

THEOREM 1. *The lattice that is generated by the basis vectors $(\sqrt{1/\phi}, \sqrt{\phi})$ and $(-\sqrt{\phi}, \sqrt{1/\phi})$ form a compatible set whose density is $1/\sqrt{5}$, where $\phi = \frac{1}{2}(1 + \sqrt{5})$ is the golden ratio.*

PROOF. For simplicity, denote $(\sqrt{1/\phi}, \sqrt{\phi})$ by (a, b) . The lattice points are compatible if and only if for all integers u and v , the lattice point $v(a, b) + u(-b, a) = (av - bu, bv + au)$ is outside the forbidden region around the origin (since the lattice is invariant under translations by its basis vectors). Equivalently, for all pairs $(u, v) \neq (0, 0)$, we must have

$$|(av - bu)(bv + au)| \geq 1.$$

We can rewrite the product as

$$\begin{aligned} (av - bu)(bv + au) &= abv^2 + (a^2 - b^2)uv - abu^2 \\ &= v^2 - \left(\phi - \frac{1}{\phi}\right)uv - u^2 \\ &= v^2 - uv - u^2. \end{aligned}$$

Since the Diophantine equation $v^2 - uv - u^2 = 0$ has no solution except $u = v = 0$, it follows that

$$|(av - bu)(bv + au)| = |v^2 - uv - u^2| \geq 1,$$

and thus the lattice points are indeed compatible.

The basic region of the lattice is square, and its area is $a^2 + b^2 = \phi + 1/\phi$, which is $\sqrt{5}$. Since there is a one-to-one correspondence between lattice points and lattice squares, the density is $1/\sqrt{5}$. \square

This lattice is not the only one that achieves a density of $1/\sqrt{5}$. In fact, Tom Leighton has observed that there are an infinite number of lattices of compatible points that have this density. For any t the lattice generated by the basis vectors

$$\left(t, \frac{1}{t}\right) \quad \text{and} \quad \left(\frac{3 + \sqrt{5}}{2}t, \frac{3 - \sqrt{5}}{2t}\right)$$

also achieves the bound. The lattice of Theorem 1 is a member of this family of lattices (choose $t = \sqrt{1/\phi}$), although the basis vectors given in the theorem are different. The advantage of the basis vectors defined in the theorem is that they define a basic region that is square, which, as we shall see in Section 4, simplifies somewhat the analysis of the discrete solution.

3. A Fibonacci Lattice Organization of Raster-Graphics Memory

This section describes an organization of raster-graphics memory using integer approximations of the lattice scheme from Theorem 1. The lattices are defined in

terms of Fibonacci numbers, which are described by the recurrence

$$\begin{aligned} F_0 &= 0, \\ F_1 &= 1, \\ F_r &= F_{r-1} + F_{r-2}, \quad r \geq 2. \end{aligned}$$

The memory organization has the property that all pixels in a rectilinearly oriented rectangle can be accessed simultaneously as long as the rectangle contains no more than N pixels. The number M of memory chips required is at most $\sqrt{5}N$, but for many practical values it is less than $2N$.

The real-world problem differs from the continuous analog given in Section 2 in that the locations of pixels must have integer coordinates. This subtle constraint causes the problem to change in two ways. First, the actual bounds are better for the discrete case than for the continuous case, although asymptotically they are the same. Second, the proofs become more involved.

Not surprisingly, the raster-graphics organization is similar to the scheme in Theorem 1, which suggests two basis vectors be used to generate the locations of all pixels within the same chip of the raster-graphics memory. Pixels are assigned to chips as follows. Let a and b be two relatively prime, nonnegative integers that will be specified precisely later. The two orthogonal vectors (a, b) and $(-b, a)$ determine a lattice in the plane, consisting of all points of the form

$$v(a, b) + u(-b, a),$$

where u and v are integers. Except for the corners, no other grid point lies on the boundary of the basic region because a and b are relatively prime. By including exactly one of the four corner points in the basic region, the region can be used to tile the entire plane. Thus the number of grid points in the basic region equals its area $a^2 + b^2$. Each of the grid points in the basic region is mapped into one of $M = a^2 + b^2$ distinct memory chips. The grid points in the plane are partitioned into M equivalence classes. Each equivalence class corresponds to a translation of the lattice. All points in the same equivalence class are assigned to the same memory chip. Since each equivalence class has a unique representative in the basic region, M memory chips are used.

In the next section, we show that the choice of successive Fibonacci numbers $a = F_r$ and $b = F_{r+1}$, which yields the number of memories $M = F_{2r+1}$, guarantees that every rectilinearly oriented rectangle containing no more than $M/\sqrt{5}$ pixels can be accessed simultaneously. Figure 4 illustrates this "Fibonacci lattice" organization for 13 memory chips ($a = 2, b = 3$). Here, the situation is even better than we promised—any rectangle with at most 11 pixels contains no 2 pixels from the same memory chip.

Furthermore, observe in the figure that 1-by-13 and 13-by-1 rectangles have no conflicts, which is the best one can do with $M = 13$ memory chips. This circumstance is not mere luck.

LEMMA 2. *Let a and b be relatively prime integers, and let $M = a^2 + b^2$. The doubly periodic memory organization with M memory chips that is based on the lattice generated by basis vectors (a, b) and $(-b, a)$ has the property that any 1-by- M or M -by-1 rectilinearly oriented rectangle contains no two pixels from the same chip.*

PROOF. Since the organization is doubly periodic, we can consider a horizontal or vertical line that starts at the origin and determine the next lattice point that falls on the line. If the line is vertical, all pixels on it have x -coordinate zero. The

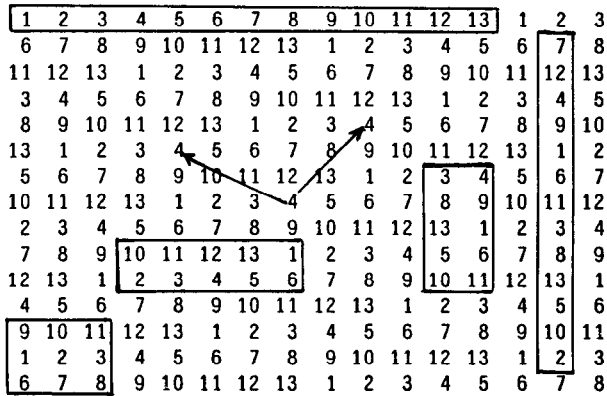


FIG. 4. The Fibonacci lattice organization for $M = 13$ memory chips. Every rectilinearly oriented rectangle having no more than $N = 11$ pixels has the property that all pixels are from distinct memory chips.

general form of lattice points is $v(a, b) + u(-b, a) = (av - bu, bv + au)$, and thus all lattice points on the line will have $av - bu = 0$. It follows that a divides bu , but since a and b are relatively prime, we can conclude that a divides u , and similarly, b divides v . Furthermore, u and v necessarily have the same sign, which means that the magnitude $|bv + au|$ of the y -coordinate is $|bv| + |au|$. Since a divides u , we have $|u| \geq a$, and by the same reasoning, $|v| \geq b$. Therefore, $|bv| + |au| \geq b^2 + a^2 = M$, and the magnitude of any lattice point on the vertical line is at least M . Thus any 1-by- M rectangle cannot contain two pixels from the same chip. Horizontal lines are treated the same way. \square

The following table describes the actual values we get for M and N in the Fibonacci lattice organization.

M	5	13	34	89	233	610
N	5	11	23	53	125	307

Notice that for all these values, the size N of rectangles that are guaranteed to have no conflicts is, in fact, larger than $M/2$. Thus for practical values of M , the overhead in allowing fast access to arbitrarily shaped rectangles of pixels is small.

4. Mathematical Analysis

In this section we analyze the properties of the Fibonacci lattice organization described in Section 3. The basis vectors for the raster-graphics memory organization are (F_r, F_{r+1}) and $(-F_{r+1}, F_r)$, where F_r is the r th Fibonacci number. (When we do not rely on the Fibonacci properties basis-vector components, we denote the basis vectors by (a, b) and $(-b, a)$.) We show in this section that the number of memory chips M in the organization is approximately $\sqrt{5}$ times the size N of the maximum-size rectangle guaranteed to have no conflicts.

The approach in this section is to find the minimum size MIN of a rectilinearly oriented rectangle containing two distinct lattice points. The *size* of a rectangle is defined to be the number of pixels in the rectangle. Notice that this definition of size differs from the continuous model since the number of pixels in a rectangle determined by two grid points equals its area plus half its perimeter plus one. Since

MIN is the minimum size of any rectilinearly oriented rectangle containing two lattice points, $N = \text{MIN} - 1$ because no rectangle of size strictly less than MIN contains two lattice points.

To find MIN, notice that, since the lattice is invariant under translations by its basis vectors, we lose no generality if, instead of discussing all pairs of lattice points, we restrict ourselves to those pairs one of whose elements is the origin. Furthermore, since we are interested in the minimal size, it suffices to consider only those rectangles that have the two lattice points at opposite corners. The first lattice point is the origin and the second lattice point has the form $v(a, b) + u(-b, a)$, and hence the size of the rectangle, its area plus half its perimeter plus one, is

$$S(u, v) = (|au + bv| + 1)(|-bu + av| + 1).$$

The value MIN is the minimum of $S(u, v)$ over all integers u and v not both 0. In order to find MIN, we first translate $S(u, v)$ into a simpler form.

LEMMA 3. *Let*

$$S(u, v) = (|F_r u + F_{r+1} v| + 1)(|-F_{r+1} u + F_r v| + 1),$$

and let $\hat{S}(u, v) = (|F_{2r} u - F_{2r+1} v| + 1)(|u| + 1)$. Then

$$\begin{aligned} \text{MIN} &\stackrel{\text{def}}{=} \min_{(u,v) \neq (0,0)} S(u, v) \\ &= \min_{(u,v) \neq (0,0)} \hat{S}(u, v). \end{aligned}$$

PROOF. We show that the range of S is the same as the range of \hat{S} by using an intermediate form B . For simplicity, we shall use the notation $a = F_r$ and $b = F_{r+1}$ introduced above.

Define the intermediate form

$$B(u, v) = S(ku - bv, -lu + av),$$

where k and l are integers such that $ak - bl = 1$. (The integers k and l exist because the greatest common divisor of a and b is 1.) The linear transformation given by

$$\begin{pmatrix} u \\ v \end{pmatrix} \rightarrow \begin{pmatrix} k & -b \\ -l & a \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix}$$

is a bijection since the determinant of the matrix is 1. Thus as (u, v) ranges over \mathbf{Z}^2 , the ordered pair $(ku - bv, -lu + av)$ also takes on all values in \mathbf{Z}^2 , and hence the range of S is the same as the range of B . Since the linear transformation is a bijection which maps $(0, 0)$ to $(0, 0)$, we have

$$\min_{(u,v) \neq (0,0)} S(u, v) = \min_{(u,v) \neq (0,0)} B(u, v).$$

If we expand $B(u, v)$, we get

$$\begin{aligned} B(u, v) &= S(ku - bv, -lu + av) \\ &= (|a(ku - bv) + b(-lu + av)| + 1)(|-b(ku - bv) + a(-lu + av)| + 1) \\ &= (|u| + 1)(|(a^2 + b^2)v - (bk + al)u| + 1), \end{aligned}$$

which has the form $(|u| + 1)(|Mv - Cu| + 1)$. (Note that $M = a^2 + b^2$ is the number of memory chips.)

In order to obtain $\hat{S}(u, v)$, we first determine the explicit coefficients M and C in $B(u, v)$ when the components of the basis vectors are the Fibonacci numbers

$a = F_r$ and $b = F_{r+1}$. We use the following two Fibonacci identities:

$$F_{i+j} = F_j F_{i+1} + F_{j-1} F_i, \quad F_{i+1} F_{i-1} - F_i^2 = (-1)^i.$$

From the first identity, we get that the number of memories M is

$$\begin{aligned} M &= a^2 + b^2 \\ &= F_r^2 + F_{r+1}^2 \\ &= F_{2r+1}. \end{aligned}$$

To find C , observe that, by the second identity, the k and l such that $ak - bl = 1$ are $k = (-1)^{r+1} F_r$ and $l = (-1)^{r+1} F_{r-1}$. Hence, by using the first identity, we have that

$$\begin{aligned} C &= bk + al \\ &= (-1)^{r+1} (F_{r+1} F_r + F_r F_{r-1}) \\ &= (-1)^{r+1} F_{r+r} \\ &= (-1)^{r+1} F_{2r}. \end{aligned}$$

Thus for $a = F_r$ and $b = F_{r+1}$, we have

$$B(u, v) = (|u| + 1)(|(-1)^r F_{2r} u + F_{2r+1} v| + 1).$$

The form \hat{S} was defined in the statement of the lemma as

$$\hat{S}(u, v) = (|u| + 1)(|F_{2r} u - F_{2r+1} v| + 1).$$

If r is odd, then $(-1)^r = -1$, and therefore $B(u, v) = \hat{S}(u, v)$. If r is even, on the other hand, then $B(u, -v) = \hat{S}(u, v)$. Since we have already shown that

$$\min_{(u,v) \neq (0,0)} S(u, v) = \min_{(u,v) \neq (0,0)} B(u, v),$$

we get

$$\min_{(u,v) \neq (0,0)} S(u, v) = \min_{(u,v) \neq (0,0)} \hat{S}(u, v),$$

which was to be proved. \square

The next lemma gives the exact solution for MIN, which by Lemma 3 is the minimum value of $\hat{S}(u, v)$. Before stating the lemma, we introduce some number theoretic notations which will be used in it.

Let ξ be a real number, and let $\{h_n/k_n\}$ be the sequence of convergents in its continued fraction expansion (see [6, chap. 10]). By the definition of continued fractions, for any triple of consecutive convergents h_{n-1}/k_{n-1} , h_n/k_n , h_{n+1}/k_{n+1} , there exists an integer $a_{n+1} \geq 0$ such that $h_{n-1} + a_{n+1}h_n = h_{n+1}$ and $k_{n-1} + a_{n+1}k_n = k_{n+1}$. The rational numbers in the (finite) sequence

$$\frac{h_{n-1}}{k_{n-1}}, \frac{h_{n-1} + h_n}{k_{n-1} + k_n}, \frac{h_{n-1} + 2h_n}{k_{n-1} + 2k_n}, \dots, \frac{h_{n-1} + a_{n+1}h_n}{k_{n-1} + a_{n+1}k_n} = \frac{h_{n+1}}{k_{n+1}}$$

are called the *secondary convergents* of ξ (see [8, pp. 162–163]).

We say that the rational number a/b is a *fair approximation* to ξ if

$$\left| \xi - \frac{a}{b} \right| = \min_{\substack{0 < y \leq b \\ x, y \text{ integers}}} \left| \xi - \frac{x}{y} \right|.$$

It is known that every fair approximation of ξ is either a convergent or a secondary convergent of ξ . Thus, if x/y is neither a convergent nor a secondary conver-

gent of ξ , then there is a rational a/b with $0 < b < |y|$ such that $|\xi - x/y| > |\xi - a/b|$.

LEMMA 4. Let $\hat{S}(u, v) = (|u| + 1)(|F_{2r}u - F_{2r+1}v| + 1)$. Then we have

$$\min_{(u,v) \neq (0,0)} \hat{S}(u, v) = (F_r + 1)(F_{r+1} + 1).$$

PROOF. We first show that

$$\begin{aligned} \text{MIN} &= \min_{(u,v) \neq (0,0)} \hat{S}(u, v) \\ &= \min_{(u,v) \neq (0,0)} (|u| + 1)(|F_{2r}u - F_{2r+1}v| + 1) \\ &= \min_{0 \leq n \leq 2r+1} (F_n + 1)(F_{2r-n+1} + 1), \end{aligned}$$

and then show that the latter minimum is $(F_r + 1)(F_{r+1} + 1)$.

It suffices to consider nonnegative values of u since $\hat{S}(u, v) = \hat{S}(-u, -v)$. The value MIN cannot exceed $\hat{S}(0, 1) = F_{2r+1} + 1$, but because $\hat{S}(u, v) \geq u + 1$ (the right factor is at least one), we need only seek a better value for MIN in the interval $0 < u < F_{2r+1}$.

The key idea is to divide the half-open interval $[1, F_{2r+1})$ into subintervals $[F_n, F_{n+1})$ for $n = 2, 3, \dots, 2r$. (Notice that $F_1 = F_2 = 1$, and thus n starts from 2.) The integer u lies inside one of these intervals. Consider the fraction F_{2r}/F_{2r+1} . The convergents of its continued fraction expansion are $F_1/F_2, F_2/F_3, \dots, F_{2r}/F_{2r+1}$. Since $(F_{n-1} + F_n)/(F_n + F_{n+1}) = F_{n+1}/F_{n+2}$, the secondary convergents of F_{2r}/F_{2r+1} coincide with the convergents of its continued fraction expansion. By the result quoted above, if $F_n \leq u < F_{n+1}$, then for every integer v we have

$$\left| \frac{F_{2r}}{F_{2r+1}} - \frac{v}{u} \right| \geq \left| \frac{F_{2r}}{F_{2r+1}} - \frac{F_{n-1}}{F_n} \right|.$$

Multiplying through on both sides yields

$$\left| \frac{uF_{2r} - vF_{2r+1}}{uF_{2r+1}} \right| \geq \left| \frac{F_{2r}F_n - F_{n-1}F_{2r+1}}{F_{2r+1}F_n} \right|.$$

Using the Fibonacci identity $|F_iF_j - F_{i+1}F_{j-1}| = F_{i-j+1}$, we get

$$\begin{aligned} |uF_{2r} - vF_{2r+1}| &\geq \frac{u}{F_n} |F_{2r}F_n - F_{n-1}F_{2r+1}| \\ &\geq |F_{2r}F_n - F_{n-1}F_{2r+1}| \\ &= F_{2r-n+1}. \end{aligned}$$

To summarize, if u falls in the interval $[F_n, F_{n+1})$, then $|uF_{2r} - vF_{2r+1}| \geq F_{2r-n+1}$. Therefore, we have

$$(|u| + 1)(|F_{2r}u - F_{2r+1}v| + 1) \geq (F_n + 1)(F_{2r-n+1} + 1),$$

and equality is achieved when $u = F_n$ and $v = F_{n-1}$. As a result, we have

$$\min_{(u,v) \neq (0,0)} (|u| + 1)(|F_{2r}u - F_{2r+1}v| + 1) = \min_{0 \leq n \leq 2r+1} (F_n + 1)(F_{2r-n+1} + 1),$$

which completes the first part of the proof.

The second part of the proof is to show that indeed

$$\min_{0 \leq n \leq 2r+1} (F_n + 1)(F_{2r-n+1} + 1) = (F_r + 1)(F_{r+1} + 1).$$

If we define

$$E(n, r) \stackrel{\text{def}}{=} (F_n + 1)(F_{2r-n+1} + 1),$$

then what we want to show is

$$\min_{0 \leq n \leq 2r+1} E(n, r) = E(r, r).$$

Since $E(n, r)$ is invariant when n is replaced by $2r + 1 - n$, it suffices to consider values of n in the interval $[0, r]$.

We now show that $E(n, r)$ is no smaller than $E(n + 1, r)$ for $n = 1, \dots, r - 1$, after which we shall complete the proof by demonstrating that $E(0, r) \geq E(r, r)$. We make use of the explicit formula

$$F_r = \frac{\phi^r - \hat{\phi}^r}{\sqrt{5}}$$

for a Fibonacci number in terms of the golden ratio ϕ and its conjugate $\hat{\phi} = \frac{1}{2}(1 - \sqrt{5})$ in order to obtain an alternative expression for the high-order term of $E(n, r)$:

$$\begin{aligned} F_n F_{2r+1-n} &= \left(\frac{\phi^n - \hat{\phi}^n}{\sqrt{5}} \right) \left(\frac{\phi^{2r+1-n} - \hat{\phi}^{2r+1-n}}{\sqrt{5}} \right) \\ &= \frac{\phi^{2r+1} + \hat{\phi}^{2r+1} - \phi^{2r+1-n} \hat{\phi}^n - \hat{\phi}^{2r+1-n} \phi^n}{5} \\ &= C_r + \frac{(-1)^{n+1}}{\sqrt{5}} \left(F_{2r-2n+1} + \frac{2\hat{\phi}^{2r-2n+1}}{\sqrt{5}} \right), \end{aligned}$$

where C_r is a constant depending on r alone. Taking advantage of the fact that $|\hat{\phi}|$ is less than 1 and using the basic recurrence for Fibonacci numbers, we have

$$\begin{aligned} E(n, r) - E(n + 1, r) &= F_{2r+1-n} + F_n + \frac{(-1)^{n+1}}{\sqrt{5}} \left(F_{2r-2n+1} + \frac{2\hat{\phi}^{2r-2n+1}}{\sqrt{5}} \right) \\ &\quad - F_{2r-n} - F_{n+1} - \frac{(-1)^{n+2}}{\sqrt{5}} \left(F_{2r-2n-1} + \frac{2\hat{\phi}^{2r-2n-1}}{\sqrt{5}} \right) \\ &\geq F_{2r-n-1} - F_{n-1} - \frac{F_{2r-2n}}{\sqrt{5}} - 1 \\ &\geq F_{2r-n-1} - F_{n-1} - F_{2r-2n} \\ &\geq 0, \end{aligned}$$

and hence $E(n, r)$ is at least as large as $E(r, r)$ for $n = 1, \dots, r - 1$.

As for the remaining inequality $E(0, r) \geq E(r, r)$, it is merely $F_r F_{r+1} + F_{r+2} + 1 \leq F_{2r+1} + 1$, and its truth may be verified by using the identity $F_r^2 + F_{r+1}^2 = F_{2r+1}$, which in turn follows from the Fibonacci identity $F_{i+j} = F_j F_{i+1} + F_{j-1} F_i$. \square

LEMMA 5. *The minimum size of a rectilinearly oriented rectangle that contains two points of the lattice generated by the basis vectors (F_r, F_{r+1}) and $(-F_{r+1}, F_r)$ is*

$$\text{MIN} = (F_r + 1)(F_{r+1} + 1).$$

PROOF. The proof follows directly from Lemmas 3 and 4. \square

THEOREM 6. *Let $M = F_{2r+1}$, and let $N = F_r F_{r+1} + F_{r+2}$. Then there is an organization for raster-graphics memory with M memory chips such that every rectilinearly oriented rectangle of size at most N contains pixels from distinct memory chips. Furthermore, N is greater than $M/\sqrt{5}$.*

PROOF. From Lemma 5, we have that $\text{MIN} = (F_r + 1)(F_{r+1} + 1)$, and since $N = \text{MIN} - 1$, we get $N = F_r F_{r+1} + F_{r+2}$. All that is left to be proved is that $N > M/\sqrt{5}$. Using the explicit formula for Fibonacci numbers, it can be verified that the sequence

$$\left\{ \frac{F_r F_{r+1} + F_{r+2}}{F_{2r+1}} \right\}_{r=1}^{\infty}$$

converges to $1/\sqrt{5}$. We now show that this sequence is monotonically decreasing, so each of its elements is at least as large as the $1/\sqrt{5}$ limit, which will complete the proof.

It is enough to show that the difference of consecutive terms in the sequence is positive, or equivalently, by multiplying through that

$$F_{2r+3}(F_r F_{r+1} + F_{r+2}) - F_{2r+1}(F_{r+1} F_{r+2} + F_{r+3}) > 0.$$

Using the explicit formula for Fibonacci numbers, we obtain the identity

$$F_{2r+3}F_r - F_{2r+1}F_{r+2} = (-1)^{r+1}F_{r+1},$$

and the identity

$$F_{2r+3}F_{r+2} - F_{2r+1}F_{r+3} = F_{2r+1}F_r + F_{2r}F_{r+2}$$

may be derived by induction.

Multiplying both sides of the first identity by F_{r+1} and adding it to the second yields

$$\begin{aligned} & F_{2r+3}(F_r F_{r+1} + F_{r+2}) - F_{2r+1}(F_{r+1} F_{r+2} + F_{r+3}) \\ &= F_{2r+1}F_r + F_{2r}F_{r+2} + (-1)^{r+1}F_{r+1}^2. \end{aligned}$$

The right-hand side is positive because F_{r+1} is less than both F_{2r} and F_{r+2} . \square

5. Optimality of the Fibonacci Lattice Organization

This section shows that the Fibonacci lattice organization from Section 3 achieves essentially the best possible by providing bounds for any raster-graphics memory organization. In order to get the bounds for the memory organization, an upper bound is first proved in the continuous model for the density of a set of compatible points. In particular, we show that any set S of compatible points in the plane has density $d(S) \leq 1/\sqrt{5}$.

In order to prove the density bound for a compatible set S , we consider a bounded region of S . The points of S in this region can be triangulated in such a way that most of the triangles have large area (at least $\sqrt{5}/2$), and hence the density of S in this region is small (at most $1/\sqrt{5} - o(1)$). Taking the limit as the radius of the region tends to infinity then establishes the upper bound of $1/\sqrt{5}$. First, however, we introduce some terminology.

Definition. Let $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$ be two points in the plane. We denote the (open) rectilinearly oriented rectangle defined by P_1 and P_2 as $R(P_1, P_2)$ and its area as $A(P_1, P_2) = |x_1 - x_2| |y_1 - y_2|$. We also denote the semiperimeter of the rectangle (the l_1 norm) as $L(P_1, P_2) = |x_1 - x_2| + |y_1 - y_2|$.

If P_1 and P_2 are compatible points, then $A(P_1, P_2) \geq 1$. Also, we have

$$\begin{aligned} L(P_1, P_2) &= |x_1 - x_2| + |y_1 - y_2| \\ &\geq 2\sqrt{|x_1 - x_2||y_1 - y_2|} \\ &\geq 2, \end{aligned}$$

because the arithmetic mean is at least the geometric mean.

We now define the notion of *good* and *bad* triangles, and show that a good triangle has area at least $\sqrt{5}/2$.

Definition. Let P_1, P_2 , and P_3 be compatible points in the plane. We say that triangle $\triangle P_1P_2P_3$ is a *good triangle* if $P_1 \notin R(P_2, P_3)$, $P_2 \notin R(P_1, P_3)$, and $P_3 \notin R(P_1, P_2)$, and a *bad triangle* otherwise.

Figure 5 gives an example of a good triangle and a bad triangle. In the bad triangle of the figure, we call the edge $\overline{P_2P_3}$ the *bad edge*, and we call the angle $\angle P_2P_1P_3$ the *bad angle*.

The next lemma provides a lower bound on the area of a good triangle.

LEMMA 7. *Any good triangle has area at least $\sqrt{5}/2$.*

PROOF.¹ Without loss of generality, we assume that the triangle is defined by the three points $(0, 0)$, (x_1, y_1) , and (x_2, y_2) , where $0 < x_2 < x_1$ and $0 < y_1 < y_2$, because any good triangle can be brought to this position by translation and reflections about the axes. The areas A , B , and C of the three rectangles defined by pairs of these points are each constrained to be at least 1 since the points are compatible, and hence

$$\begin{aligned} A &= x_1y_1 \geq 1, \\ B &= x_2y_2 \geq 1, \\ C &= (x_1 - x_2)(y_2 - y_1) \geq 1. \end{aligned}$$

The area of the triangle is $\frac{1}{2}(x_1y_2 - x_2y_1)$, which we wish to show is at least $\sqrt{5}/2$.

Substituting $y_1 = A/x_1$ and $x_2 = B/y_2$ into this last inequality yields

$$C = \left(x_1 - \frac{B}{y_2}\right)\left(y_2 - \frac{A}{x_1}\right).$$

Multiplying through by x_1y_2 gives

$$(x_1y_2)^2 - (A + B + C)x_1y_2 + AB = 0.$$

Similarly, substituting $x_1 = A/y_1$ and $y_2 = B/x_2$ into the third equation gives

$$(x_2y_1)^2 - (A + B + C)x_2y_1 + AB = 0.$$

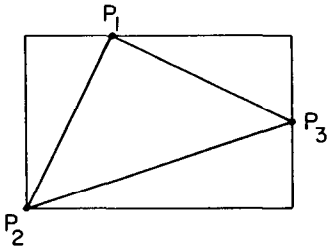
Thus, both x_1y_2 and x_2y_1 are roots of the equation $s^2 - (A + B + C)s + AB = 0$, and, since $x_1y_2 > x_2y_1$, we have

$$x_1y_2 = \frac{1}{2}(A + B + C + \sqrt{(A + B + C)^2 - 4AB})$$

and

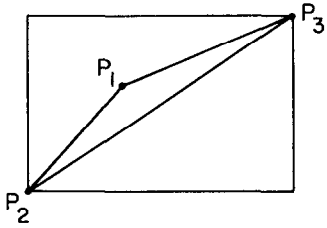
$$x_2y_1 = \frac{1}{2}(A + B + C - \sqrt{(A + B + C)^2 - 4AB}).$$

¹ Thanks to Don Coppersmith of IBM who provided this proof, which is simpler than our original.



(a)

FIG. 5. (a) A good triangle. (b) A bad triangle.



(b)

We can now bound the area of the triangle by simple algebraic manipulation:

$$\begin{aligned}
 \frac{1}{2}(x_1y_2 - x_2y_1) &= \frac{1}{2}\sqrt{(A + B + C)^2 - 4AB} \\
 &= \frac{1}{2}\sqrt{(A - B)^2 + C^2 + 2AC + 2BC} \\
 &\geq \frac{1}{2}\sqrt{0 + 1 + 2 + 2} \\
 &= \frac{1}{2}\sqrt{5}.
 \end{aligned}$$

□

Although the area of a good triangle is bounded from below, the area of a bad triangle can be arbitrarily small. In order to get a good upper bound on the density of a compatible set, we now show that any disk-shaped portion of such a set has a triangulation with few bad triangles.

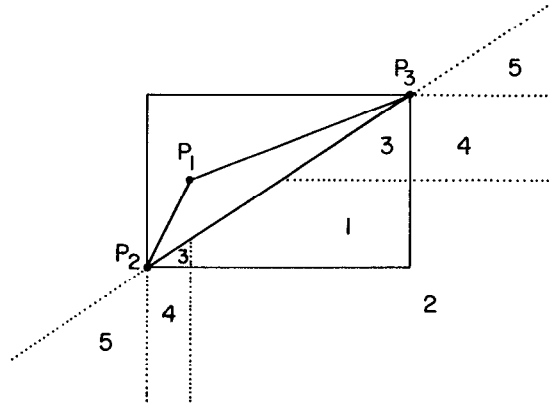
LEMMA 8. *Let S be a set of compatible points in the plane, and let \mathcal{D}_r be a disk of radius r . Then there is a triangulation of the points in $S \cap \mathcal{D}_r$, which contains no more than $4r$ bad triangles.*

PROOF. Consider the triangulations of the points in $S \cap \mathcal{D}_r$, that minimize the number of bad triangles, and of those pick one that minimizes the sum of the bad angles of the bad triangles. Take any bad triangle $\Delta P_1P_2P_3$ with bad edge $\overline{P_2P_3}$. If the bad edge $\overline{P_2P_3}$ does not lie on the boundary of the convex hull of $S \cap \mathcal{D}_r$, then there is a fourth point $P_4 \in S \cap \mathcal{D}_r$, such that triangle $\Delta P_2P_3P_4$ shares the edge $\overline{P_2P_3}$. The point P_4 is lying in the half plane defined by $\overline{P_2P_3}$, which does not contain P_1 .

Figure 6 shows a bad triangle in the triangulation, which is the general case except for reflections about the axes. The figure also outlines the five possible regions in which P_4 could lie. We show by case analysis that in a "minimal" triangulation, the only feasible region for P_4 is Region 5. If P_4 is in any of the other regions, replacing edge $\overline{P_2P_3}$ with edge $\overline{P_1P_4}$ would improve the triangulation thus contradicting its minimality.

Case 1. If P_4 is in Region 1, we replace the two bad triangles $\Delta P_1P_2P_3$ and $\Delta P_2P_3P_4$ by the two good triangles $\Delta P_1P_2P_4$ and $\Delta P_1P_3P_4$.

FIG. 6. A bad triangle whose bad edge $\overline{P_2P_3}$ is not on the boundary, and the five regions in which point P_4 could lie.



Case 2. If P_4 is in Region 2, we replace the bad triangle $\Delta P_1P_2P_3$ and the good triangle $\Delta P_2P_3P_4$ by two good triangles $\Delta P_1P_2P_4$ and $\Delta P_1P_3P_4$.

Case 3. If P_4 is in Region 3, we replace the bad triangles $\Delta P_1P_2P_3$ and $\Delta P_2P_3P_4$ by the bad triangles $\Delta P_1P_2P_4$ and $\Delta P_1P_3P_4$. Although this modification does not reduce the number of bad triangles, it does reduce the sum of the bad angles. We assume without loss of generality that P_4 is in the upper right portion of Region 3. The two bad angles were originally $\angle P_2P_1P_3$ and $\angle P_2P_4P_3$, and they are replaced by the bad angles $\angle P_2P_1P_4$ and $\angle P_2P_4P_3$. This is an improvement since $\angle P_2P_1P_4 < \angle P_2P_1P_3$ and $\angle P_1P_4P_3 < \angle P_2P_4P_3$.

Case 4. If P_4 is in the upper right portion of Region 4, we replace the bad triangle $\Delta P_1P_2P_3$ and the good triangle $\Delta P_2P_3P_4$ by the bad triangle $\Delta P_1P_2P_4$ and the good triangle $\Delta P_1P_3P_4$. The new bad angle $\angle P_2P_1P_4$ is smaller than the original bad angle $\angle P_2P_1P_3$. The lower left portion is dealt with similarly.

Thus we may conclude that the point P_4 is in Region 5, and without loss of generality we assume it is in the upper right portion of the region. The triangle $\Delta P_2P_3P_4$ is bad, and $L(P_2, P_4) = L(P_2, P_3) + L(P_3, P_4)$. Since the points P_3 and P_4 are compatible, we have already shown that $L(P_3, P_4) \geq 2$, and hence $L(P_2, P_4) \geq L(P_2, P_3) + 2$.

Applying the same arguments to the triangle $\Delta P_2P_3P_4$, we obtain a chain of adjacent bad triangles with increasing bad-edge lengths in the L norm. The chain cannot cycle back on itself because the edge lengths are strictly increasing. Thus the chain must terminate with a bad edge $\overline{P_iP_j}$ on the boundary of the convex hull of $S \cap \mathcal{D}_r$. In fact, the bad edge $\overline{P_iP_j}$ can be the terminating edge of more than one chain because there can be a tree of bad triangles rooted at the triangle with bad edge $\overline{P_iP_j}$ on the boundary of the convex hull of $S \cap \mathcal{D}_r$ (see Figure 7).

If the tree contains k triangles, then the boundary edge $\overline{P_iP_j}$ has length $L(P_i, P_j) \geq 2k + 2$, which we now show by induction on k . For $k = 0$, the length bound holds for any two compatible points. Let k_1 be the number of bad triangles in the subtree converging to $\overline{P_iP_i}$, and let k_2 be the number converging to $\overline{P_iP_j}$. (The values k_1 and k_2 may be zero.) Then $k = k_1 + k_2 + 1$, and hence the induction hypothesis holds for both subtrees because k_1 and k_2 are each less than k . Therefore,

$$\begin{aligned} L(P_i, P_j) &= L(P_i, P_i) + L(P_i, P_j) \\ &\geq (2k_1 + 2) + (2k_2 + 2) \\ &= 2(k_1 + k_2 + 1) + 2 \\ &= 2k + 2, \end{aligned}$$

as desired.

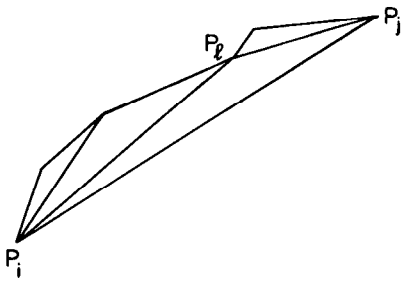


FIG. 7. A tree of four bad triangles that terminate with edge $\overline{P_i P_j}$.

We have just shown that if the tree has k triangles, then the boundary edge $\overline{P_i P_j}$ has length $L(P_i, P_j) \geq 2k + 2$, and hence $k < \frac{1}{2}L(P_i, P_j)$. Furthermore, since the trees rooted at two bad boundary edges consist of disjoint sets of bad triangles, we can bound the total number of bad triangles in the whole triangulation in terms of the length (in the L norm) of the boundary of the convex hull of $S \cap \mathcal{D}_r$. But the length of the boundary is at most $8r$, and therefore, the total number of bad triangles is less than $8r/2 = 4r$. \square

Lemma 8 shows that a set of compatible points in a disk can be triangulated with few bad triangles. Therefore, most of the triangles are good triangles which, by Lemma 7, have large area. These results allow us to give a $1/\sqrt{5}$ upper bound on the density of compatible sets.

THEOREM 9. *Let S be a set of compatible points in the plane. Then the density $d(S)$ of S satisfies $d(S) \leq 1/\sqrt{5}$.*

PROOF. Let \mathcal{D}_r be a disk of radius r , let n be the number of points in $S \cap \mathcal{D}_r$, and suppose the boundary of the convex hull of $S \cap \mathcal{D}_r$ contains m points. By Lemma 8, there exists a triangulation of the n points with at most $4r$ bad triangles. Every triangulation of the m points contains $2n - m - 2$ triangles, and thus the number of good triangles is at least $2n - m - 2 - 4r$. A lower bound on the length in the L norm of the boundary of the convex hull of $S \cap \mathcal{D}_r$ is $2m$, and an upper bound is $8r$. Hence $m \leq 4r$, and the number of good triangles is at least $2n - 8r - 2$.

By Lemma 7 the area of a good triangle is at least $\sqrt{5}/2$, and thus the total area occupied by the good triangles is at least $(2n - 8r - 2)(\sqrt{5}/2) = (n - 4r - 1)\sqrt{5}$. The area of the good triangles cannot exceed the circle area, so $(n - 4r - 1)\sqrt{5} \leq \pi r^2$ and $n \leq \pi r^2/\sqrt{5} + 4r + 1$. The density of points within $S \cap \mathcal{D}_r$ is just

$$\frac{n}{\pi r^2} \leq \frac{1}{\sqrt{5}} + \frac{4r + 1}{\pi r^2}.$$

Letting $r \rightarrow \infty$ implies $d(S) \leq 1/\sqrt{5}$, as desired. \square

We are now prepared to show that the Fibonacci lattice organization from Section 3 is essentially the best possible.

THEOREM 10. *For any organization of raster graphics memory with M memory chips such that every rectilinearly oriented rectangle of size N contains no two pixels from the same memory chip, the relation $M \geq \sqrt{5}N - O(N^{3/4})$ holds.*

PROOF SKETCH. The proof parallels that of Theorem 9. The principal difference is that the size of a rectangle includes not only its area, but also half its perimeter plus 1.

Let S be a set of grid points in the plane such that for every pair of points $P_1 = (x_1, y_1)$, $P_2 = (x_2, y_2)$ in it, the compatibility condition $(|x_1 - x_2| + 1)(|y_1 - y_2| + 1) \geq N$ holds. Multiplying through yields $|x_1 - x_2| |y_1 - y_2| \geq N - |x_1 - x_2| - |y_1 - y_2| - 1$. If P_1 and P_2 are two points contained within a circle of radius r , then we have $|x_1 - x_2| + |y_1 - y_2| \leq 2\sqrt{2}r$, so $|x_1 - x_2| |y_1 - y_2| \geq N - 2\sqrt{2}r - 1$.

Letting $c = N - 2\sqrt{2}r - 1$ and using the notation of Section 2, we have $A(P_1, P_2) \geq c$. Using the same techniques as in the proof of Lemma 7, Lemma 8, and Theorem 9, we can prove that the density d_r of points in $S \cap \mathcal{D}_r$ satisfies $cd_r \leq 1/\sqrt{5} + (4r' + 1)/\pi r'^2$, where $r' = r/\sqrt{c}$. Letting $r = N^{3/4}$, we have $r' \geq N^{1/4}$, and hence

$$(N - 2\sqrt{2}N^{3/4} - 1)d_r \leq \frac{1}{\sqrt{5}} + \frac{4N^{1/4} + 1}{\pi N^{1/2}}.$$

Since d_r is an upper bound on the density of pixels which can be stored on a single memory chip, we have $M \geq 1/d_r$, and thus

$$M \geq \sqrt{5}N \frac{(1 - 2\sqrt{2}N^{-1/4} - 1/N)}{(1 + (\sqrt{5}/\pi)[(4N^{1/4} + 1)/N^{1/2}])} = \sqrt{5}N - O(N^{3/4}). \quad \square$$

6. Addressing Scheme

The organization for raster-graphics memory proposed in Section 3 guarantees that small rectangles contain pixels from distinct memory chips. In order for the entire system performance to benefit from this organization, however, the address calculations must be easily implemented. We do not try to solve all the engineering problems associated with making this memory organization scheme work, but in this section we give indications of how the address calculations might be efficiently computed.

The addressing mechanism must be able to convert the x - and y -coordinates of a pixel to the chip number and address within the chip. Suppose the lattice organization is determined by two basis vectors (a, b) and $(-b, a)$. Two pixels at locations (x_0, y_0) and (x, y) that differ by an integral linear combination of the basis vectors lie in the same memory chip. That is, they have the same memory number if there exist (unique) integers U and V , such that

$$(x, y) - (x_0, y_0) = U(a, b) + V(-b, a).$$

One natural, but inefficient, addressing mechanism is based on the fact that each of the $M = a^2 + b^2$ memory chips contains exactly one representative in the basic region with corners $(0, 0)$, (a, b) , $(-b, a)$ and $(a - b, a + b)$. The chip number of a pixel (x, y) can be determined by computing which pixel (x_0, y_0) in the basic region is from the same chip, and then using the ordered pair (x_0, y_0) as the chip number. By letting

$$U = \left\lfloor \frac{ax + by}{a^2 + b^2} \right\rfloor, \quad V = \left\lfloor \frac{ay - bx}{a^2 + b^2} \right\rfloor,$$

the chip number (x_0, y_0) of a pixel (x, y) is then $(x_0, y_0) = (x, y) - U(a, b) - V(-b, a)$. Furthermore, the ordered pair (U, V) forms an appropriate address for a pixel (x, y) within the chip.

The addressing mechanism can be simplified substantially if we notice that any arbitrary set of M pixels, no two of which are from the same chip, can be used as

a set of representatives. In particular, any pixel differs by an integral linear combination of the basis vectors from a unique pixel in the horizontal line extending from $(0, 0)$ to $(0, M - 1)$. This scheme corresponds to tiling the plane with 1-by- M bricks instead of tilted squares. (Holladay [7] uses a similar tiling scheme for halftone generation.)

To derive an appropriate addressing scheme, we choose an alternative pair of basis vectors that span the same lattice. Since a and b are relatively prime, there exist integers k and l such that $ak - bl = 1$. The two vectors $(bk + al, 1)$ and $(a^2 + b^2, 0)$ generate the same lattice as the original basis vectors (a, b) , $(-b, a)$. Thus any pixel (x, y) can be mapped to a pixel (x_0, y_0) where $y_0 = 0$ and $x_0 \in [0, M)$, which means x_0 alone can serve as the chip number for the pixel. If we denote $C = bk + al$, and recalling that $M = a^2 + b^2$, the chip number for an arbitrary pixel (x, y) is $x - Cy(\text{mod } M)$. The address of the pixel is the ordered pair $(\lfloor x/M \rfloor, y)$, which is also easy to compute.

An advantage of any doubly periodic organization that should be mentioned concerns the communication among the memory chips. Typically, each chip has a single connection to an M -pixel buffer. To move a rectangle of pixels, three steps are required. The rectangle of pixels is read into the buffer, the pixels in the buffer are permuted, and the pixels are written back to the memory chips at different locations. The advantage of the periodic organization is that the set of permutations encompasses only circular shifts of the buffer. Thus a standard barrel shifter can be used for all permutations.

One issue that we have not faced is the problem of generating addresses for each of the M chips given some standard specification of the rectangle to be accessed. Whether the address calculations can be made possible at reasonable cost requires an engineering analysis. The competing concerns are the strong regularity of the lattice-based organization, which should help the design, versus the need to perform modular arithmetic, which could require much hardware.

7. Comments

The Fibonacci lattice organization of memory allows all rectilinearly oriented rectangles of a given size to be accessed. Not surprisingly, some economy in hardware can be gained by being more restrictive. For example, the memory organization based on the lattice generated by the basis vectors $(1, s)$ and $(-s, 1)$ allows three types of rectangles— s -by- s , 1-by- $s^2 + 1$, and $s^2 + 1$ -by-1—to be accessed efficiently. The number of memories required by this scheme is $M = s^2 + 1$.

The Fibonacci lattice organization can also be used to speed up the access rate in machines with interleaved memories. For example, the organization might be useful for matrix- and image-processing applications [1].

An interesting question is how to extend the constructions and bounds of this paper to dimensions higher than two. For example, the analogous question for three dimensions would be, "How does one construct a dense set of points in the three-dimensional space, such that no two points of the set are contained in the interior of a rectilinearly oriented box of volume 1?" and, if such construction is at all possible, "What is the maximum density possible for such a set?" In fact, we can construct a lattice whose density is $\frac{1}{\sqrt{3}}$, satisfying the "compatibility" requirements. Using a tilted cube of edge length $\sqrt{3}$, we can show (by an argument similar to that in [2, Theorem 1]) that $1/3\sqrt{3}$ is an upper bound on the density of such sets. It remains to be seen whether the techniques from Section 2 can be applied to achieve a tighter lower bound for the three-dimensional case.

Fiat and Shamir [4] have recently pointed out another application of the techniques in this paper. They show that a "universal" systolic array based on a Lucas and Fibonacci lattice can emulate any rectangular systolic array in a near optimal fashion. In fact, by double folding the basic square region, the Fibonacci lattice of our paper achieves almost all the properties they show of their "polymorphic arrays." The address calculation for their lattice seems somewhat simpler, however. No doubt, further research will lead to better understanding of the theory underlying both of our structures.

ACKNOWLEDGMENTS. A preliminary analysis of the continuous model was performed in conjunction with Satish Gupta and Don Hoey of Carnegie-Mellon University. Don Coppersmith of IBM contributed the elegant proof to Lemma 7, which replaced our more complicated argument. Don also independently proved Theorem 9 after we left it as an open problem in our extended abstract [2]. Amos Fiat of Wietzman Institute pointed out an error in an earlier proof of Lemma 4. Ron Pinter of MIT helped write several computer programs that originally convinced us that Fibonacci memory organization really works. We would also like to thank Tom Leighton and Shai Haran of MIT for helpful discussions and Lyle Ramshaw of Xerox PARC for acquainting us with Reference [7]. Finally, thanks to the anonymous referees for their detailed comments.

REFERENCES

1. CATMULL, E., AND SMITH, A. R. 3-D transformations of images in scanline order. *SIGGRAPH Conference Proceedings 14*, 1 (July 1980), 279-285.
2. CHOR, B., LEISERSON, C. E., AND RIVEST, R. L. An application of number theory to the organization of raster graphics memory (extended abstract). In *Proceedings of the 23rd Annual Symposium on Foundations of Computer Science* (October). IEEE, New York, 1982, pp. 92-99.
3. CLARK, J. H., AND HANNAH, M. R. Distributed processing in a high-performance smart image memory. *Lambda*, fourth quarter (1980), 40-45.
4. FIAT, A., AND SHAMIR, A. Polymorphic arrays: A novel VLSI layout for systolic computers. In *Proceedings of the 25th Annual Symposium on Foundations of Computer Science* (Oct.). IEEE, 1984, pp. 37-45.
5. GUPTA, S. Architectures and algorithms for parallel updates of raster scan displays. Ph.D. dissertation. Carnegie-Mellon Univ., Pittsburgh, Pa., 1981.
6. HARDY, G. H., AND WRIGHT, E. M. *An Introduction to the Theory of Numbers*. Oxford Univ. Press, London, 1968.
7. HOLLADAY, T. M. An optimum algorithm for halftone generation for displays and hard copies. *Proc. Soc. Inf. Display 21*, 2 (1980), 185-192.
8. NIVEN, I., AND ZUCKERMAN, H. S. *An Introduction to the Theory of Numbers*. Wiley, New York, 1972.
9. SPROULL, R. F., SUTHERLAND, I. E., THOMPSON, A., GUPTA, S., AND MINTER, C. The 8-by-8 display. Tech. Rep. CMU-CS-82-105. Carnegie-Mellon Univ., Pittsburgh, Pa., 1981.

RECEIVED FEBRUARY 1984; REVISED JUNE 1985; ACCEPTED JUNE 1985