Abstract

Iterative algorithms based on runs and runs of runs are presented to calculate the cells of the two–dimensional lattice intersected by a line of real slope and intercept. The technique is applied to the problem of traversing a ray through a two–dimensional grid. Using runs or runs of runs provides a significant improvement in the efficiency of ray traversal for all but very short path lengths when compared to the DDA algorithm implemented using floating or fixed point arithmetic.

Keywords: Digital geometry, line digitisation, ray traversal, ray tracing, volume visualisation.

1. Introduction

Volumetric data sets are becoming increasingly important within scientific and medical endeavors but our ability to generate and collect this type of information has quickly outgrown our ability to store, visualise and analyse it [9]. One fundamental technique for operating with volumetric data is the process of calculating the cells of the volume touched by a ray as the ray traverses the volume. Ray traversal techniques are central to a range of techniques for volume visualisation, computer generated imagery, collision detection, pattern recognition, visibility determination and computer vision.

However ray traversal algorithms are currently moribund to operating at the cell level by calculating the path of the ray a cell at a time. Solutions to the related problem of line drawing, which is essentially the process of digitising a line of rational slope and intercept, have been presented using runs [7], run length slices [2] and runs of runs [10, 11]. The use of higher order primitives in these techniques has provided a significant improvement in efficiency over pixel–based algorithms. However similar algorithms for ray traversal, the process of digitising a line with real slope and intercept, have not followed.

In this paper we present two techniques for iteratively calculating the cells of the two–dimensional lattice intersected by a two–dimensional line of real slope and intercept by using runs and runs of runs. We discuss the application of these results to the problem of ray traversal in two dimensions. To extend these results to three dimensions it is possible to use the two–dimensional algorithm in a similar manner to how the 3D–DDA algorithm uses the 2D–DDA algorithm [4, 5].

2. The Line and Ray

To describe the path of the ray through a two–dimensional lattice, we will represent the problem as the process of digitising the line $y = \alpha x + \beta$ where $\alpha$ and $\beta$ are algebraic real numbers. We will consider the minimal 8–connected version of the digital line such that the true–line intersects the left face of each cell in the digitised line at the point $(x_j, y_j + \beta_j^0)$ where $x_j$ and $y_j$ are integers, $\beta_j^0$ is real and $0 \leq \beta_j^0 < 1$. If the 4–connected line is required, the additional cells intersected by the line can be added by extending the beginning of each run of pixels within the line by one pixel except where a corner intersection occurs, as shown in Figure 1. A line is 8–connected if each pixel in the line $(x, y)$ is connected to another from the set $(x - 1, y - 1), (x, y - 1), (x + 1, y - 1), (x + 1, y), (x + 1, y + 1), (x, y + 1), (x - 1, y - 1), (x - 1, y)$ and 4–connected if each neighbour is from the set $(x, y - 1), (x + 1, y), (x, y + 1), (x - 1, y)$. To simplify the problem, we assume the slope of the line is within the range $0 \leq \alpha \leq 1$ as all other lines can be generated from this line given the 8–way symmetry of Cartesian space about its origin.

To describe the structure of the runs and runs of runs within the digital line or ray, we will concentrate on the line of real slope and zero intercept $y = \alpha x$. This is because
the structure of the line depends only on the slope of the line as can be seen by the definitions presented by Brons [3] and Wu [12]. A non-zero value of the intercept does not serve to alter the structure of the line but causes a shifting of the structure within the line. Therefore having described the structure of runs and runs of runs within the line, we will describe the effect of introducing a non-zero intercept to the line and apply our results to the problem of ray traversal.

3. Runs

A run is traditionally defined as a set of contiguous cells having the same \( y \) coordinate. It is characterised by its length, \( r_j \), the number of composite cells and its start point, \((x_j, y_j)\), the position of the left most cell in the run. To describe the minimal 8-connected digital line using runs, we can use the fact that each run in the line is corner connected. Therefore the line can be described as an ordered set of run lengths where the position of each composite run can be calculated from the position and length of the previous:

\[
(x_{j+1}, y_{j+1}) = (x_j + r_j, y_j + 1).
\] (1)

This reduces our problem of digitising the line to that of calculating the length of each composite run. A property of the lengths of the runs in a line critical to how we form this calculation is the fact that runs within a digital line occur in a maximum of two lengths which are consecutive integers [7, 8]. We are assuming that the line is unbounded as the extreme runs in any line segment may be truncated. However these runs have simply been truncated from one of the, at most, two possible lengths.

The case where only one run length exists occurs for lines of slope \( \frac{1}{k} \) where \( k = 2, 3, \ldots \). We will however ignore this case for now and discuss it later as it can be handled very efficiently as a special case. We therefore assume that two run lengths exist in the line.

As there are only two possible run lengths in the line, we can refer to each run as being either short or long and to discuss the structure of runs in a line, denote a short run by the symbol \( s \) and a long run by \( l \). For example, the run sequence \((lslslslslslsls)\) describes the structure of the runs in the digital line \( y = \frac{17}{22}x \) shown in Figure 2, our example line. The long runs in the figure are designated by light grey and the short runs, dark grey. The upper collections of runs we will use in our discussion of runs of runs. We should note that we choose a line of rational slope to allow a simpler discourse as this sequence of run lengths displayed in the line segment is repeated throughout the entire line.

To determine the lengths of runs that can occur in any given line, consider that the slope of the line over a segment of the line is given by \( \alpha = \frac{\delta y}{\delta x} \), where \( \delta x \) is the number of cells in the line segment and \( \delta y \) is the number of runs. Therefore, it can be said, the slope of the line defines the ratio of runs to cells in the line.

As the possible run lengths are two, integral and measured in cells, the length of a short run must be \( r = \left\lfloor \frac{1}{\alpha} \right\rfloor \) and the length of a long run \( r + 1 = \left\lceil \frac{1}{\alpha} \right\rceil \) as \( \frac{1}{\alpha} \) is the average length of a run in the line [7]. Within our example line segment in Figure 2, we have 41 cells and 17 runs. Therefore the possible run lengths are \( r = 2 \) and \( r + 1 = 3 \).

3.1. Sizing Them Up

Within Figure 2 we have also described the sequence of values of the fractional component of the intercept of the line at the start of each run, \( \beta_j \) for \( j = 0, 1, 2, \ldots \), in the upper sequence of numbers. We shall refer to this sequence of values as the intercept sequence. From Figure 2, it can be seen the values of the intercept sequence are all less then the slope of the line, \( \alpha = \frac{17}{22} \), and the long runs within the line correspond to the intercept values less than \( \frac{17}{22} \). The
short runs coincide with the values of the intercept sequence greater than or equal to \( \frac{2}{\pi} \). What is even more interesting is that the ratio of long runs to cells in the line is also \( \frac{2}{\pi} \), which is not coincidental.

Where \((x_j, y_j)\) is the coordinate of the first cell in a run in the line, the intercept sequence can be calculated by

\[
\beta_j = \alpha x_j - y_j \tag{2}
\]

hence the magnitude of each value of the intercept sequence is less than one, \(0 \leq \beta_j < 1\). From the definition of the connectivity of runs within the line given by Equation 1 and the definition of the intercept sequence given by Equation 2, we have the difference between two successive values in the intercept sequence:

\[
\beta_{j+1} - \beta_j = \alpha r - 1 \tag{3}
\]

Figure 3 describes the intercept geometry of a run and the continuous representation of the line. The line is set such that it intersects the end-point of a short run, \((x_{j+1}, y_{j+1}) = (x_j + r, y_j + 1)\) and therefore \(r_j = r\) and \(\beta_{j+1} = 0\). This position of the line is the critical position for deciding the length of the run. At this position, the value of intercept of the run and the line we denote as \(\nu\). If the line lies below this point, the run must be long. If the line lies on or above this point, the run must be short. Remember that these are the only two possibilities! Therefore when \(\beta_j \geq \nu\) the length of the run is short and when \(\beta_j < \nu\) long.

The value of \(\nu\) has an important correspondence to the structure of runs being the ratio of long runs to cells in the line. When we place the line on \(\nu\), we have \(\beta_j = \nu, r_j = r\) and \(\beta_{j+1} = 0\). As \(\alpha\) is the ratio of runs to cells and \(r\) is the length of a short run, from Equation 3, we have that

\[
\nu = 1 - \alpha r. \tag{4}
\]

Similarly we can define \(\mu\) to be the ratio of short runs in the line to cells:

\[
\mu = \alpha - \nu = \alpha(r + 1) - 1. \tag{5}
\]

![Figure 3. The geometry of the intercept of a run and the line.](image)

The ratios of long and short runs to cells in the line also play a role in the iterative calculation of the intercept sequence. The next value of the intercept sequence, \(\beta_{j+1}\), can be calculated from the current, \(\beta_j\), if the current run is short, \(r_j = r\), from Equations 3 and 4 by

\[
\beta_{j+1} - \beta_j = \alpha r - 1 = -\nu. \tag{6}
\]

If the run is long, \(r_j = r + 1\), by Equations 3 and 5, we have

\[
\beta_{j+1} - \beta_j = \alpha(r + 1) - 1 = \mu. \tag{7}
\]

This completes our definition for the structure of runs within the line \(y = \alpha r\). To extend this result for traversing a ray through a lattice, the essential difference between the line and ray is that the ray is projected outwards from a starting point and the path of the ray through the lattice is 4-connected except where a corner intersection occurs.

### 3.2. The First Step

As the ray emanates from a starting point, the first run in the path may be truncated. We can assume the starting point of the ray within the lattice is defined by \((x_0, y_0 + \beta)\) where \(x_0\) and \(y_0\) are integers and \(\beta\) is real such that \(0 \leq \beta < 1\). For the first run in the line to be truncated, its length must be less than that of a short run, \(r_0 < r\). The determining position of the continuous line for truncation to occur is such that the line intersects the point \((x_0 + r - 1, y_0 + 1)\) and

\[
\beta \geq 1 - (r - 1)\alpha = \alpha + \nu. \tag{8}
\]

Therefore if the line lies on or above this point the initial run is truncated.

If the first run is truncated, its length must be calculated as well as the first intercept sequence value that corresponds to the start of the first full length run in the path of the ray. From geometry, the length of a truncated run, where \(\Delta r = \left\lfloor \frac{\beta - \nu}{\alpha} \right\rfloor\), is

\[
r_0 = \frac{1 - \beta}{\alpha} = r + \frac{\nu - \beta}{\alpha} = r - \frac{\beta - \nu}{\alpha} = r - \Delta r. \tag{9}
\]

To calculate the first value in the intercept sequence from the value of the intercept of the line, we have that \(\beta_1 = \beta\) if \(\beta < \alpha + \nu\) as the first run is not truncated. If the first run is truncated however from Equation 3

\[
\beta_1 = \beta + r_0\alpha - 1 = \beta + (r - \Delta r)\alpha - 1 = (\beta - \nu) - \Delta r\alpha. \tag{11}
\]
To make the decision of run length and the initialisation of the initial truncated run length and initial intercept value more efficient, we can translate the values of the intercept sequence by the value $-\nu$ by initially subtracting $\nu$ from $\beta$ before initialising the algorithm. Therefore, the initial run is not truncated if $\beta - \nu < \alpha$ and hence $r_0 = 0$ and $\beta_1 = \beta - \nu$. If the initial run is truncated, our calculations of $r_0$ and $\beta_1$ are simplified, as is our decision of which run length is next in the line or path of the ray, which can be made against zero and implemented as a sign check. Also if $\frac{1}{2} < \alpha < 1$, a truncated run is not possible as the length of a short run is always one. Therefore we need only consider the possibility of an initial truncated run for slopes in the range $0 < \alpha < \frac{1}{2}$.

To convert the definition of the path of the ray from being $8$–connected to $4$–connected, we can extend each run back by one cell with no overhead to the algorithm. However this extension should only take place if there is no corner intersection, which can only occur at the beginning of a run coinciding with an intercept value of zero (or $-\nu$).

4. Runs of Runs

To discuss the structure of run of runs within the digital line, we will again be starting with the line of zero intercept, $y = \alpha x$. To define exactly what we will refer to as a run of runs, we will use a second important fact about the structure of runs in the line. Between any two runs of one of the two possible run lengths, there must be at least one run of the other length. Therefore within the run sequence of a line, patterns such as $ssll$ and $llss$ cannot occur. Also we know that the length of the first run in the line $y = \alpha x$ past the origin, $r_0 = \lceil \frac{1}{\alpha} \rceil$, is always long [10, 11].

Therefore we can define a run of runs to be the maximal sequence of contiguous long runs followed by the maximal sequence of contiguous short runs. At least one of these sequences will consist of only one run, the other at least one run. This definition explicitly implies one of two shapes onto each run of runs within the line, shape 0: $l^+s$ or shape 1: $ls^+$ where $l^+$ and $s^+$ denotes a sequence of one or more long or short runs respectively. Therefore, if there are less short runs in the line, the runs of runs will have shape 0 and less long runs than short, shape 1. Where $\alpha$ denotes the ratio of runs to cells, we will use $\alpha'$ to describe the ratio of runs of runs to cells in the line. From our definition of runs of runs, the number of runs of runs within the line is the number of the least occurring run length. Therefore, if $\mu < \nu$, the runs have shape 0: $l^+s$ and $\alpha' = \mu$. If $\nu < \mu$, we have shape 1: $ls^+$ and $\alpha' = \nu$.

Similar to the structure of runs, runs of runs also occur in the line in at most two lengths, which are consecutive integers when the length of a run of runs is measured in runs [8, 10]. To calculate the possible lengths of the runs of runs, short $r'$ and long $r' + 1$, ignoring for now the case where only one length exists, we use the ratio of runs of runs to runs. Therefore

$$r' = \left\lfloor \frac{\alpha}{\alpha'} \right\rfloor.$$  \hfill (12)

Within Figure 2 we have also described the seven runs of runs within our example line of $y = \frac{2}{7}x$. The runs of runs have shape 1: $ls^+$ as there are less long runs in the line than short. The run lengths that occur are $r' = 2$ and $r' + 1 = 3$.

To define the algorithm to construct the digital line using runs of runs, we will be following the development of the run–based algorithm. We will show how to iteratively calculate the intercept sequence and use this sequence to define the length of each run of runs. We will then describe how to adapt this process for ray traversal. As there exist two possible shapes of the runs of runs, we will consider each shape separately.

4.1. Shape 0: $1^+s$

Within a run of runs of shape 0: $l^+s$, length $r_j$, there are $r_j - 1$ long runs and one short run. The difference between two consecutive values within the intercept sequence associated with the runs of runs of shape 0 can be calculated from Equations 6 and 7 and the fact that $\alpha' = \mu$,

$$\beta_{j+1} - \beta_j = \frac{(r_j - 1)\mu - \nu}{r_j \alpha' - \alpha}. \hfill (13)$$

In Figure 4, the critical value of the intercept of a run of runs of shape 0 is described. In this case, the line is set such that it intersects the end of a short run therefore $\beta_j = 0$ and $r_j = r'$. If the line were to lie above or on this point the run of runs must be short. From the figure for example, the runs intersected would not include the cell with the dashed outline and the run lengths would be $(lls)$, which in this case is a short run. If the line was to lie below this point, the runs intersected in the figure would include the extra cell and the run sequence would be $(lll)$, which defines that the run of runs must be long. The critical value of the intercept sequence is

$$\beta_j = \alpha - r' \alpha' = 0.$$  \hfill (14)

To iteratively calculate the intercept sequence, if the run of runs is short from Equation 13

$$\beta_{j+1} - \beta_j = r' \alpha' - \alpha = -\nu.' \hfill (15)$$

and long

$$\beta_{j+1} - \beta_j = (r' + 1) \alpha' - \alpha = \mu'. \hfill (16)$$

To complete the treatment for runs of runs of shape 0, we must handle the possibility that the introduction of a non–zero intercept for the line causes the initial run and runs of
runs to be truncated. The first step is to determine if the first run is truncated and if so, calculate the length \( r_0 \) and the first value in the intercept sequence for runs \( \beta_1 \), the process for which we described in Section 3. The value of \( \beta_1 \), can then be used to determine if the first runs of runs is truncated.

A run of runs of shape 0: \( l^+s \) is truncated if its length, \( r_0' \), is shorter than a short run of runs, \( r_0' < r' - 1 \). Therefore the truncated run of runs must have at least \( r' - 2 \) long runs and 1 short run. Therefore the initial run is truncated if

\[
\beta_1 \geq \nu - (r' - 2)\mu \\
\geq \alpha' + \alpha - r'\alpha' \\
\geq \alpha' + \nu'.
\]

If the first run of runs is truncated, the length of the run of runs, where \( \Delta r' = \left[ \frac{\beta_1 - \nu'}{\alpha'} \right] \), is

\[
r_0' = \left[ \frac{\alpha - \beta_1}{\alpha'} \right] \\
\quad = r' - \left[ \frac{\beta_1 - \nu'}{\alpha'} \right] \\
\quad = r' - \Delta r'.
\]

If the first run of runs is truncated, to calculate the first value in the intercept sequence corresponding to the first full length run of runs, we know within a run of runs of shape 0 of length \( r_0' \) there are \( r_0' - 1 \) long runs and one short. Therefore from Equations 15 and 16

\[
\beta_1 = \beta_1 + (r_0' - 1)\mu - \nu \\
\quad = \beta_1 + (r' - 2\Delta r')\alpha' - \alpha' \\
\quad = (\beta_1 - \nu') - \Delta r'\alpha'.
\]

If the first run of runs is not truncated \( \beta_0' = \beta_1 \) as \( r_0' = 0 \).

Similar to our discussion of runs, a benefit can be found in translating the intercept sequence for runs of runs by \(-\nu'\).

### 4.2. Shape 1: \( ls^+ \)

Within a run of runs of shape 1: \( ls^+ \), length \( r_j' \), there are \( r_j' - 1 \) short runs and one long run. Therefore the difference between two consecutive values within the intercept sequence associated with the runs of runs of shape 1 can be calculated from Equations 6 and 7 given that \( \alpha = \mu + \nu \) and \( \alpha' = \nu' \):

\[
\beta_{j+1} - \beta_j = \mu - (r_j' - 1)\nu \\
= \alpha - r_j'\alpha'.
\]

In Figure 5, the intercept geometry of a run of runs with shape 1 is described. The line is set such that \( \beta_{j+1} = 0 \) and \( r_j' = r' + 1 \), the run of runs is long. If the line lies below this point, the runs within the figure will have the lengths \( (ls) \) and the run of runs must be short. Above or on the point, the last run is short, \( (ls) \), and the run of runs is long. Therefore the critical value for the values of the intercept sequence at the runs of runs level is defined to be

\[
\beta_j = (r' + 1)\alpha' - \alpha = \nu'
\]

where \( \nu' \) is defined such that \( \nu' = r'\alpha' - \alpha \) and \( \nu' \) such that \( \alpha' = \nu' + \mu' \). It can be seen in Figure 2 that the short runs of runs coincide with intercept sequence values which are less than \( \nu' = \frac{1}{4}\nu' \). The long runs of runs, with values greater than or equal to \( \nu' \).

---

**Figure 4. The intercept geometry of the line and a run of runs of shape 0: \( l^+s \).**

**Figure 5. The intercept geometry of a run of runs of shape 1: \( ls^+ \).**
also short. We know from Equation 4 that the length of the first run is short if

$$\beta_1 \geq \nu \geq \alpha'$$.

(24)

To calculate the first intercept value which coincides with a full run of runs within the line, we will use the fact that the remaining runs in the run of runs are all short. The values of the intercept sequence defined at the run level are decremented by \(\nu\) for a short run. Eventually a value in the intercept sequence will be less than \(\alpha'\). This value coincides with the first long run in the line and therefore the first full length run of runs. Therefore the length of the truncated run of runs, can be calculated by

$$r'_0 = \left\lfloor \frac{\beta_1}{\alpha'} \right\rfloor$$

(25)

and the first value in the intercept sequence for the run of runs is

$$\beta'_1 = \beta_1 - r'_0 \alpha'$$.

(26)

5. Special Cases

We have until now ignored the case where only one length run or one length run of runs exists in the line. If only one length run exists in the line, the slope of the line must have the form \(\frac{k}{d}\) for \(k = 2, 3, \ldots\). This case is special as the intercept sequence does not have to be calculated as we already know the lengths of each run in the line. It occurs when \(\nu = 0\) and dictates that there are no runs of runs in the line or in other words the length of the only run of runs in the line is infinite.

The case may also occur that each run of runs in the line is of the same length. For shape 0, this case occurs when \(\nu' = 0\) and the slope of the line has the form \(\frac{k}{d-1}\) for \(k = 2, 3, \ldots\). For shape 1, it occurs when \(\mu' = 0\) and the slope of the line is \(\frac{k}{d-1}\) for \(k = 2, 3, \ldots\). There is also a situation that can be easily detected when \(\nu = \mu\) and the runs of runs do not have a distinct shape. Their shape \(l_8\) belongs equally to the classification of shape 0 and 1. This situation occurs only in lines of slope \(\frac{2}{d+1}\) for \(k = 1, 2, \ldots\).

6. Results

To provide an experimental estimation of the efficiency of using runs and runs of runs to traverse a ray through a two–dimensional uniform spatial subdivision we sought to compare our algorithms against the DDA algorithm implemented using fixed point arithmetic, which is the standard for such an application. Two experiments were conducted to show the dependence of the behaviour of each algorithms on the length of the path of the ray and it slope. However the results we present should only be used to gain a graphical understanding of the performance of each algorithm as the results cannot be taken out of the context of the experiment.

To show the relative benefits of using runs and runs of runs for all but very short ray paths, the results of the experiment based on path length are presented in Figure 6. In total 100,000 rays with random slopes, intercepts were used with random path length between 1 and 128 pixels. For spatial subdivisions or expected path lengths of a reasonable size, the use of runs and runs of runs over cells or pixels induces an impressive speed–up. For small scale subdivisions, the use of runs and runs of runs may not be suitable for a given application due to the increase in initialisation costs of each algorithm. To improve the costs of initialising the algorithms a number of ideas for run–based line–drawing algorithms may also be suitable for ray–based applications [6, 10].

![Figure 6. The dependence of the ray traversal algorithms on the length of the path of the ray.](image)

While the dependence to the path length of the ray of each of the techniques is linear, the dependence on the slope of the line is more interesting and warrants closer inspection. For this experiment, the path length was set at 256 pixels and 100,000 rays of random slopes and intercepts were used. The results as presented in Figure 7.

The order of the DDA algorithm is \(O(n)\) where \(n\) is the number of pixels in the 8–connected path of the ray. However we are using fixed point arithmetic and a 4–connected line both of which complicate matters. Therefore as the slope increase and there are more pixels in the 4–connected line the performance decreases slightly. The order of the run–based algorithm is \(O(\alpha) = O(\frac{n}{r})\) where \(\alpha\) is the slope of the line (proportion of runs to cells) and \(r\) is the length of a short run [10]. The order of the runs of runs algorithm is \(O(\alpha') = O(\frac{m}{r'})\) where \(\alpha'\) is the proportion of runs of runs to runs and \(r'\) is the length of a short run of runs [10].
Hence the algorithm achieves its maximum efficiency when the length of the runs and the run of runs are at a maximum. The length of the runs is at a maximum for a slope of $BC$ and a minimum for a slope of $BD$, therefore there is a general trend for the performance of each algorithm to decline as the slope increases. The length of a run of runs for the lines of slope $CZ$ where $CZ = BP / BD$ is theoretically infinite and these values of slope coincide with the dips in the graph. The length of the runs of runs is at a minimum for lines of slope $BE$ where $BE = CZ / BP / BD$ each of which correspond with a peak in the graph.

7. Conclusion

The use of runs and runs of runs can be used to significantly accelerate the process of traversing a ray through a uniform spatial subdivision. Runs and runs of runs are however only the first two levels of a hierarchy of runs that exists within the digital line [10, 11]. We have however concentrated on only the first two orders of this hierarchy as runs and runs of runs offer the best solution for volume sizes in typical use today. As the average length of a run measured in cells at each level of the hierarchy increases exponentially with the level within the hierarchy, beyond runs of runs four shapes are possible $|s|$, $t^+s$, $s t^+$ and $s^+t$ and there is a relatively constant increase in the initialisation costs for each level used, it becomes more difficult to justify using high order algorithms unless the volume in question is extremely large. The development of higher order algorithms does however follow directly from the presentation we have made.

The use of runs and runs of runs to trace the digital ray also offers more than to simply make traversal more efficient. It offers another abstraction of the path of the ray that can be used as an alternative for spatial subdivisions. In their comparison of uniform and nonuniform spatial subdivisions based on the lattice, Arvo and Kirk [1] comment that while uniform spatial subdivisions allow the use of efficient incremental techniques for traversing the ray, nonuniform spatial subdivision are more sensitive to the distribution of space within the subdivision. Run–based spatial subdivisions offer the coherence to space provided by uniform spatial subdivisions, the coherence to the scene provided by nonuniform spatial subdivisions such as the octree and now, coherence to the structure of a ray. This coherence offers opportunities to further improve ray traversal algorithms.

References