The region of device space to be painted by the **image** operator is determined similarly to that of a filled shape, though not identically. The interpreter transforms the image's source rectangle into device space and defines a half-open region, just as for fill operations. However, only those pixels whose *centers* lie within the region are painted. The position of the center of such a pixel—in other words, the point whose coordinate values have fractional parts of one-half—is mapped back into source space to determine how to color the pixel. There is no averaging over the pixel area; if the resolution of the source image is higher than that of device space, some source samples will not be used.

For clipping, the clipping region consists of the set of pixels that would be included by a fill operation. Subsequent painting operations affect a region that is the intersection of the set of pixels defined by the clipping region with the set of pixels for the region to be painted.

Scan conversion of character shapes is performed by a different algorithm from the one above. That font rendering algorithm uses hints in the character descriptions and techniques that are specialized to character rasterization.

7.5.2 Automatic Stroke Adjustment

When a stroke is drawn along a path, the scan conversion algorithm may produce lines of nonuniform thickness because of rasterization effects. In general, the line width and the coordinates of the endpoints, transformed into device space, are arbitrary real numbers not quantized to device pixels. A line of a given width can intersect with different numbers of device pixels, depending on where it is positioned. Figure 7.6 illustrates this effect.

For best results, it is important to compensate for the rasterization effects to produce strokes of uniform thickness. This is especially important in low-resolution display applications. To meet this need, LanguageLevel 2 provides an optional *stroke adjustment* feature. When stroke adjustment is enabled, the line width and the coordinates of a stroke are automatically adjusted as necessary to produce lines of uniform thickness. The thickness is as near as possible to the requested line width—no more than half a pixel different.

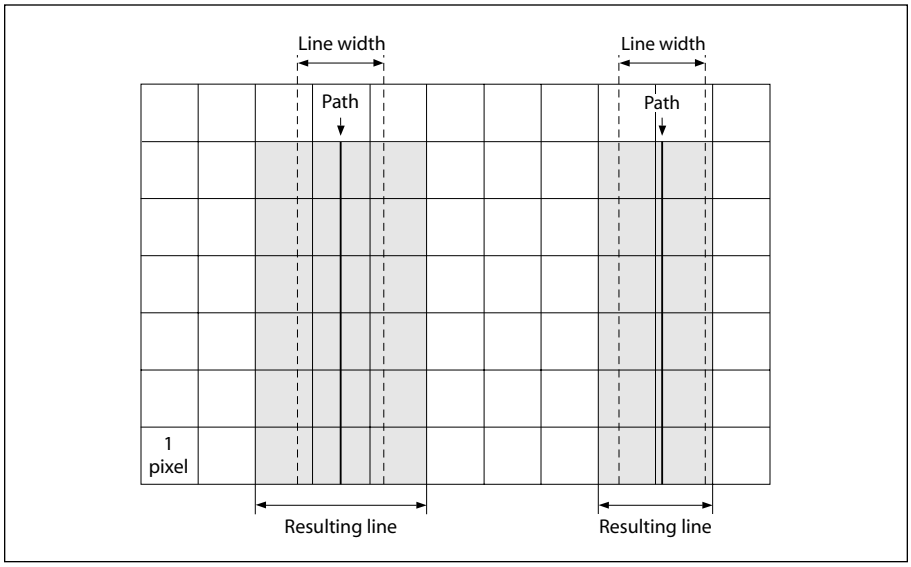

FIGURE 7.6 *Rasterization without stroke adjustment*

Note: *If stroke adjustment is enabled and the requested line width, transformed into device space, is less than half a pixel, the stroke is rendered as a single-pixel line. This is the thinnest line that can be rendered at device resolution. It is equivalent to the effect produced by setting the line width to 0 (see Section 7.5.1, "Scan Conversion Rules").*

Because automatic stroke adjustment can have a substantial effect on the appearance of lines, an application must be able to control whether the adjustment is to be performed. The **setstrokeadjust** operator alters a boolean flag in the graphics state that determines whether stroke adjustment will be performed during subsequent **stroke** and related operations.