



WIKIPEDIA
The Free Encyclopedia

Main page

Contents

Featured content

Current events

Random article

Donate to Wikipedia

Wikipedia store

Interaction

Help

About Wikipedia

Community portal

Recent changes

Contact page

Tools

What links here

Related changes

Upload file

Special pages

Permanent link

Page information

Wikidata item

Cite this page

Print/export

Create a book

Download as PDF

Printable version

Languages

Español

Українська

 Edit links

Article [Talk](#)

[Read](#)

[Edit](#)

[View history](#)

Liang–Barsky algorithm

From Wikipedia, the free encyclopedia

(Redirected from [Liang–Barsky](#))

In [computer graphics](#), the **Liang–Barsky algorithm** (named after [You-Dong Liang](#) and [Brian A. Barsky](#)) is a [line clipping](#) algorithm. The Liang–Barsky algorithm uses the parametric equation of a line and inequalities describing the range of the clipping window to determine the intersections between the line and the clipping window. With these intersections it knows which portion of the line should be drawn. This algorithm is significantly more efficient than [Cohen–Sutherland](#). The idea of the Liang-Barsky clipping algorithm is to do as much testing as possible before computing line intersections.

Consider first the usual parametric form of a straight line:

$$x = x_0 + u(x_1 - x_0) = x_0 + u\Delta x$$

$$y = y_0 + u(y_1 - y_0) = y_0 + u\Delta y$$

A point is in the clip window, if

$$x_{\min} \leq x_0 + u\Delta x \leq x_{\max}$$

and

$$y_{\min} \leq y_0 + u\Delta y \leq y_{\max},$$

which can be expressed as the 4 inequalities

$$up_k \leq q_k, \quad k = 1, 2, 3, 4,$$

where

$$p_1 = -\Delta x, q_1 = x_0 - x_{\min} \text{ (left)}$$

$$p_2 = \Delta x, q_2 = x_{\max} - x_0 \text{ (right)}$$

$$p_3 = -\Delta y, q_3 = y_0 - y_{\min} \text{ (bottom)}$$

$$p_4 = \Delta y, q_4 = y_{\max} - y_0 \text{ (top)}$$

To compute the final line segment:

1. A line parallel to a clipping window edge has $p_k = 0$ for that boundary.
2. If for that k , $q_k < 0$, the line is completely outside and can be eliminated.
3. When $p_k < 0$ the line proceeds outside to inside the clip window and when $p_k > 0$, the line proceeds inside to outside.
4. For nonzero p_k , $u = \frac{q_k}{p_k}$ gives the intersection point.
5. For each line, calculate u_1 and u_2 . For u_1 , look at boundaries for which $p_k < 0$ (i.e. outside to inside). Take u_1 to be the largest among $\left\{0, \frac{q_k}{p_k}\right\}$. For u_2 , look at boundaries for which $p_k > 0$ (i.e. inside to outside). Take u_2 to be the minimum of $\left\{1, \frac{q_k}{p_k}\right\}$. If $u_1 > u_2$, the line is outside and therefore rejected.

See also [\[edit\]](#)

Algorithms used for the same purpose:

- [Cyrus–Beck](#)
- [Nicholl–Lee–Nicholl](#)
- [Fast-clipping](#)

References [\[edit\]](#)

- Liang, Y.D., and Barsky, B., "A New Concept and Method for Line Clipping", *ACM Transactions on Graphics*, 3(1):1-22, January 1984.

- Liang, Y.D., B.A., Barsky, and M. Slater, *Some Improvements to a Parametric Line Clipping Algorithm*, CSD-92-688, Computer Science Division, University of California, Berkeley, 1992.
- James D. Foley. *Computer graphics: principles and practice* [↗](#). Addison-Wesley Professional, 1996. p. 117.

External links [\[edit\]](#)

- <http://hinjang.com/articles/04.html> [↗](#)

Categories: [Clipping \(computer graphics\)](#)

This page was last modified on 12 August 2015, at 06:32.

Text is available under the [Creative Commons Attribution-ShareAlike License](#); additional terms may apply. By using this site, you agree to the [Terms of Use](#) and [Privacy Policy](#). Wikipedia® is a registered trademark of the [Wikimedia Foundation, Inc.](#), a non-profit organization.

[Privacy policy](#) [About Wikipedia](#) [Disclaimers](#) [Contact Wikipedia](#) [Developers](#) [Mobile view](#)

