Software Rendering Characteristics C

XGL provides a software-only implementation of most primitives and attributes. This software implementation may be used in cases when a hardware device cannot process a primitive or attribute. This appendix provides information on some of the properties and limitations of software-only rendering.

Antialiasing

The following notes list characteristics of software rendering of antialiased primitives.

- For all the antialiasing attributes, the software implementation only handles rendering through a 3D Context. The antialiasing attributes are the following:

  ```
  XGL_CTX_LINE_AA_BLEND_EQ
  XGL_CTX_LINE_AA_FILTER_WIDTH
  XGL_CTX_LINE_AA_FILTER_SHAPE

  XGL_CTX_EDGE_AA_BLEND_EQ
  XGL_CTX_EDGE_AA_FILTER_WIDTH
  XGL_CTX_EDGE_AA_FILTER_SHAPE

  XGL_CTX_MARKER_AA_BLEND_EQ
  XGL_CTX_MARKER_AA_FILTER_WIDTH
  XGL_CTX_MARKER_AA_FILTER_SHAPE
  ```

XGL_CTX_STEXT_AA_BLEND_EQ
XGL_CTX_STEXT_AA_FILTER_WIDTH
XGL_CTX_STEXT_AA_FILTER_SHAPE

XGL_CTX_SURF_AA_BLEND_EQ
XGL_CTX_SURF_AA_FILTER_WIDTH
XGL_CTX_SURF_AA_FILTER_SHAPE

- Software antialiasing works when the XGL_DEV_COLOR_TYPE is set to XGL_COLOR_RGB. When the device color type is XGL_COLOR_INDEX, strokes and dot markers are drawn unantialiased.
- Antialiasing smooths out single-pixel-wide edges, lines, markers, text, hollow surfaces, and dots by spreading out the stroke (or dot) over two adjacent pixels. The blurring of the geometry is done by filtering the primitive's pixels onto neighboring pixels. The width of the filter is controlled by the attribute XGL_CTX_{*primitive_type*}_AA_FILTER_WIDTH. This attribute specifies the number of pixels touched by the filter. A hardware device will use the largest filter width it supports that is less than or equal to the specified filter width. The software implementation of antialiasing allows a maximum value of 3 for XGL_CTX_{*primitive_type*}_AA_FILTER_WIDTH. In other words, if this attribute is set to a value greater than 3 by the application, XGL uses the value of 3 for rendering through software.
- If you are running XGL on a device that does not do gamma correction in hardware, then you can specify the gamma correction value to be used by setting the XGL environment variable XGL_AA_GAMMA_VALUE to a float value between 1.0 and 3.0 (both inclusive). Values will be clamped between 1.0 and 3.0. *Note that this value only affects the visual appearance of antialiased strokes and dot markers drawn in software.* Also, XGL attempts to adjust colors of antialiased primitives such that non-antialiased primitive color is close to the apparent color of the antialiased primitive of the same color.

  Internally XGL uses a value of 2.22 for gamma correction, although this value may change in future releases. Setting the environment variable to a value of 1.0 will result in no change in the visual appearance of antialiased primitives on a device that does gamma correction in hardware.

  For general information on gamma correction, see Foley, van Dam, Feiner, and Hughes, *Computer Graphics: Principles and Practice*, Second Edition, Addison-Wesley Publishing Company, 1990.

- When antialiasing is done in software, antialiased stroke primitives (including edges) and antialiased dot markers look best on a TrueColor visual. On a PseudoColor visual, dithering causes some deterioration in the appearance of antialiased primitives.

Transparency

This section provides information on the software rendering of transparent primitives.

- If the `XGL_3D_CTX_SURF_TRANSP_METHOD` is `XGL_TRANSP_BLENDED` and transparency is in effect (which means that the three transparency attributes have values implying transparency) but `XGL_DEV_COLOR_TYPE` is `XGL_COLOR_INDEX`, then the transparency method used in the software implementation of transparency is `XGL_TRANSP_SCREEN_DOOR`.

  If the `XGL_CTX_ROP` is set to anything other than `XGL_ROP_COPY` and transparency is in effect, then the attribute `XGL_TRANSP_BLENDED` is interpreted as `XGL_TRANSP_SCREEN_DOOR`.

- When blended transparency is done in software, primitives rendered with blended transparency looks best on a TrueColor visual. On a PseudoColor visual, dithering causes some deterioration in the appearance of primitives rendered with blended transparency.