GX and GXplus Accelerators 2

The GX and GXplus graphics accelerators accelerate 2D and 3D solid-color polygons, solid and patterned vectors, and 2D and 3D transformations. The GX and GXplus do not have hardware Z-buffers or perform color interpolation. Z-buffered, depth cued, and Gouraud-shaded objects are not accelerated.

The hardware draws triangles and most quadrilaterals directly. More complex polygons must first be subdivided (tessellated) into triangles or quadrilaterals, either by the user or internally by XGL.

Device-Dependent Issues

Accelerated Rendering Using Screen Coordinates (2D Only)

The use of integer screen coordinates on the GX or GXplus results in increased performance, provided all attributes are set to allow acceleration and the following conditions are met:

- `XGL_CTX_VDC_MAP` must be set to `XGL_VDC_MAP_DEVICE` (the default).
- `XGL_CTX_VDC_ORIENTATION` must be set to `XGL_Y_DOWN_Z_AWAY` (the default).
- `XGL_CTX_LOCAL_MODEL_TRANS` and `XGL_CTX_GLOBAL_MODEL_TRANS` must be set to identity (the default).

- When using screen coordinates, the `xgl_multipolyline()` primitive will render substantially faster with these point types: *Xgl_pt_i2d*, *Xgl_pt_color_i2d*, *Xgl_pt_flag_i2d*.

Deferral Mode

Rendering with the GX and GXplus is substantially faster using `XGL_DEFER_ASTI` mode. Setting the deferral mode to `XGL_DEFER_ASAP` disallows certain optimizations and forces XGL to do an internal context post after every primitive.

Backing Store

When using a backing store, rendering is generally slower if the window is partially or fully obscured; it will be *much* slower if something is actually drawn into an obscured portion of the window (since that requires rendering into a memory raster). An optimization for multipolylines allows them to be drawn at highest speed into a partially obscured window if nothing is actually drawn into the obscured portion.

Picking

The GX uses the geometry semantic for picking. (See the man page for a complete description of this semantic.)

Line Join

The `XGL_JOIN_DEVICE` value of the attribute `XGL_CTX_LINE_JOIN` is equivalent to `XGL_JOIN_BEVEL`.

Hardware Double Buffering

The GXplus supports hardware double buffering. Double buffering can also be done on the GX using color map double buffering (see the *XGL Programmer's Guide*), but because the GX has an 8-bit deep frame buffer, the number of displayable colors is limited to 16.

X Visuals

The following Xlib visuals are supported:

- 8-bit StaticGray
- 8-bit GrayScale
- 8-bit StaticColor
- 8-bit PseudoColor (default)

Attributes That Affect Acceleration

This section provides an alphabetical list of the attributes affect the rendering performance of the GX and GXplus accelerators. Check the attribute list to ensure that your application program is not setting attributes that will degrade performance.

In the description of each attribute, the attribute name appears on the left, and the attribute's default, if any, appears to the right. Below the attribute name is a brief description of the attribute with a description of how particular settings enable the accelerator to fully accelerate a primitive or prevent the accelerator from fully accelerating a primitive. Below the description is the list of XGL primitives the attribute affects. For a more complete description of each attribute, consult the man page online or see the *XGL Reference Manual*.

`XGL_CTX_CLIP_PLANES` `NULL` (none enabled)

This attribute defines which of the sizes possible for view-clip planes are enabled. There is no acceleration if one, but not both, of the planes `XGL_CLIP_ZMIN` and `XGL_CLIP_ZMAX` is enabled. Therefore, it is recommended that the `ZMIN` and `ZMAX` planes be either both enabled or both disabled.

Affects all primitives.

`XGL_3D_CTX_DEPTH_CUE_MODE` `XGL_DEPTH_CUE_OFF`

This attribute turns on or off depth cuing for 3D Contexts. There is no acceleration if any value other than `XGL_DEPTH_CUE_OFF` is used.

Affects all primitives.

XGL_3D_CTX_HLHSR_MODE XGL_HLHSR_NONE

This attribute defines the Hidden Line and Hidden Surface Removal method to be used by the application. There is no acceleration if any value other than XGL_HLHSR_NONE is used.

Affects all primitives.

XGL_3D_CTX_LINE_COLOR_INTERP FALSE

This attribute enables or disables interpolation of color between vertices of a polyline. If any value other than the default is used, there is no acceleration.

Affects the following primitive:

```
xgl_multipolyline()
xgl_nurbs_curve()
```

XGL_3D_CTX_SURF_FACE_CULL XGL_CULL_OFF

This attribute specifies the facet-culling mode XGL applies to surfaces. If any value other than the default is used, these primitives are not fully accelerated.

Affects the following primitives:

```
xgl_multi_simple_polygon()
xgl_polygon()
xgl_quadrilateral_mesh
xgl_triangle_strip()
xgl_triangle_list()
xgl_nurbs_surface()
```

XGL_3D_CTX_SURF_FACE_DISTINGUISH FALSE

This attribute determines whether back-facing attributes apply to back-facing surfaces. If any value other than the default is used, these primitives are only partially accelerated.

Affects the following primitives:

```
xgl_multi_simple_polygon()
xgl_polygon()
xgl_quadrilateral_mesh()
xgl_triangle_strip()
xgl_triangle_list()
xgl_nurbs_surface()
```

XGL_3D_CTX_SURF_BACK_FPAT
XGL_CTX_SURF_FRONT_FPAT NULL

These attributes specify the raster containing the fill pattern for stipple-filled surfaces. There is no acceleration for primitives using XGL_CTX_SURF_FRONT_FILL_STYLE of XGL_SURF_FILL_STIPPLE or XGL_SURF_FILL_OPAQUE_STIPPLE.

Affects the following primitives:

xgl_multi_simple_polygon()
xgl_multiarc()
xgl_multicircle()
xgl_multi_elliptical_arc()
xgl_multirectangle()
xgl_polygon()
xgl_quadrilateral_mesh()
xgl_triangle_strip()
xgl_triangle_list()
xgl_nurbs_surface()

XGL_3D_CTX_SURF_FRONT_ILLUMINATION
XGL_3D_CTX_SURF_BACK_ILLUMINATION XGL_ILLUM_NONE

These attributes specify the type of illumination applied to front- and back-facing surfaces respectively. If either attribute is set to XGL_ILLUM_VERTEX or XGL_ILLUM_NONE_INTERP_COLOR, there is no acceleration.

Affects the following primitives:

xgl_multi_simple_polygon()
xgl_polygon()
xgl_quadrilateral_mesh()
xgl_triangle_strip()
xgl_triangle_list()
xgl_nurbs_surface()

XGL_CTX_LINE_STYLE XGL_LINE_SOLID

This attribute specifies the line style (solid, patterned, alt-patterned) for rendering lines and curves. If any value other than the default is used, lines will be slower although still accelerated.

Affects the following primitive:

xgl_multipolyline()
xgl_nurbs_curve()

XGL_CTX_LINE_WIDTH_SCALE_FACTOR 0.0

This attribute specifies the line width of lines being rendered in Device Coordinates. The value is produced by multiplying the specified scale factor by the nominal line width of the device (one pixel).

- 2D lines are accelerated if the value is less than 1.0.
- 3D lines are accelerated if the value is less than or equal to 1.0.
- If a value from 1 to less than 2 is given, 2D and 3D lines are not accelerated.
- If a value of 2 or greater is given, 3D lines are partially accelerated, but 2D lines are not accelerated.

Affects the following primitives:

```
xgl_multipolyline()
xgl_nurbs_curve()
```

XGL_CTX_EDGE_STYLE XGL_LINE_SOLID

This attribute specifies the line style for surface edges. If XGL_CTX_SURF_EDGE_FLAG is set to FALSE, surface edges are accelerated regardless of the value of XGL_CTX_EDGE_STYLE. If XGL_CTX_SURF_EDGE_FLAG is set to TRUE, edges are partially accelerated if XGL_CTX_EDGE_STYLE is set to XGL_LINE_SOLID (the default), but not accelerated if the edge style is set to any value other than solid edges.

Affects the following primitives:

```
xgl_multi_simple_polygon()
xgl_multiarc()
xgl_multicircle()
xgl_multi_elliptical_arc()
xgl_multirectangle()
xgl_polygon()
xgl_quadrilateral_mesh()
xgl_triangle_strip()
xgl_triangle_list()
xgl_nurbs_surface()
```

XGL_CTX_EDGE_WIDTH_SCALE_FACTOR 0.0

This attribute specifies the line width of edges being rendered in Device Coordinates. The value is produced by multiplying the specified scale factor by the nominal line width of the device (one pixel). If a value greater than the default is given, and `XGL_CTX_SURF_EDGE_FLAG` is other than `FALSE` (the default), primitives drawing polygons with edges are not fully accelerated.

Affects the following primitives:

```
xgl_multi_simple_polygon()
xgl_multiarc()
xgl_multicircle()
xgl_multi_elliptical_arc()
xgl_multirectangle()
xgl_polygon()
xgl_quadrilateral_mesh()
xgl_triangle_strip()
xgl_triangle_list()
xgl_nurbs_surface()
```

Reference Table of Primitives and Attributes

Table 2-1 on page 14 lists the primitives that can be tuned to render with increased performance. Note also the following:

- The `xgl_multi_simple_polygon()` primitive is not accelerated for quadrilaterals that are bowtie or V-shaped. There is reduced acceleration if one or more of these facet flags (polygon classification flags) are not set:
  - `XGL_FACET_FLAG_SIDES_ARE_3`
  - `XGL_FACET_FLAG_SIDES_ARE_4`
  - `XGL_FACET_FLAG_SHAPE_CONVEX`
- The `xgl_polygon()` primitive is not accelerated for quadrilaterals that are *bowtie* or *V*-shaped, or for polygons with more than one boundary.
- If a specified transform maps circles to ellipses, lines, or points, circles drawn by `xgl_multicircle()` are not fully accelerated.
- No primitive is accelerated if `XGL_3D_CTX_HLHSR_MODE` is set to any value other than `XGL_HLHSR_NONE`.
- No primitive is accelerated if `XGL_3D_CTX_DEPTH_CUE_MODE` is set to any value other than `XGL_DEPTH_CUE_OFF`.

Table 2-1 Primitives and Related Attributes for the GX

Primitive	Attributes that Affect Acceleration
xgl_annotation_text()	XGL_CTX_CLIP_PLANES XGL_3D_CTX_HLHSR_MODE XGL_3D_CTX_DEPTH_CUE_MODE
xgl_multi_simple_polygon()	XGL_CTX_CLIP_PLANES XGL_3D_CTX_SURF_FACE_CULL XGL_3D_CTX_SURF_FACE_DISTINGUISH XGL_3D_CTX_SURF_BACK_FPAT XGL_CTX_SURF_FRONT_FPAT XGL_3D_CTX_SURF_FRONT_ILLUMINATION XGL_3D_CTX_SURF_BACK_ILLUMINATION XGL_CTX_EDGE_STYLE XGL_CTX_EDGE_WIDTH_SCALE_FACTOR XGL_3D_CTX_HLHSR_MODE XGL_3D_CTX_DEPTH_CUE_MODE
xgl_multiarc()	XGL_CTX_CLIP_PLANES XGL_3D_CTX_SURF_BACK_FPAT XGL_CTX_SURF_FRONT_FPAT XGL_CTX_EDGE_STYLE XGL_CTX_EDGE_WIDTH_SCALE_FACTOR XGL_3D_CTX_HLHSR_MODE XGL_3D_CTX_DEPTH_CUE_MODE
xgl_multicircle()	XGL_CTX_CLIP_PLANES XGL_CTX_EDGE_STYLE XGL_CTX_EDGE_WIDTH_SCALE_FACTOR XGL_3D_CTX_SURF_BACK_FPAT XGL_CTX_SURF_FRONT_FPAT XGL_3D_CTX_HLHSR_MODE XGL_3D_CTX_DEPTH_CUE_MODE
xgl_multi_elliptical_arc()	XGL_CTX_CLIP_PLANES XGL_3D_CTX_SURF_BACK_FPAT XGL_CTX_SURF_FRONT_FPAT XGL_CTX_EDGE_STYLE XGL_CTX_EDGE_WIDTH_SCALE_FACTOR XGL_3D_CTX_HLHSR_MODE XGL_3D_CTX_DEPTH_CUE_MODE

Table 2-1 Primitives and Related Attributes for the GX (Continued)

Primitive	Attributes that Affect Acceleration
xgl_multimarker()	XGL_CTX_CLIP_PLANES XGL_3D_CTX_HLHSR_MODE XGL_3D_CTX_DEPTH_CUE_MODE
xgl_multipolyline()	XGL_CTX_CLIP_PLANES XGL_3D_CTX_LINE_COLOR_INTERP XGL_CTX_LINE_STYLE XGL_CTX_LINE_WIDTH_SCALE_FACTOR XGL_3D_CTX_HLHSR_MODE XGL_3D_CTX_DEPTH_CUE_MODE
xgl_multirectangle()	XGL_CTX_CLIP_PLANES XGL_3D_CTX_SURF_BACK_FPAT XGL_CTX_SURF_FRONT_FPAT XGL_CTX_EDGE_STYLE XGL_CTX_EDGE_WIDTH_SCALE_FACTOR XGL_3D_CTX_HLHSR_MODE XGL_3D_CTX_DEPTH_CUE_MODE
xgl_nurbs_curve()	XGL_CTX_CLIP_PLANES XGL_3D_CTX_LINE_COLOR_INTERP XGL_CTX_LINE_STYLE XGL_CTX_LINE_WIDTH_SCALE_FACTOR XGL_3D_CTX_HLHSR_MODE XGL_3D_CTX_DEPTH_CUE_MODE Note that NURBS curves are tessellated into polylines and rendered with the polyline attributes. The NURBS curve attributes also affect performance in that they affect the decomposition of curves into polylines.

Table 2-1 Primitives and Related Attributes for the GX (Continued)

Primitive	Attributes that Affect Acceleration
xgl_nurbs_surface()	XGL_CTX_CLIP_PLANES XGL_3D_CTX_SURF_FACE_CULL XGL_3D_CTX_SURF_FACE_DISTINGUISH XGL_3D_CTX_SURF_BACK_FPAT XGL_CTX_SURF_FRONT_FPAT XGL_3D_CTX_SURF_FRONT_ILLUMINATION XGL_3D_CTX_SURF_BACK_ILLUMINATION XGL_CTX_EDGE_STYLE XGL_CTX_EDGE_WIDTH_SCALE_FACTOR XGL_3D_CTX_HLHSR_MODE XGL_3D_CTX_DEPTH_CUE_MODE Note that NURBS surfaces are tessellated into triangle strips and rendered with the triangle strip attributes. The NURBS surface attributes also affect performance in that they affect the decomposition of surfaces into triangle strips.
xgl_polygon()	XGL_CTX_CLIP_PLANES XGL_3D_CTX_SURF_FACE_CULL XGL_3D_CTX_SURF_FACE_DISTINGUISH XGL_3D_CTX_SURF_BACK_FPAT XGL_CTX_SURF_FRONT_FPAT XGL_3D_CTX_SURF_FRONT_ILLUMINATION XGL_3D_CTX_SURF_BACK_ILLUMINATION XGL_CTX_EDGE_STYLE XGL_CTX_EDGE_WIDTH_SCALE_FACTOR XGL_3D_CTX_HLHSR_MODE XGL_3D_CTX_DEPTH_CUE_MODE
xgl_quadrilateral_mesh()	XGL_CTX_CLIP_PLANES XGL_3D_CTX_SURF_FACE_CULL XGL_3D_CTX_SURF_FACE_DISTINGUISH XGL_3D_CTX_SURF_BACK_FPAT XGL_CTX_SURF_FRONT_FPAT XGL_3D_CTX_SURF_FRONT_ILLUMINATION XGL_3D_CTX_SURF_BACK_ILLUMINATION XGL_CTX_EDGE_STYLE XGL_CTX_EDGE_WIDTH_SCALE_FACTOR XGL_3D_CTX_HLHSR_MODE XGL_3D_CTX_DEPTH_CUE_MODE

Table 2-1 Primitives and Related Attributes for the GX (Continued)

Primitive	Attributes that Affect Acceleration
xgl_stroke_text()	XGL_CTX_CLIP_PLANES XGL_3D_CTX_HLHSR_MODE XGL_3D_CTX_DEPTH_CUE_MODE
xgl_triangle_list()	Currently not accelerated.
xgl_triangle_strip()	XGL_CTX_CLIP_PLANES XGL_3D_CTX_SURF_FACE_CULL XGL_3D_CTX_SURF_FACE_DISTINGUISH XGL_3D_CTX_SURF_BACK_FPAT XGL_CTX_SURF_FRONT_FPAT XGL_3D_CTX_SURF_FRONT_ILLUMINATION XGL_3D_CTX_SURF_BACK_ILLUMINATION XGL_CTX_EDGE_STYLE XGL_CTX_EDGE_WIDTH_SCALE_FACTOR XGL_3D_CTX_HLHSR_MODE XGL_3D_CTX_DEPTH_CUE_MODE