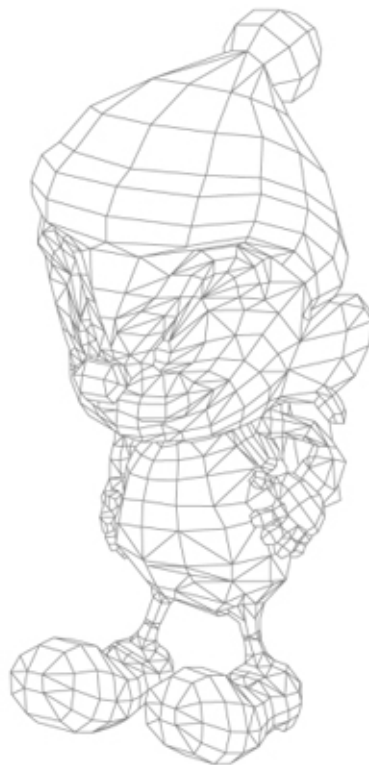


MascotCapsule Renderion

White Paper

English Manual



Ver. 1.1



©2009 HI CORPORATION. All Rights Reserved.

- Table of Contents -

1.. Product presentation.....	1
1.1. Renderion Components.....	1
1.1.1. EGL implementation.....	1
1.1.2. OpenGL ES implementation:	2
2.. Product Specification.....	3
2.1. Data Sheet.....	3
3.. Reference Materials.....	4

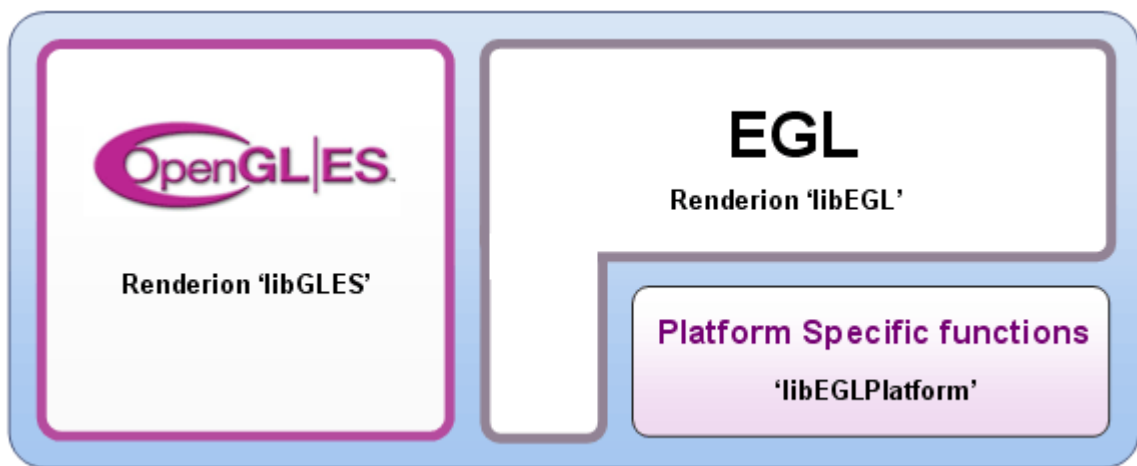
1. Product presentation

MascotCapsule® Renderion is a complete software implementation of OpenGL® ES, industry standard for embedded 3D Graphics API managed by the Khronos Group, Inc. Renderion conforms to OpenGL ES 1.1 API.

1.1. Renderion Components

From API point of view, Renderion consists of two modules;

- The EGL implementation represents an Interface between OpenGL ES and the native window system.
- The OpenGL ES implementation is responsible for the drawing operations.



MascotCapsule Renderion Components

1.1.1. EGL implementation

Renderion's EGL implementation is separated in to parts:

- The libEGL part implements the EGL APIs as described in the EGL specifications.
- The libEGLPlatform part implements platform specific functionalities used by Renderion's libEGL.

It is important to note that, from an application point of view, there are just two libraries: libGLES and libEGL. The application does not need to call or link to libEGLPlatform.

The libEGLPlatform contains:

- Memory management functions: memory allocation, etc.
- Rendering Surface management: window surface creation, pixmap, etc.

The libEGLPlatform library's source code is freely available. This offers the flexibility to port Renderion to any platform and window-system.

Two platforms are currently implemented:

- Windows / Windows Mobile
- Linux

1.1.2. OpenGL ES implementation:

Renderion's rendering pipeline is composed by:

- Transform and Lighting engine: performs vertex and coordinates transformations keeping the maximum accuracy while delivering the best performance.

The engine relies on a highly optimized float library to provide accuracy for every range of values. A smart vertex-cache algorithm ensures that each vertex do not get transformed and lit several times.

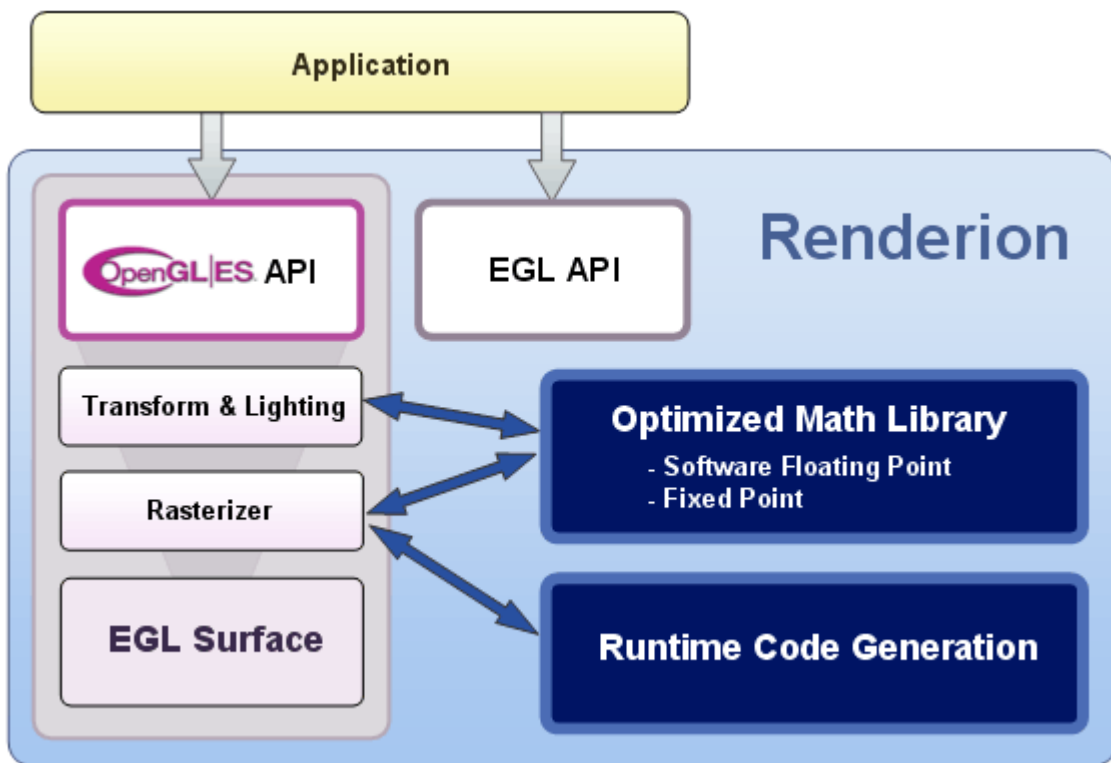
- Rasterization engine: is able to rasterize triangles using incoming data from the Transform and Lighting engine. The rasterizer uses only fixed-point arithmetic and is therefore able to run on low-end processors at the best speed.

In term of quality, color calculations are always performed with the best accuracy.

Perspective-correction is applied on every interpolated component: texture-coordinates, colors and fog to deliver the most coherent output.

Dithering is also available and can be used to avoid the "banding" effect; generally visible with color gradients. It is possible to get appealing results even on low color depth displays (e.g. 12bits display).

The efficiency of the rasterization is allowed by the use of a Runtime Code Generation framework able to output code of the same quality than hand-written assembler.



MascotCapsule Renderion Components

2. Product Specification

2.1. Data Sheet

Supported APIs	OpenGL ES 1.1 Common EGL 1.4
Processors	ARM V5 or later
Operating System	Microsoft Windows Mobile: PocketPC2003, Windows Mobile 5.x, 6.x Linux (kernel 2.6.x)
Maximum Resolution	1024 x 1024
Supported Color Formats	Window, Pixmap and Client-Pixmap(extension) surface: RGB565 RGBA4444 RGBA5551 RGBA8 ARGB8 Pbuffer Surface: RGB565, RGBA4444, RGBA5551, RGBA8
Supported Depth Buffer Formats	16 and 24 Bits per pixel
Supported Stencil Buffer Format	8 Bits per pixel
Code / ROM / Heap usage	170K byte / 180K byte / 60K byte (excluding EGL surfaces)
Number of Lights	8
Number of User Clip-Planes	6
Model-View stack depth	16
Projection & Texture stack depth	2
Number of Sub-pixels bits	4
Number of Texture Units	2
Maximum Texture resolution	1024 x 1024
Non power-of-two textures	Not supported
Multisampling	Not supported
Supported OpenGL ES / EGL Extensions	<ul style="list-style-type: none"> • GL_OES_byte_coordinates • GL_OES_fixed_point • GL_OES_single_precision • GL_OES_matrix_get • GL_OES_read_format • GL_OES_compressed_paletted_texture • GL_OES_point_size_array • GL_OES_point_sprite • GL_OES_query_matrix • GL_OES_draw_texture

	<ul style="list-style-type: none">• EGL_KHR_lock_surface• EGL_HI_clientpixmap (proprietary extension)• EGL_HI_colorformats (proprietary extension)
--	--

3. Reference Materials

"OpenGL ES Common/Common-Lite Profile Specification Version 1.1.12 (Full Specification)" April 24, 2008

"Khronos Native Platform Graphics Interface (EGL Version 1.4)" May 12, 2010

MascotCapsule Renderion
White Paper **Ver. 1.1**

English Manual

Version 1.1
Publication Date 6/3/2010
Publisher HI CORPORATION
Meguro Higashiyama Bldg. 5F, 1-4-4, Higashiyama,
Meguro-ku, Tokyo 153-0043, Japan
<http://www.hicorp.co.jp/>

-Trademarks, Copyrights & Disclaimer-

- This document is copyright law protected. Neither the whole nor any part of the information (including software and program) contained in this document may be reproduced or reprinted in any material form except with the written permission of HI CORPORATION.
- MascotCapsule(R) is the registered trademark of HI CORPORATION in Japan. All brand names and product names in this document are trademarks or registered trademarks of their respective companies.
- HI CORPORATION is not responsible for any problem, loss or damages that may be caused by utilizing the materials or information in this document.
- The contents of this document are subject to change for revision without prior notice.

©2010 HI CORPORATION. All Rights Reserved.