

# ZOOM OUT BREW 2008 CONFERENCE

**OpenVG: Enabling Rich Graphics and  
Applications on BREW® Devices**

Brent Sammons, Product Manager

Zack Zhou, Staff Engineering Manager

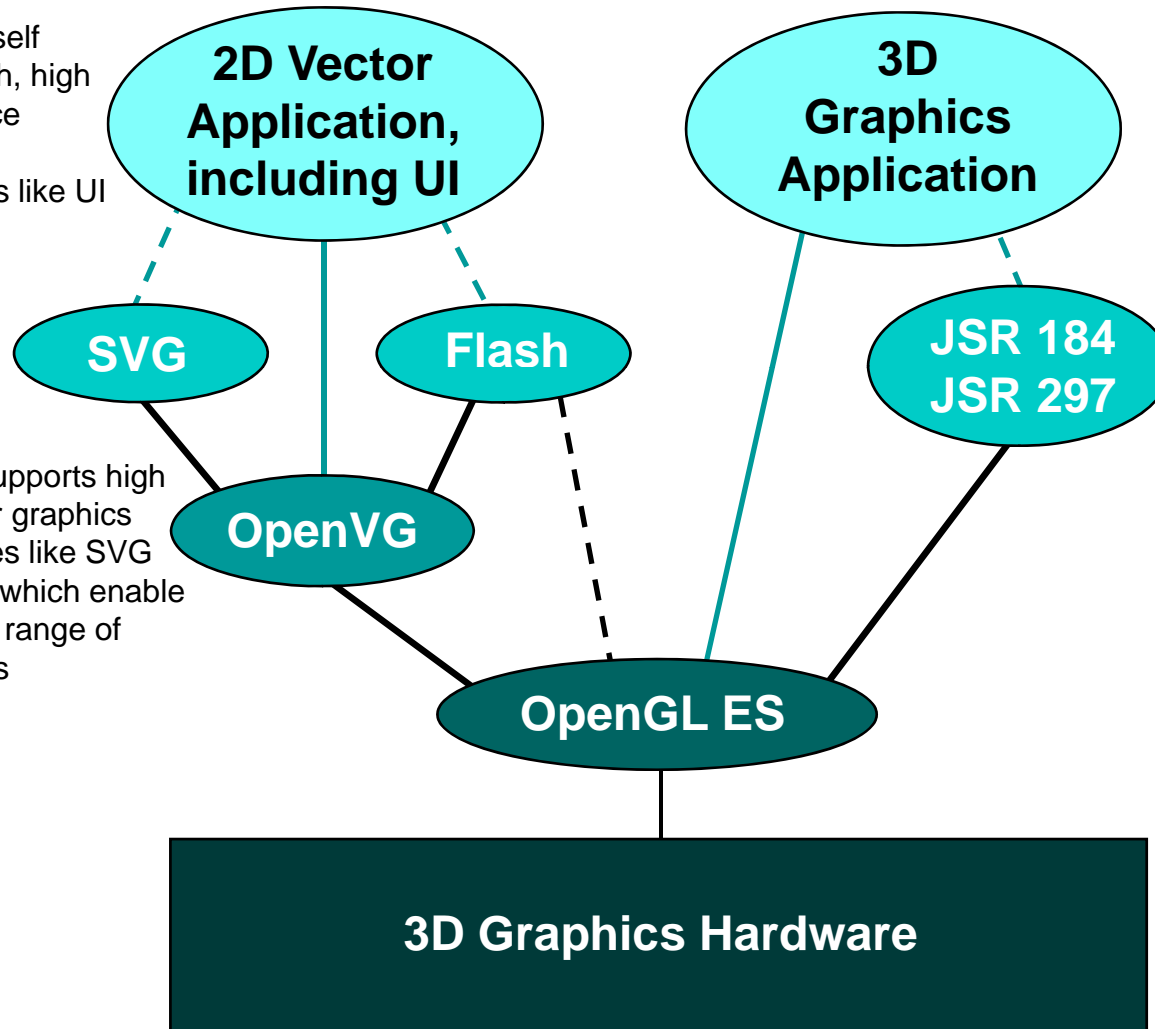
Qualcomm Incorporated

**QUALCOMM**

# Qualcomm® Graphics API Hierarchy

OpenVG itself enables rich, high performance graphical applications like UI and games

OpenVG supports high level vector graphics technologies like SVG and Flash, which enable a yet wider range of applications



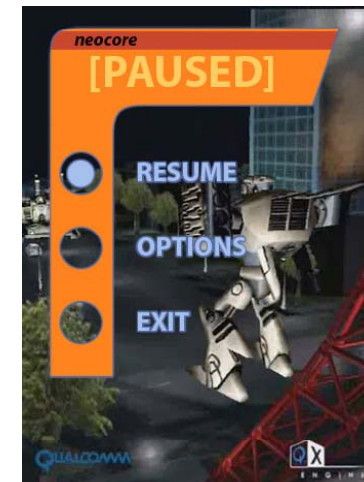
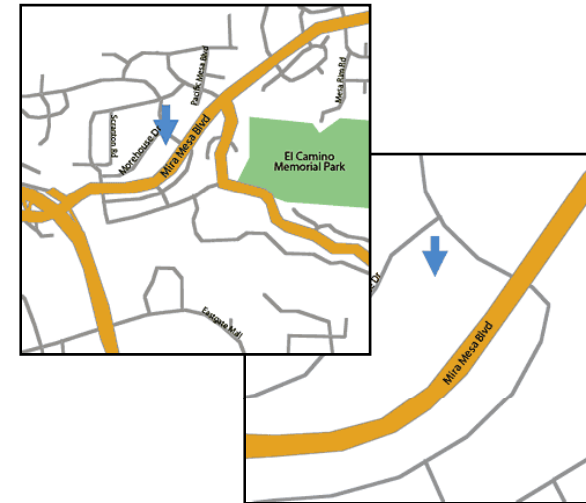
## OPENVG

- Open, industry standard API for Vector Graphics acceleration
- Hardware abstraction layer to accelerate SVG & Flash
- Enables concurrent use of 2D graphics with other formats and functions
- Designed to work with OpenGL ES
- Provides fast interactive performance at lower power levels
- Dramatically improve user experience

# OpenVG Use Cases



- Enables more robust content
  - Robust, dynamic UIs
  - SVG Viewers
  - Portable Mapping Applications
    - Dynamic features for map display: dynamic placement and sizing of street names and markers, efficient viewport culling
  - E-book Readers
    - Fast rendering of clear text in Western, Asian, and other scripts
  - Games
    - Useful for defining sprites, backgrounds, and textures for use in both 2D and 3D games
  - Provide two-dimensional overlays (e.g., for UI, maps or game scores) on top of 3D content
  - Low-Level Graphics Device Interface
    - May be used as a low-level graphics device interface
    - Other graphical toolkits, such as windowing systems, may be implemented above OpenVG





## OpenVG Going Strong

- OpenVG shipping on Qualcomm® chipsets as of February 2008
- All chipsets with OpenGL ES will have OpenVG
  - CDMA, UMTS and Worldmode chipsets
  - AMSS, WinMobile and Linux platforms
- OpenVG being mandated widely
- SVG acceleration on OpenVG by Q3 2008
- First devices with Qualcomm's OpenVG solution launching in 2008
- UI and acceleration of other 2D Vector formats are primary use cases

# OpenVG Integration & Implementation

Qualcomm's OpenVG solution is simple and flexible for OEMs and Developers to use

Qualcomm Provides:	OEM can:
Fully implemented OpenVG solution on multiple Operating Systems; AMSS, WinMobile and Linux	Enable OpenVG with no further integration effort
EGL wrapper	Use their own EGL implementations
Memory management hooks for OpenVG and EGL in source code	Plug in their own memory management systems
Platform-independent layer on OpenVG	Port OpenVG on different platforms based on our reference implementation
Font engine inside of OpenVG as well framework to integrate external font engine	Integrate font engine of choice
EGL extensions*	Enable more OpenVG-related features*

- Some EGL extensions can be used by OEMs and application developers to enable capabilities like overlay, scale-up and rotation
- Example Use case: OEM has background display video – animated character or text on top
  - Render small screen and scale to full screen
  - Pop-up menus



# Advantages of SVG & Flash<sup>®</sup> on OpenVG

- More efficient CPU utilization
  - Vector graphics in Software ~ 70% CPU
  - Hardware ~ 30% CPU
- Richer graphics can be enabled
  - Textures can be accelerated by hardware
- Support for higher density / higher resolution displays
- Significantly faster frame rate
- Dramatically improved user experience



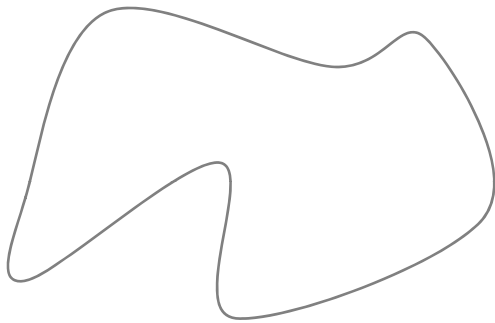
# OpenVG Performance & Power

- OpenVG hardware acceleration for SVG-T provides:
  - Up to 10x greater frame rate performance improvement over software-only rendering
  - 30 - 45% reduction in power consumption over software-only rendering
- Performance varies across MSMs
  - We keep consistent performance matrix on same tier chipset (6K, 7K, etc.)
  - We fully utilize the hardware on all chipset tiers
    - We utilize each hardware's unique features such as Stencil Buffer and Image Filter
  - We maximize performance for each platform
  - Anti-aliasing supported as of 8K shader-based solution
- Animation
  - We do dynamic re-tessellation based on scale factor
  - It provides a better balance of performance and quality

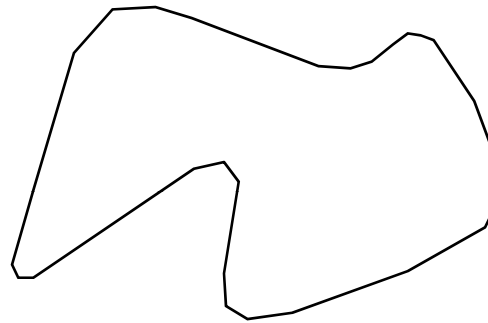
# OpenVG Rendering

- At the time a path is filled, transformation may be applied

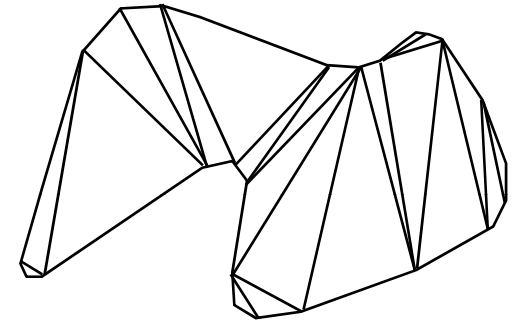
Implicit curve



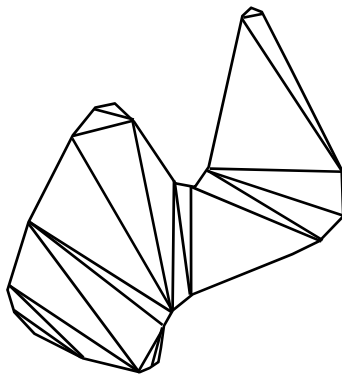
Tessellation



Triangulation



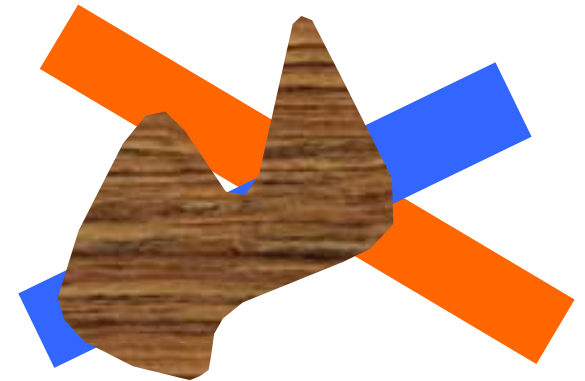
Transform as tristrip



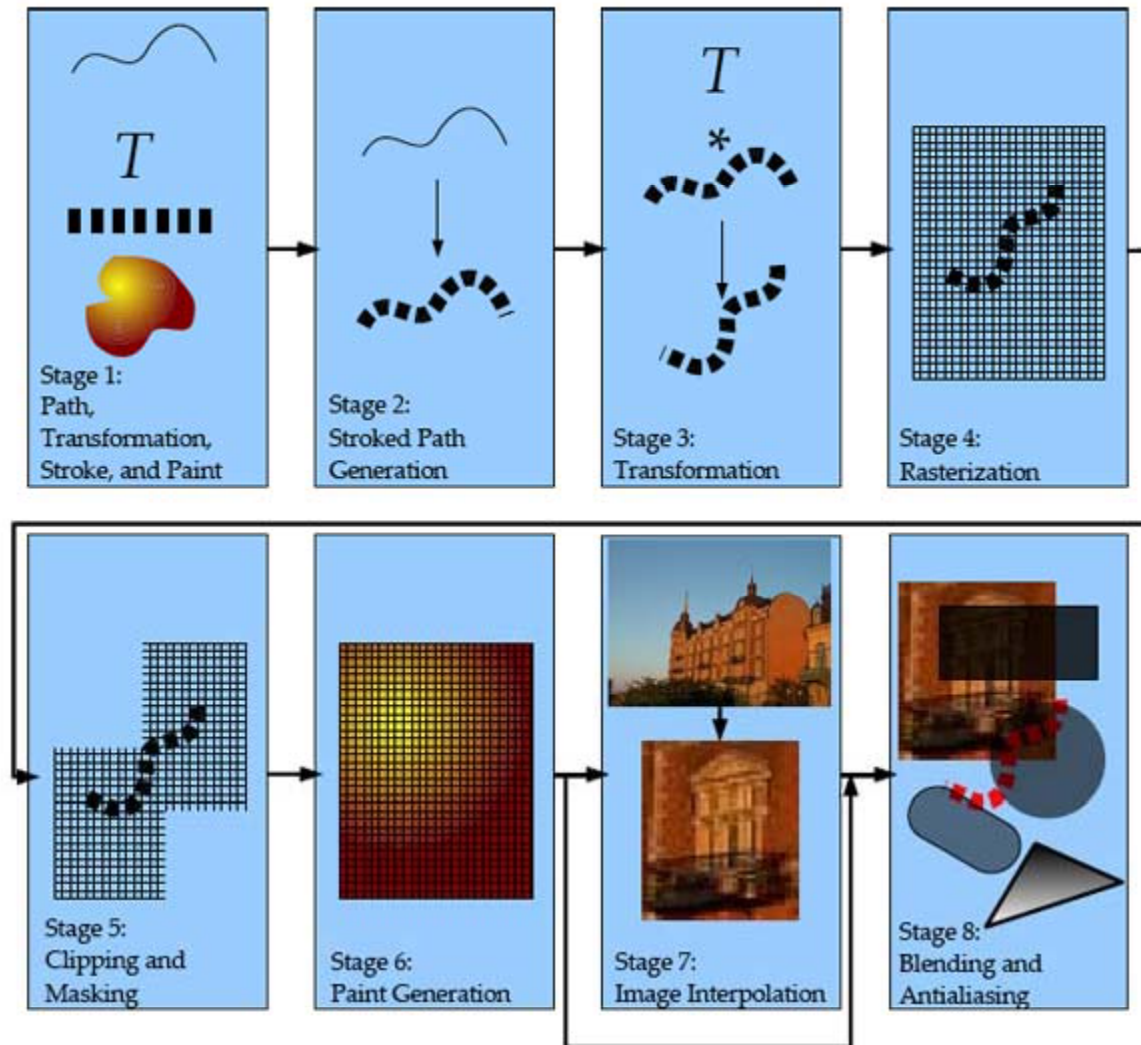
Shaded Fill



Pixel Blend



# OpenVG Pipeline



\* Image from W3C site

# OpenVG Features: Hardware or Software?

Stage 1: Path, transformation, stroke, paint	
Application Operations	Software
Stage 2: Stroked Path Generation	
Path stroking and filling	Hardware
Text stroking and filling	Hardware
Fill rule	Software
Stage 3: Transformation	
Path, Image, Paint transforms	Hardware
Glyph transforms	Hardware
Stage 4: Rasterization	
Coverage values	Hardware
Antialiasing	Hardware
Stage 5: Clipping and Masking	
Clipping	Hardware
Scissoring	Hardware
Masking	Hardware
Stage 6: Paint Generation	
Plain color paint	Hardware
Linear gradient paint	Hardware
Radial gradient paint	Hardware
Color ramps	Hardware
Pattern paint	Hardware

Stage 7: Image Interpolation	
Draw image – normal	Hardware
Draw image – multiply	Hardware
Draw image – stencil	Hardware
Text as images	Hardware
Stage 8: Color Transform, Blending, Antialiasing	
Color transformation	Software
Porter-Duff blending	Hardware
Blend multiply, screen, darken, lighten	Hardware
Additive blending	Hardware
Image Operations (Outside Main Pipeline)	
Pixel operations	Hardware
Sub-data operations	Hardware
Child images	Software
Format normalization	Software
Color combination	Software
Separable convolution	Software
General convolution	Software
Gaussian blur	Software
Lookup	Software
Tiled fill	Hardware
Fast clear	Hardware



# OpenVG for Developers

- OpenVG is part of the Khronos ecosystem, which also includes OpenGL ES, EGL, etc.
- OpenVG is a standard API, much like OpenGL ES
- Many developers are already familiar with OpenGL ES
- OpenVG integrates with BREW, though it is not BREW-specific
- OpenVG enables developers to directly access hardware
- OpenVG is a pure graphics API
  - Can work with other low level multimedia technologies
    - 3D application, video, etc.
  - Can work as a compositing engine
  - Can work with any other display application with overlay



# OpenVG for Developers

- Qualcomm will provide:
  - Hardware accelerated solution on top of existing OpenGL ES hardware
    - For AMSS, Windows and Linux 6K and 7K platforms
  - Consistent development environment across performance tiers
  - OpenVG BREW extension as part of the BREW SDK®
  - OpenVG sample application for various platforms
  - OpenVG Developer Guide
  - BREW-style C-only template for creating OpenVG applications