Accessing PowerVR 2DC Features Under Windows® CE

Jason Powell

GDC - March 18, 1999

www.powervr.com

Talk Overview

- PowerVR Chip Features
- Windows CE for PowerVR-2DC
- Enabling PowerVR Features

PowerVR-2DC

- Graphics chip for the console.
- Chip Features
  - Renders triangles, triangle strips, and quadrilaterals.
  - Culling of back faces and tiny polygons.
  - ARGB Gouraud shading, flat shading.
  - Specular highlighting.
  - Perspective-correct mip-mapped texturing.
  - Texture clamping, wrapping, and mirroring.
  - Bilinear, trilinear, and anisotropic filtering.
  - Full-scene anti-aliasing.
  - Vertex fog as well as all table fog modes.

PowerVR-2DC

- Features Continued...
  - Per-pixel translucency sorting.
  - 32-bit on-chip z-buffering.
  - Hardware clipping to a viewport.
  - 640x480x24 maximum resolution.
  - RGBA 5650, 5551 and 4444 texture formats.
  - YUV 422, 420 texture formats.
  - 8bpp and 4bpp palletized texture formats.
  - Bump mapping.
  - VQ texture compression.
  - Scene capture architecture.

Windows CE for PowerVR-2DC

- Windows API compatible
- Windows 32-based
- DirectX™ 5.0
  - D3D IM, DDraw, DSound, DInput, DPlay, Dshow
  - Developer Studio™ 97 IDE
- VC++ 5.0
- Priorities on performance and portability

Punch-Through

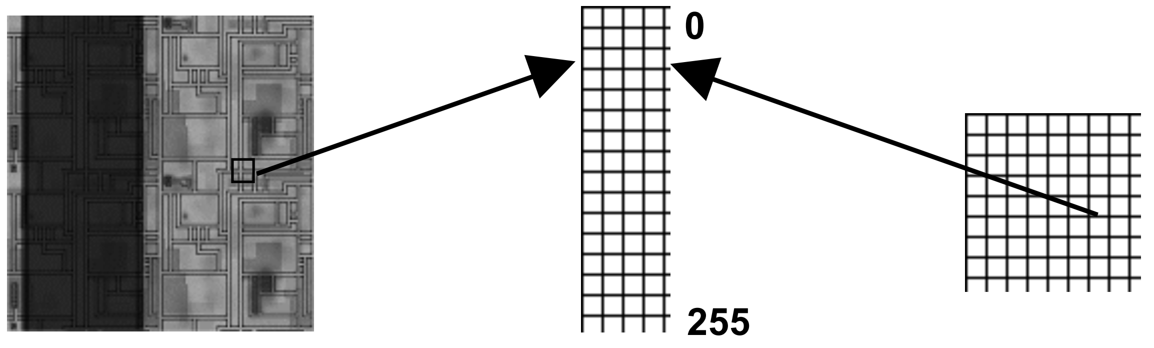
- Really a color key mode.
- A new render pass in chip to increase color key performance.
- First opaque, punch through, then translucent pass.
- Punch through updates z *only* when texel is opaque.
- Performance should fall between opaque and translucent performance.

Punch-Through

- To create a punch through surface….
  - Option 1
    - Create 1555 surface.
    - Set ALPHABLENDENABLE to TRUE.
  - Option 2
    - Create 8 or 4 bit palettized surface.
    - Set COLORKEYENABLE to TRUE.
  - Will not work when…
    - BLENDMODE = MODULATEALPHA, DECALALPHA

Texture Compression

- Vector Quantization (VQ)
  - 7:1 compression ratio.
  - "Lossiness" depends on the size and noise of the texture compressed.



Compression achieved when 2x2 blocks refer to same entry in codebook.

256x256 Texture is divided into 2x2 pixel blocks

Full RGBA of all 4 pixels are written to codebook

Resulting "texture" is a 128x128 array of indices into it's codebook

Texture Compression

- Larger textures compress better, but are “lossier”.
- Diminishing returns at 64x64.
- Compression ratios...

	Original Texture Size	Compressed Texture Size	Codebook Size (bytes)	Compression Ratio (1:n)
32x32	2048	256	2048	0.89
64x64	8192	1024	2048	2.67
128x128	32768	4096	2048	5.33
256x256	131072	16384	2048	7.11
512x512	524288	65536	2048	7.76
1024x1024	2097152	262144	2048	7.94

Texture Compression

- Texture compression done at author time.
  - PVRCONV.EXE converts .bmp's to VQ format.
- To create a VQ surface…
  - Use PVRCONV to create your artwork.
  - CreateSurface…
    - ddpfPixelFormat.dwFlags |= DDPF_COMPRESSED.
    - Width & height = dimensions of *uncompressed* texture.
  - Copy VQ bytes directly to texture.

Bump Mapping

- Requires a two pass render.
- Bump map contained in second texture.
- Each texel (aka buxel) is a vector which is normal to the “bump”.

Bump Mapping

- Additional light states…
  - D3DLIGHTSTATE_BUMPINTENSITY
    - The strength of the highlight.
    - Type float.
  - D3DLIGHTSTATE_BUMPDIRECTION
    - The light’s direction within the scene.
    - Type D3DVECTOR.
  - D3DLIGHTSTATE_BUMPAMBIENT
    - The light color.
    - Type D3DVECTOR.

Bump Mapping

- To create a bump map surface…
  - Use FOURCC code for CreateSurface.
    - #define FOURCC_PNBM MAKEFOURCC('P', 'N', 'B', 'M')
    - ddsd.ddpfPixelFormat.dwFourCC = FOURCC_PNBM;
    - Create surface in system memory.
  - Fill surface with bump map pixels.
    - Easiest way is to use HeightToBump_Wrap function in WinCE SDK.
    - Converts a height map into K1K2K3Q pixel (aka buxel) format.
  - Create video memory surface and load from system surface.

Bump Mapping

- Rendering the bump map surface…
  Pass 1: Bump texture pass
  - Set Z compare mode to LESS_EQUAL.
  - Set BUMPINTENSITY, BUMPDIRECTION, and BUMPAMBIENT light states.
  - Set texture blend modes…
    - SRCBLEND = ONE
    - DESTBLEND = ZERO
    - TEXTUREMAPBLEND = DECALALPHA
    - ALPHABLENDENABLE = FALSE
  - Render all bump map surfaces.

Bump Mapping

- **Rendering the bump map surface…**
  Pass 2: Base texture pass
  - **Set texture blend modes…**
    - **SRCBLEND = DESTCOLOR**
    - **DESTBLEND = ZERO**
    - **TEXTUREMAPBLEND = MODULATEALPHA**
    - **ALPHABLENDENABLE = TRUE**
  - **Render all base textures.**

Optimized Textures

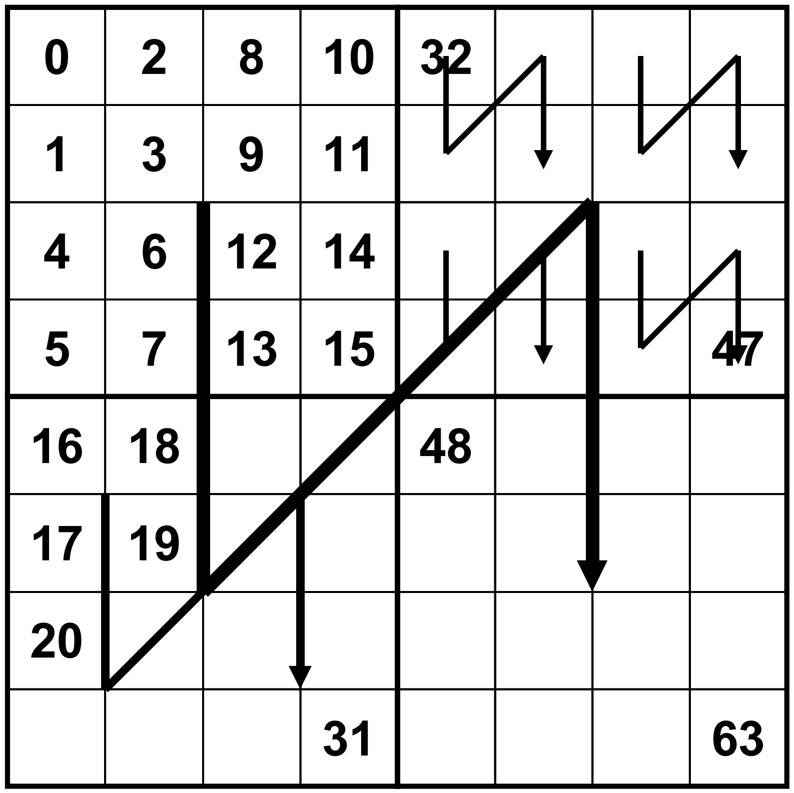
- Non-linear video memory format.
- Also referred to as “twiddled”.
- Slight performance hit on texture load.
- Bilinear filtering becomes free.

Optimized Textures

- To create an optimized surface…
  - Option 1:
    - Create system memory surface (non-optimized).
    - Load image data (ie. bitmap) onto system memory surface.
    - Create video memory texture surface with ddsd.ddsCaps.dwCaps |= DDSCAPS_ALLOCONLOAD
    - Load system memory surface onto video memory surface.
    - Driver automatically optimizes (“twiddles”) texture on load.
  - Option 2:
    - Create video memory texture surface with ddsd.ddsCaps.dwCaps |= DDSCAPS_ALLOCONLOAD
    - Write to texture memory directly in optimized (“twiddled”) format.

Writing to Texture Memory

- Texture memory is linear except optimized/twiddled format.
- Twiddled format is…



- Bits are ordered in a set of reverse “N”s.
- Bits 0-15 represent a texel, 16-31 the next texel, etc.
- Supports all texture formats.

That’s All Folks

- Questions

- For additional information…
  - Visit Sega, Microsoft, NEC on show floor at booth #808.
  - Sign on with Sega as a Dreamcast Developer!
    - Call the New Developer Hotline at 415-701-7070.