ETC2-PACKAGE

TEXTURE COMPRESSION CODECS MANDATORY IN OPENGL ES 3.0 AND OPENGL 4.3

JACOB STROM, ERICSSON RESEARCH
QUICK SUMMARY

› New codecs that provide
  – Higher Quality
  – Alpha (1bit and full alpha)
  – R- and RG-textures

› Mandatory in OpenGL ES 3.0 / OpenGL 4.3
  – Finally mandatory texture compression in OpenGL/OpenGL ES!

same bit rate!
Problems:
- Memory can get full
- Bus can get congested (performance bottleneck)
WHY TEXTURE COMPRESSION?

Benefits:

- More textures fit in memory
- Less traffic on bus = higher performance
- Less traffic on bus = lower power consumption
- Cheaper to transmit textures over networks
HISTORY

› ETC1 was standardized in OpenGL ES
› ETC1 is supported in Android with v2.2 (Froyo) and above, which means over 370 million devices.
› However, it was never mandatory in OpenGL ES 2.0, so developers would have to check if it was there.
› Also, it lacked support for alpha and RG-textures.
ETC2 – NEW VERSION

› ETC2 is an updated version of ETC1
› Same bitrate as ETC1 – but enhanced quality
   – Now about 1.0 dB better than ETC1
   – About 0.8 dB better than S3TC/DXTC.
   – About 1.6 dB better than PVR-TC
QUALITY COMPARISON

<table>
<thead>
<tr>
<th>Original</th>
<th>S3TC/DXTC</th>
<th>ETC1</th>
<th>ETC2</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.jpg" alt="Original Image" /></td>
<td><img src="image2.jpg" alt="S3TC/DXTC Image" /></td>
<td><img src="image3.jpg" alt="ETC1 Image" /></td>
<td><img src="image4.jpg" alt="ETC2 Image" /></td>
</tr>
<tr>
<td><img src="image5.jpg" alt="Original Image" /></td>
<td><img src="image6.jpg" alt="S3TC/DXTC Image" /></td>
<td><img src="image7.jpg" alt="ETC1 Image" /></td>
<td><img src="image8.jpg" alt="ETC2 Image" /></td>
</tr>
<tr>
<td><img src="image9.jpg" alt="Original Image" /></td>
<td><img src="image10.jpg" alt="S3TC/DXTC Image" /></td>
<td><img src="image11.jpg" alt="ETC1 Image" /></td>
<td><img src="image12.jpg" alt="ETC2 Image" /></td>
</tr>
</tbody>
</table>
QUALITY COMPARISON

<table>
<thead>
<tr>
<th></th>
<th>original</th>
<th>S3TC/DXTC</th>
<th>ETC1</th>
<th>ETC2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parrot</td>
<td><img src="original_parrot.png" alt="Image" /></td>
<td><img src="s3tc_dxtc_parrot.png" alt="Image" /></td>
<td><img src="etc1_parrot.png" alt="Image" /></td>
<td><img src="etc2_parrot.png" alt="Image" /></td>
</tr>
<tr>
<td>Flower</td>
<td><img src="original_flower.png" alt="Image" /></td>
<td><img src="s3tc_dxtc_flower.png" alt="Image" /></td>
<td><img src="etc1_flower.png" alt="Image" /></td>
<td><img src="etc2_flower.png" alt="Image" /></td>
</tr>
</tbody>
</table>
LIGHT MAPS

original

S3TC

ETC2
ETC2 is backward compatible with ETC1: If you have an old ETC1 texture, you can load it as an ETC2 texture and the hardware will decode it correctly.

This means, that if you are developing content, if you create an ETC1 texture, it will be able to run on:
- All OpenGL ES 3.0 devices
- All OpenGL 4.3 devices
- All of the 370 million Android handsets that can handle ETC1

Also means ETC2 can never be worse than ETC1 even for a single block of data
ETC1 and ETC2 textures share a lot of blocks.

It is possible to store an ETC1 textures (for legacy devices) and ETC2 textures (for OpenGL ES 3.0 devices) in a single file.
FLAVORS OF ETC2

- ETC1 could only handle RGB textures
- ETC2 comes in several flavours:

<table>
<thead>
<tr>
<th></th>
<th>RGB</th>
<th>sRGB</th>
</tr>
</thead>
<tbody>
<tr>
<td>RGB only (4 bpp)</td>
<td>RGB8_ETC2</td>
<td>SRGB8_ETC2</td>
</tr>
<tr>
<td>RGBA (8 bpp)</td>
<td>RGBA8_ETC2_EAC</td>
<td>SRGB8_ALPHA8_ETC2_EAC</td>
</tr>
<tr>
<td>RGB punchthrough A (4 bpp)</td>
<td>RGB8_PUNCHTHROUGH_ALPHA1_ETC2</td>
<td>SRGB8_PUNCHTHROUGH_ALPHA1_ETC2</td>
</tr>
</tbody>
</table>
EAC

- ETC1 could only handle RGB textures
- Two-channel (RG) textures needed for normal maps.
- EAC gives high-quality one-channel (R) and two-channel (RG) channel textures
- Signed version can preserve 0 exactly

<table>
<thead>
<tr>
<th></th>
<th>unsigned</th>
<th>signed</th>
</tr>
</thead>
<tbody>
<tr>
<td>one channel data (R, 4 bpp)</td>
<td>R11_EAC</td>
<td>SIGNED_R11_EAC</td>
</tr>
<tr>
<td>two channel data (RG, 8 bpp)</td>
<td>RG11_EAC</td>
<td>SIGNED_RG11_EAC</td>
</tr>
</tbody>
</table>
COMPRESSOR

- Downloadable from http://devtools.ericsson.com/etc
- Compresses all new formats plus ETC1
- Two modes; fast and slow
  - "slow" mode faster than previous compressor and does exhaustive compression, i.e., optimal quality
  - "fast" mode provides good compression
- Even quicker compression is possible; we have implemented compression with speed equivalent of 1280x720 in 28 fps on a quad core 1.4 GHz ARM
NETWORK TRANSFER

› ETC2 is 4 bpp, but sending a JPEG with equal quality is cheaper
› However, using different packing methods (ZIP, LZA) it is possible with more efficient transmission
› Recent research shows that special-purpose packing methods can compress ETC-files to sizes smaller than JPEG for equal quality, at around 2.2 bpp.
SUMMARY

› New codecs that provide
  – Higher Quality
  – Alpha (1bit and full alpha)
  – R- and RG-textures

› Mandatory in OpenGL ES 3.0 / OpenGL 4.3
  – Finally mandatory texture compression in OpenGL/OpenGL ES!

› Compressor available online http://devtools.ericsson.com/etc

same bit rate!
Differences ETC2/EAC vs ASTC

› ETC2/EAC codecs are mandatory, ASTC is optional
  – Developers can rely on ETC2/EAC to be there
› ETC2 is backwards compatible with ETC1
  – An ETC1 texture can be used in new devices and millions of old
  – An ETC1 and ETC2 texture can be stored in the same file with less than 2x the size
› ETC2/EAC is likely smaller in silicon area size than ASTC
› ETC2 has punchthrough alpha

› ASTC support more bit rates
  – ETC2 can only support 4 bits per pixel
› ASTC slightly higher quality
  – around half a dB