Bitmap Compression

Introduction

Bitmap compression aims to improve the application performance by reducing the bandwidth consumption and asset size and also by speeding up the bitmaps load and upload.

Candera supports following formats for compressing bitmaps, that are described in detail on this page:

- Run-legth Encoding
- Color Look-Up Table
- Erricson Texture Compression

Run-legth Encoding

Description

Run-length encoding (RLE) is a simple form of data compression in which runs of data are stored as a single data value and count, rather than as the original run. This is most useful on data that contains many such runs so, the content of the data will affect the compression ratio achieved by RLE. Although most RLE algorithms cannot achieve the high compression ratios of the more advanced compression methods, RLE is both easy to implement and quick to execute, making it a good alternative to either using a complex compression algorithm or leaving the image data uncompressed.

RLE compression example

For example, consider an image containing plain black text on a solid white background. There will be many long runs of white pixels in the blank space, and many short runs of black pixels within the text. A hypothetical scan line, with B representing a black pixel and W representing white, might read as follows:

```
WWWWWWWWWWBBWWWWWBWWWWWWWWWWW
```

With a run-length encoding (RLE) data compression algorithm applied to the above hypothetical scan line, it can be rendered as follows:

```
10W2B5W1B11W
```

The run-length code represents the original 30 characters in only 13.

RLE Compression in SC

You can take advantage of using the RLE compression on a bitmap by setting an appropriate pair of converter and format for its bitmap profile as described in the SceneComposer documentation in the Bitmap Profile chapter.

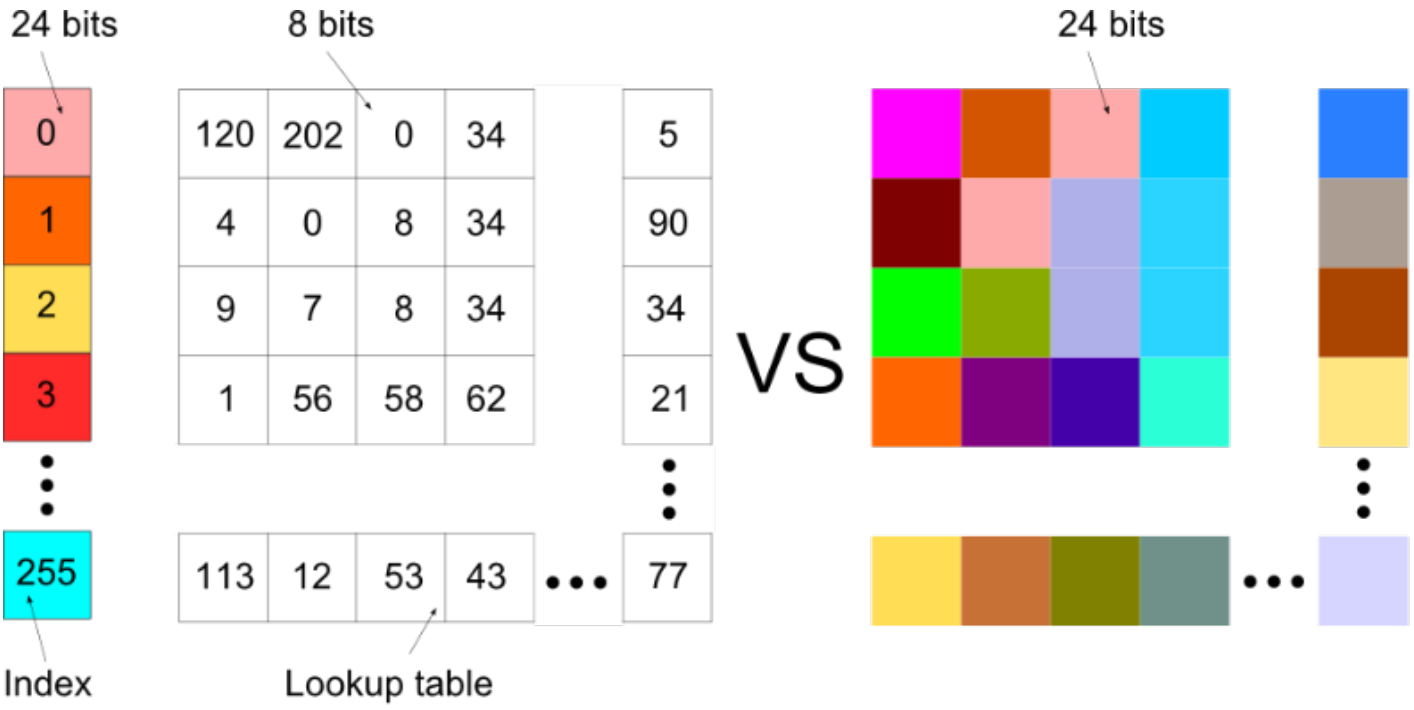
Color Look-Up Table

Description

A **color lookup table** (CLUT) is a table which associates a numeric pixel value with a color to be displayed on the output device. The color is typically specified as a mixture of varying levels of alpha, red, green, and blue, and thus a color lookup table will usually contain three or four components for each possible pixel value. The number of entries in the palette determines the maximum number of colours which can appear simultaneously in the bitmap. Changes to the palette affect the whole image at once and can be used to produce special effects which would be much slower to produce by updating pixels.

CLUT Example

For instance, if a CLUT bitmap has a palette of 256 available colors, the index can be represented by an 8-bit value. The color table is then used to lookup a corresponding 24-bit color value (eight bits of red, green, and blue). Thus, comparing to a full RGB color bitmap, the CLUT bitmap size is roughly one-third.



8-bit CLUT vs 24-bit RGB

The key to the efficiency of the indexed color approach is limiting the number of colors that can be represented.

CLUT Compression in SC

In order to use CLUT compression, set the bitmap with a bitmap profile that uses an appropriate converter and one of the 72 preset CLUT compressions as format. Please consult in the SceneComposer documentation the Bitmap Profile chapter for more details.

Erricson Texture Compression

Description

Ericsson Texture Compression (ETC) is a lossy texture compression scheme which was mandated as part of the OpenGL ES specification for the versions **3.0** and **4.3**. The original 'ETC1' compression scheme provides 6x compression of 24-bit RGB data and it does not support the compression of images with Alpha components. 'ETC2' is an updated version of 'ETC1' which offers higher quality than its predecessor and also introduces support for alpha pixels and R-/RG-textures.

ETC2 allows for more textures to fit in video memory, leads to less traffic on the bus for higher performance and lower power consumption, and it becomes cheaper to transit textures over networks.

The following ETC2 codecs are supported in Candera:

- Candera::Bitmap::Etc2CompressedRgbPixelFormat : Compresses RGB888 data.
- Candera::Bitmap::Etc2EacCompressedRgbaPixelFormat : Compresses RGBA8888 data with full alpha support.
- Candera::Bitmap::Etc2Alpha1CompressedRgbaPixelFormat : Compresses RGBA data where pixels are either fully transparent or fully opaque.

sRGB variants of the above codecs are also available.

EAC is built on the same principles as ETC1/ETC2 but is used for one- or two-channel data. The following EAC codecs are supported:

- Candera::Bitmap::EacCompressedUnsignedRPixelFormat : 4 bpp compressed single channel unsigned R format
- Candera::Bitmap::EacCompressedSignedRPixelFormat : 4 bpp compressed single channel signed R format (can preserve 0 exactly)

- Candera::Bitmap::EacCompressedUnsignedRGPixelFormat : 8 bpp compressed two channel unsigned RG format
- Candera::Bitmap::EacCompressedSignedRGPixelFormat : 8 bpp compressed two channel signed RG format (can preserve 0 exactly)

ETC2/EAC Compression in SC

Please consult in the SceneComposer documentation the chapter Import .KTX Image Format, to find out how to generate and import ETC2/EAC compressed textures.

Revision #10
Created 2023-02-17 08:09:39 UTC by Tsuyoshi.Kato
Updated 2025-01-24 10:04:15 UTC by Elisabeth Zauner