3Dc

From Wikipedia, the free encyclopedia

3Dc (FourCC : ATI2), also known as **DXN**, **BC5**, or **Block Compression 5** is a lossy data compression algorithm for normal maps invented and first implemented by ATI. It builds upon the earlier DXT5 algorithm and is an open standard. 3Dc is now implemented by both ATI and Nvidia.

Contents [hide]

Target application [edit]

The target application, normal mapping, is an extension of bump mapping that simulates lighting on geometric surfaces by reading surface normals from a rectilinear grid analogous to a texture map - giving simple models the impression of increased complexity. This additional channel however increases the load on the graphics system's memory bandwidth. Pre-existing lossy compression algorithms implemented on consumer 3D hardware lacked the precision necessary for reproducing normal maps without excessive visible artefacts, justifying the development of 3Dc.

Algorithm [edit]

Surface normals are three-dimensional vectors of unit length. Because of the length constraint only two elements of any normal need to be stored. The input is therefore an array of two-dimensional values.

Compression is performed in 4×4 blocks. In each block the two components of each value are compressed separately. For each block, each of the two components have a palette of 8 values to choose from. The palettes are generated from two values representing the start and end of a line and the other six values being generated as linear combinations of the start and end values.

Compression is natively achieved by finding the lowest and highest values of the 16 pixels to be compressed and storing each of those as an 8-bit quantity. Individual elements within the 4×4 block are then stored with 3-bits each, representing their position on an 8 step linear scale from the lowest value to the highest. Each pixel's 3-bit value (the palette index) would be chosen by choosing the palette entry with the minimum distance from the original values.

Total storage is 128 bits per 4x4 block once both source components are factored in. In an uncompressed scheme with similar 8-bit precision, the source data is 32 8-bit values for the same area, occupying 256 bits. The algorithm therefore produces a 2:1 compression ratio.

The compression ratio is sometimes stated as being "up to 4:1" as it is common to use 16-bit precision for input data rather than 8-bit. This produces compressed output that is literally 1/4 the size of the input but it is not of comparable precision.

3Dc+ [edit]

3Dc+ (FourCC: ATI1) (also known as **BC4** or **Block Compression 4**) can compress textures, i.e. light maps, shadow maps, HDR textures and material properties. 3Dc+ provides 2:1 compression ratio with single component (DXT5 alpha) 8-bit integer textures and 4:1 compression ratio with normal maps and textures consisting of two 8-bit integer components

References [edit]

- 3Dc White Paper (PDF) 📄
- Microsoft Developer Network article on Block Compression in Direct3D 10 ⧉
- What is a Normal Map? ⧉

- CREATING AND USING NORMAL MAPS ⧉
- Creating Normal Maps ⧉
- 3Dc - higher quality textures with better compression ⧉

Categories: Texture compression | Open formats | Lossy compression algorithms | 3D graphics software