Computer Graphics 1 **Omkar Gadgil**
Spring 2018 **UID: 01737244**
Email: Omkar_gadgil@student.uml.edu

Literature Review 2

Visual Computer (Springer)

1) Title: A new approach to combine texture compression and filtering. (Primary Paper)

Graphics rendering has a growing importance in the Software Industry specifically the gaming world and other entertainment sources. Due to the advancement in the technology we are seeing improve displays with higher resolution and color gamut. Thus it becomes important that the underlying hardware supports this level of rendering.

Texture mapping has been widely used to improve the quality of 3D rendered images. Nowadays the size of all the applications and software have increased considerably due to the added 3D rendering codes and various environments in the application. This has ultimately taken a toll on the storage devices and also the rendering time of any such application. To reduce the storage and bandwidth impact of texture mapping, compression systems are commonly used to store the data .To further increase the quality of the rendered images, texture filtering is also often adopted. These two techniques are generally considered to be independent. First, a decompression step is executed to gather texture samples, which is then followed by a separate filtering step. The authors of the paper have investigated a system based on linear transforms that merges both phases together. This allows more efficient decompression and filtering at higher compression ratios. This paper formally presents the approach for any linear transformation, how the commonly used discrete cosine transform can be adapted to this new approach, and how this method can be implemented in real time on current-generation graphics cards using shaders. Through reuse of the existing hardware filtering, fast magnification and minification filtering is achieved.

Texture systems are a special case of traditional image compression methods in that only certain pixels which are needed by the rendered image need to be decoded. Moreover, these decoded pixels may be requested in a random order determined by the rasterizer. Hence, texture compression systems need to support random access of the coded bitstream. Finally, since textures may be arbitrarily warped by the projection of the geometry on the screen, filtering of texture samples is commonly used to improve the visual quality of the rendered results. The transformation approach described is as follows:

1) Transform-domain magnification filtering: - Here two filtration are performed intra block filtering and inter block filtering. Transformation based image codecs split the image in a

Omkar Gadgil
UID: 01737244
Email: Omkar_gadgil@student.uml.edu

number of blocks. Texture pixels within such a coded block can be efficiently filtered without requiring evaluating the decoding transformation multiple times. Filtering the samples within a discrete cosine transform (DCT) coded block can be trivially implemented by relying on the fact that the DCT is a superposition of cosines. This leads to a more generic and more efficient implementation that can reuse existing texture filtering hardware. In other words this means that we can filter the decoded samples within a block by simply interpolating the rows of the decoding transformation matrix and then doing the dot product with the resulting vector.

2) Transform domain minification filtering:- Here minification filtering similar to ripmapping is done by using the properties of transform coding from the previous method. Ripmaps work by using a series of pre-filtered and down sampled versions of the input image. These down sampled versions are generally generated offline and then stored together with the full resolution texture in texture memory. Based on the resulting filter characteristics, a set of suitable coefficients to discard when generating ripmaps could then be selected..

Bringing it together the first step in coding a texture with our proposed system is the Color Space Conversion (CSC) which transforms the pixels to the YCoCg color space. The next step is then to split the image in blocks of size N×N and transform them, this results in aN2 dimensional coefficient vector per block. The transform we use is a custom variant of the DCT transform where the DCT coefficient is pre-multiplied with the IDCT scale factor. The coefficient vectors are then reordered into coefficient planes by grouping coefficients corresponding to the same basis vector.

Visual Computer (Springer)

2) Title: A JPEG-like texture compression with adaptive quantization for 3D graphics application (Secondary paper)

DCT-based compression is widely used in video and image compression for a high compression ratio, but it suffers from random-access problem when applied to texture compression. In this paper a JPEG-Like DCT-based texture compression technique is described which is suitable for 3D graphics rendering system. DCT-based compression techniques are based on the concept of compacting energy into fewer coefficients in the transform domain and then encoding these coefficients. In the encoding process, only the Huffman code words and quantization tables are

Omkar Gadgil
UID: 01737244
Email: Omkar_gadgil@student.uml.edu

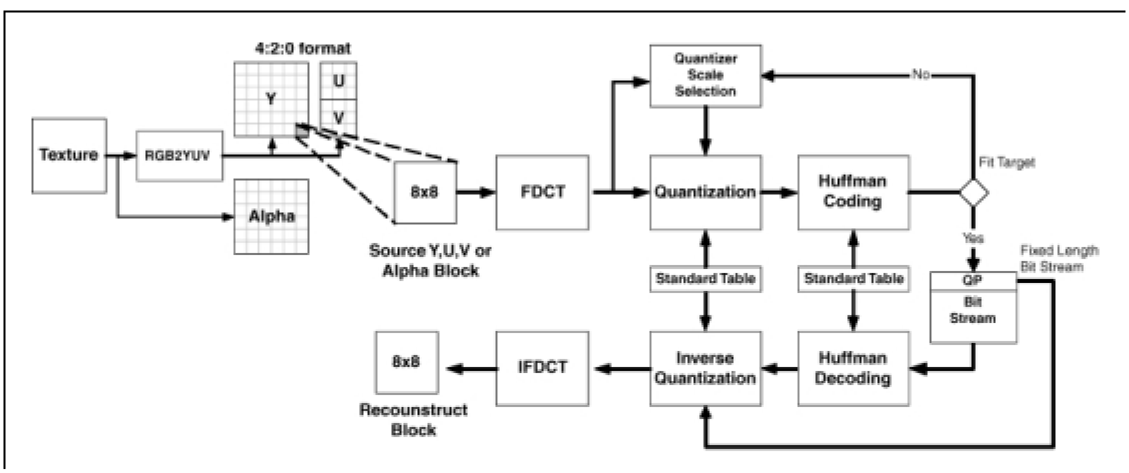
left for the user to control the visual quality and compression ratio. The major distinctions of the proposed algorithm from JPEG are:

•It operates on one 8×8 block individually. Each 8×8 Y or U or V or Alpha block is encoded, decoded and accessed independently.

•It compresses every block toward a fixed length bit stream, thus random access can be achieved and memory space can be utilized efficiently. The bit stream length of every encoded block can be any size, as controlled by user definition or quality control.

•Every 8×8 block has its corresponding quantizer scale, which is pre-defined to control quality and CR.

•The "Alpha" component, which represents the transparency of every texel, is compressed in addition to the R, G and B color components.



Above figure gives the compression/decompression flow of JPEG texture of the described algorithm.

Thus in this paper the authors have reviewed the texture compression techniques of previous works and address the requirements of texture compression. A high-compression-ratio technique such as TREC suffers from the texture random access problem. By analyzing the properties of JPEG, the paper proposed a JPEG-like DCT-based texture compression technique to obtain higher compression ratios(>12) and has solved the random access problem in TREC.

Omkar Gadgil
UID: 01737244
Email: Omkar_gadgil@student.uml.edu

Relation in two papers

The papers summarized above are related with one another in terms of the underlying fundamental DCT implementation. The secondary paper which was published prior to the primary paper gives the DCT based texture compression technique for the texture rendering and mapping. However it falls short in terms of the bandwidth optimization and filtration and the real time rendering of the 3D application. The primary paper published later used the DCT based algorithm to improve the real time rendering 3D rendering. The aforementioned systems assume a dedicated custom hardware decoder and offer very limited scalability in bitrate or complexity. None of these methods deal with texture filtering. It is implicitly assumed that filtering will be handled (by additional hardware)after the decompression of samples. The primary paper addresses this by Transform-domain magnification filtering and Transform domain minification filtering.