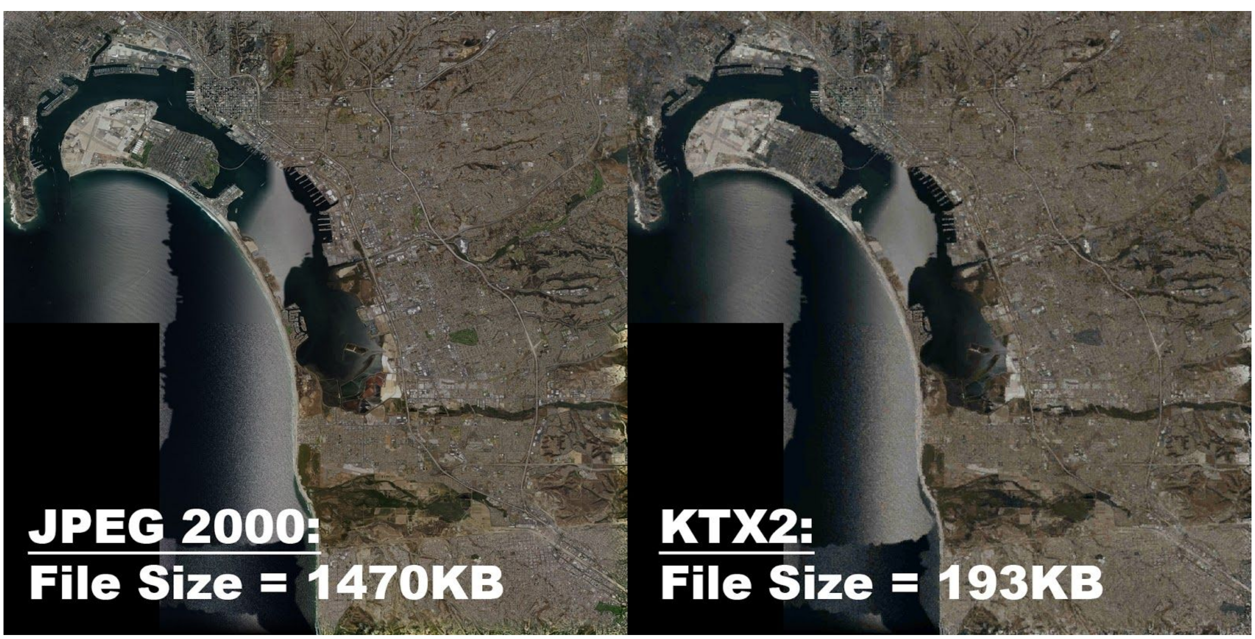
Milestone 1 Report
Technical Report and Test Plan for Consensus-based Emerging Image Compression Standards

Actionable & Accurate 3D Visualization at the Tactical Edge

Contract #: H92405-20-9-P011



1. Executive Summary

The availability of high-resolution geospatial data for imagery, terrain, and 3D models, combined with the increased deployment of handheld devices, such as ATAK, has made it possible for warfighters and operators to gain a high degree of geospatial situational awareness within an operations center and in the field. Warfighters have access to geospatial data for a larger area of interest, and at a higher resolution. This capability has brought forward a complex challenge regarding storage and runtime performance of high-resolution geospatial data that needs to be deployed for streaming and disconnected use. Traditional image formats have ubiquity but come at the cost of either storage, transmission, or runtime/GPU performance, but do not provide the best of all three. Warfighters have thus been limited by the amount of data that can be streamed efficiently or taken offline, and efficiently visualized on their devices.

KTX2 is an emerging consensus-based standard for texture compression under development by the Khronos Group. KTX2 uses supercompressed textures that allow images to have a small storage size comparable to JPEG, while maintaining efficiency for transmission over the web and offline use, and can be transcoded efficiently to GPU native compressed formats. Using KTX2 in geospatial data can enable warfighters to access more data faster, visualize larger areas of interest at higher resolutions, and have a smaller runtime footprint at the same time.

Initial results from testing KTX2 in CesiumJS show a reduction in storage size of 50-85% compared to the same texture compressed using traditional formats such as JPEG/JPEG2000 with only up to 2% quality degradation. In this report, we discuss these results and present a test plan for USSOCOM to undertake to further evaluate KTX2 for geospatial data as a format deployable at the tactical-edge for warfighters and operators.

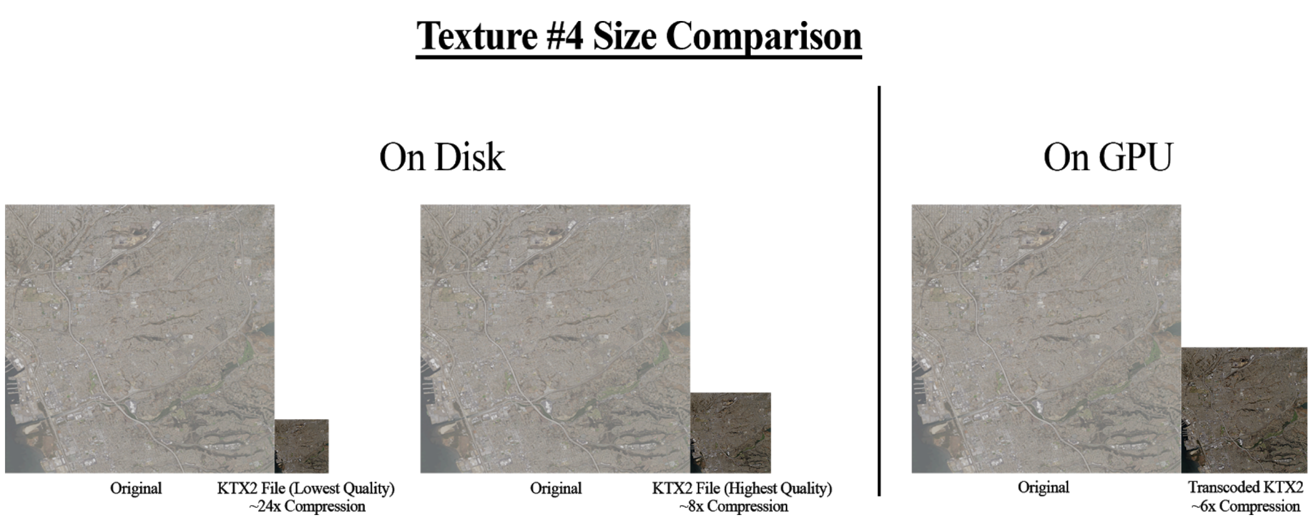


Figure 1. Comparison of Disk and GPU memory footprint of KTX2 vs original JPEG2000 image from from San Diego CDB

2. Background

Geospatial data is a crucial part of situational awareness for operators and warfighters. This includes imagery, terrain, buildings and urban structures, moving models, and other geospatial data. All of these data types are represented, at least in part, by images and textures to add both visual and numerical context depending on the use-case. The availability of higher resolution geospatial data, made possible by modern sensors and technologies, allows warfighters to gain a higher degree of situational awareness by combining the data with 3D geospatial engines for simulation and training, as well as operational use.

Another trend on the rise for operators and warfighters is the use of handheld devices like ATAK, especially in disconnected, intermittent, limited bandwidth (DIL) environments. These devices and their hardware constraints require that ground forces and disconnected operators carry all the data for missions with them, and be able to visualize and analyze that data on portable devices.

This dichotomy of increased availability of high-resolution geospatial data and the use of portable devices like ATAK has presented a challenge that impedes warfighters and operators from maximum situational awareness. Geospatial data has traditionally been designed for desktop computers and enterprise simulation systems. The amount of disk storage and compute resources in these systems is sufficient for geospatial analysis. Hence, the traditional approach to software applications has been to improve the applications, rather than on improving the data.

With mobile devices, storage and compute resources are significantly constrained. Add to that the need to take data offline and on-the-move, as well as battery-life constraints, the storage and performance requirements need to go beyond what software applications can improve, and consider how we can improve the data itself.

Geospatial data represented by images that can use KTX2		
Satellite Imagery	Raster Terrain	Urban 3D Models (Buildings)
Aerial and Drone Imagery	Photogrammetry Models	3D and Moving Models

Table 1: A quick summary of potential applications of KTX2 in geospatial data

We can categorize the goals for images in geospatial datasets for modern situational awareness applications into the following:

1. Reduced disk storage size
2. Fast encoding
3. Fast decoding
4. Low runtime GPU memory usage

5. Increased interoperability with other standards
6. Increased portability across devices and platforms
7. Support for offline usage

A similar challenge currently is being tackled in the gaming industry. The demand for more realistic gaming experience and visual quality, as well as increased gaming performance, has the gaming industry looking to innovative compression standards for images. The Khronos Group, is a consortium of leading graphics and gaming companies and experts, and has developed and standardized many technologies for computer graphics and gaming, including OpenGL, Vulkan, and glTF. The Khronos Group has now undertaken development of the next generation of image compression formats that aim to optimize both compression and runtime performance.

Common image formats such as JPEG, PNG, and GeoTIFF have been optimized for storage on disk and portability across platforms. The ecosystem and support for these formats is widespread and ubiquitous. On the other hand, GPU hardware is not designed to read these images directly. When using these common image formats, software applications need to decode the images first and then send the raw data to GPUs. This results in high usage of GPU memory which is generally much more limited, and even more so on mobile devices.

	JPEG / PNG / GeoTIFF	**GPU Native (ETC, BCn, PVRTC)**	**KTX2 with Basis Universal Texture Compression**
Compressed Disk Storage	Yes	Depends, need to store multiple formats	Yes, stored as supercompressed texture
Fast Encoding	Yes	Depends on format	Yes, done offline, native tools available as open-source software
Fast Decoding to GPU formats	No	No decoding required	Yes, transcodes to GPU native texture format
Low GPU Memory Usage	No, uncompressed on GPU	Yes, GPU native texture format	Yes, uses GPU native compressed format
Portable	Yes	No, formats are hardware specific	Yes, transcoded to hardware specific format

Table 2. A simplified table highlighting the key criteria laid out for image formats

A common solution to this high memory usage problem in games is to use GPU native compression formats like ETC, BC and PVRTC. While texture data in these formats is much smaller compared to uncompressed GPU data being sent to the GPU, the downside of these formats is potentially large compression time and limited portability. Each platform uses its own preferred GPU native formats which may not work on other devices. Additionally, the cost of transcoding textures to these GPU native formats can be a resource intensive process at runtime. As interoperability of data is a significant requirement, trying to store various GPU native formats of the same images increases the disk storage size and thus is not a useful solution.

3. KTX2 Image Compression Format

KTX2 image compression format is an emerging consensus-based format currently under development in Khronos, for storing images that makes it easier to use compressed textures natively, universally and with the possibility of offline and streaming use-cases. KTX2 introduces supercompression, which enables the format to have compressed disk storage size comparable to JPEG and PNG, while being easily transcoded to GPU native compressed format which lowers runtime GPU memory usage.

KTX2 allows for cross-platform direct on-the-fly transcoding of textures to natively supported GPU textures on-device. KTX2 textures can also be used as part of glTF, the 3D model format. Figure 1 displays an example of how this new pipeline would look like.

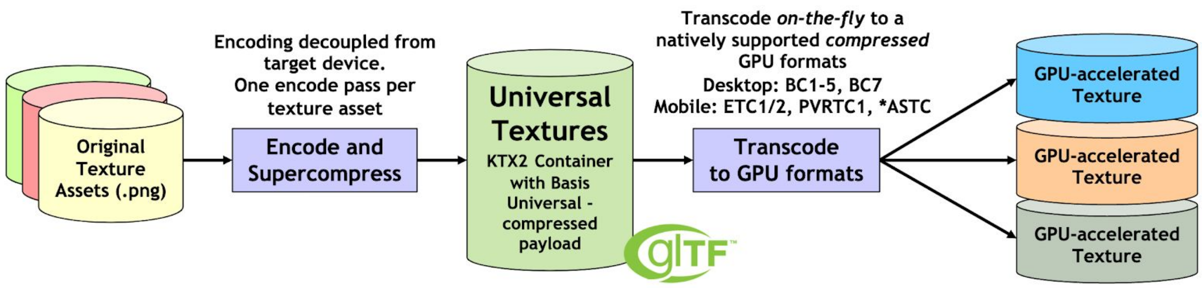


Figure 2: An example pipeline for KTX2 compressed textures. Showing how KTX2 (center) provides an easily transcoded container for textures. (Credit: [Khronos Group, Siggraph 19](#))

KTX2 internally supports the *.basis* format for compression, a lossy texture compression format with a storage size comparable to JPEG in compression, but built for GPU textures. This allows for efficient GPU transcoding, and is the first format to support a universal GPU texture format on the web. The format allows for a tuneable quality level, where a higher quality comes at a slight increase in transcoding time and storage size. This allows for much flexibility for dealing with geospatial data that may need to be losslessly transcoded.

The thesis for evaluating KTX2 as a potential solution to the problems laid out implied that KTX2 checks all the boxes for criteria for images as described in the table below. The application of KTX2 format to geospatial data and situational awareness could address the problems laid out with the current image formats. It can be applied to imagery, terrain, 3D model textures, and other data represented by images or rasters.

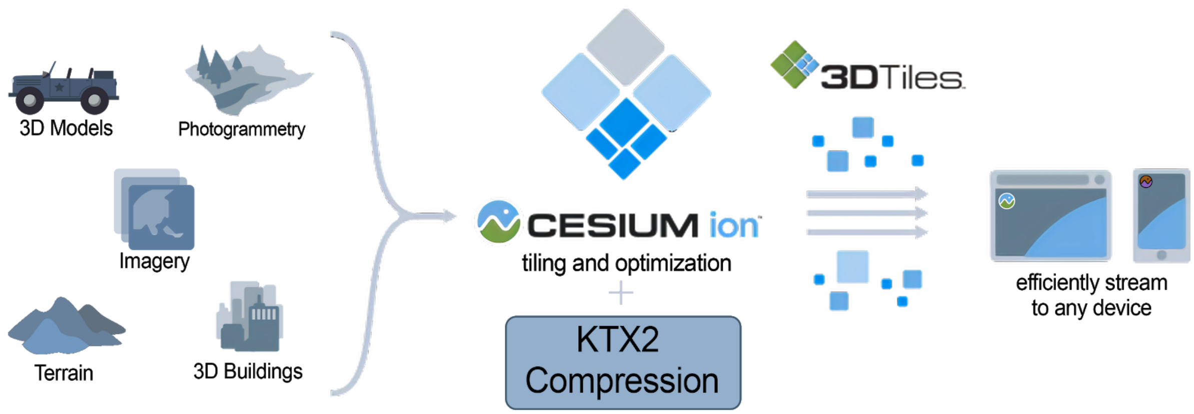


Figure 3. A representation of the pipeline used to create and evaluate KTX2 for storage and streaming into CesiumJS that was used to evaluate the results for this report.

As part of this deliverable, Cesium ran experiments on geospatial data, including orthoimagery, terrain, and textures from 3D models and photogrammetry to produce an initial report on the expected gains of using KTX2 as an image format in geospatial datasets. Cesium has added support for KTX2 to CesiumJS and our data pipelines which has allowed us to create KTX2 images and evaluate compression, encoding and decoding performance, and GPU memory used.

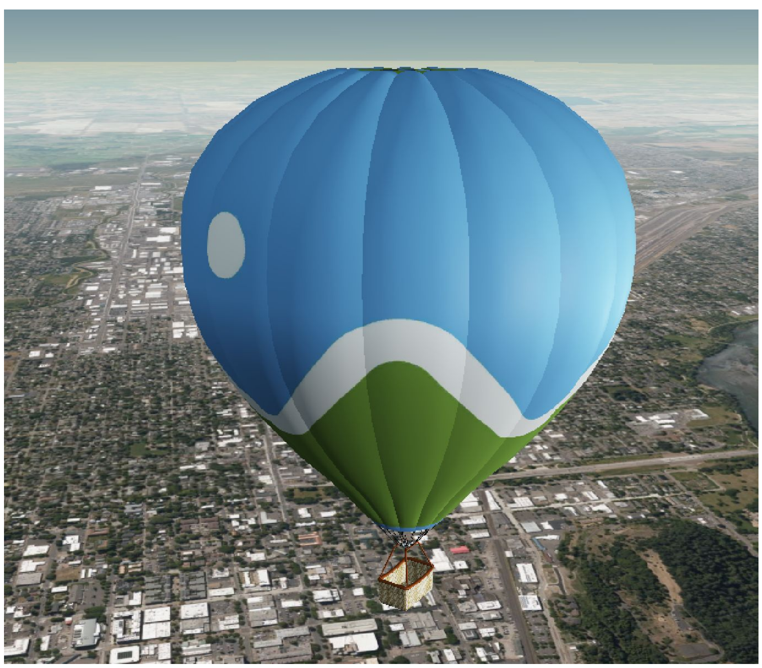


Figure 4. The "Cesium Hot Air Balloon" model compressed using KTX2

Figure 4 shows an example of a 3D model which was compressed to 38% of its original file size using a combination of Draco mesh compression and KTX2 texture compression.

The power of KTX2 compression extends far beyond simple models such as the Cesium Hot Air Balloon. In the following section, the results of more than 25 different textures from satellite imagery to terrain and building photogrammetry textures are laid out, which show massive drops in storage size through KTX2 compression.

4. Test Results

KTX2 has been successfully integrated into the CesiumJS runtime engine. We conducted empirical tests to evaluate results of KTX2 compression applied to geospatial texture data. A representative subset of results from our tests is shown in this report, but the trends extend to the entire initial experiment dataset of 28 varied textures. A full report is attached as Appendix - KTX2 spreadsheet.

Note on JPEG2000 Images: The JPEG2000 compression algorithm builds multiple resolution representations of the image during the compression process. This pyramid representation can be used for purposes beyond compression, for example to represent the image when zoomed out. The JPEG2000 images in San Diego CDB dataset are 1024x1024 pixels each, with internal pyramid representations of 512x512 and 256x256 pixels. As this feature is built into the algorithm itself, for the purposes of this comparison we have used the full file size of JPEG2000 images. This pyramid structure does not affect runtime analysis.

Disk Storage Size: KTX2 compression allows the ability to compress textures very efficiently for storage on disk, and rapidly transcodes them directly to a compressed GPU texture representation. The KTX2 format also allows for variable quality (between 1 and 255) to allow for high quality compression when extreme precision is needed. Below are some examples of the size of compressed textures before and after they are converted into KTX2 files.

	Traditional Format Sizes (KB)		KTX2 Size (KB)	
Content	**PNG/JPG Size**	**JPEG2000**	**Quality = 1**	**Quality = 255**
#1: Low LOD Imagery	1204	910	47	140
#2: Low LOD Imagery	1914	1470	69	193
#3: Medium LOD Imagery	2325	1841	86	221
#4: Medium LOD Imagery	2450	1954	82	220
#5: Water Imagery	1790	1270	69	178
#6: High LOD Imagery	1079	1012	71	177
#7: Photogrammetry Texture	40	N/A	5	20
#8: Photogrammetry Texture	95	N/A	13	48
#9: Photogrammetry Texture	295	N/A	38	129

Table 3. Comparison of Disk Size of nine different sample textures from our test dataset

We can see what this looks like for texture #1, the low LOD satellite imagery from San Diego CDB dataset, which is 15.4% of its original JPEG2000 file size while not compromising too much on image quality. For a satellite imagery, the limited visual quality loss may not impact situational awareness.

Figure 5. Texture #1 in a compressed PNG format (left) vs. KTX2 compressed format at highest quality

Quantitatively we can measure the pixel difference between the original image and the compressed image by examining the root mean squared error (RMSE) between the images. This is the percentage of the pixel's color information lost during compression. Taking Texture #1 as an example, we can see that the RMSE as a function of compression quality shows that both Desktop and Android compression formats have very low RMSE values which drop to ~0.6% as we get to the highest compression quality, Q255.

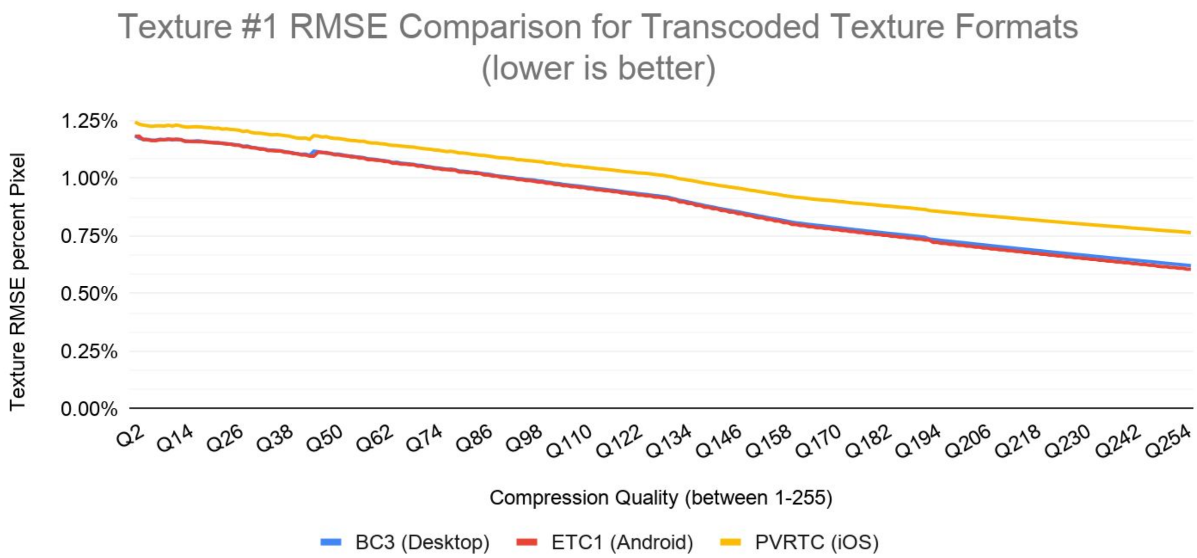


Figure 6. Texture #1 RMSE across various different transcoded textures.

Transcoding Time: As mentioned earlier, any form compression does incur a cost. With KTX2, this cost is minimized due to the nature of the compression format. The format is optimized for GPU transcoding, which enables it to be quickly converted to the format required for the device's GPU. The table below shows our transcoding times in milliseconds for our sample textures. These results were collected on three devices:

- Macbook Pro with a 2.3 GHz 8-Core Intel Core i9 and 16 GB RAM
- Samsung Galaxy Note 10+ with a Qualcomm Snapdragon 855 chipset and 12 GB RAM
- Apple iPhone 7

	Transcoding Time (ms)					
Content	**Desktop (Q1)**	**Desktop (Q255)**	**Android (Q1)**	**Android (Q255)**	**iOS (Q1)**	**iOS (Q255)**
#1: Low LOD Imagery	20.0	37.8	67.1	41.2	62.7	95.8
#2: Low LOD Imagery	17.7	25.3	65.9	52.2	54.5	76.0
#3: Medium LOD Imagery	19.4	24.9	94.4	52.5	55.6	77.9
#4: Medium LOD Imagery	17.6	23.9	67.0	48.8	57.8	77.0
#5: Water Imagery	17.2	21.4	61.9	50.1	78.8	78.9
#6: High LOD Imagery	17.7	25.6	167.4	100.8	64.1	91.5
#7: Photogrammetry Texture	6.8	8.6	63.0	51.8	9.2	15.2
#8: Photogrammetry Texture	8.8	11.9	80.2	52.6	18.2	34.8
#9: Photogrammetry Texture	12.8	20.5	154.5	102.0	57.0	69.3

Table 4. Time taken to transcode KTX2 texture into the platform's native GPU format

Texture GPU Memory Usage (at runtime): In addition to compressed storage, one of the most crucial aspects of KTX2 compression is reducing texture size on a device's GPU using GPU native compression formats at runtime. This allows for more textures to be loaded into GPU memory at the same time, allowing for better visual experience and analytics. The table below shows the size (in KB) on the GPU for both the uncompressed and compressed versions of our textures. It is important to note here that once uploaded to the GPU, the texture size is the same regardless of device or quality.

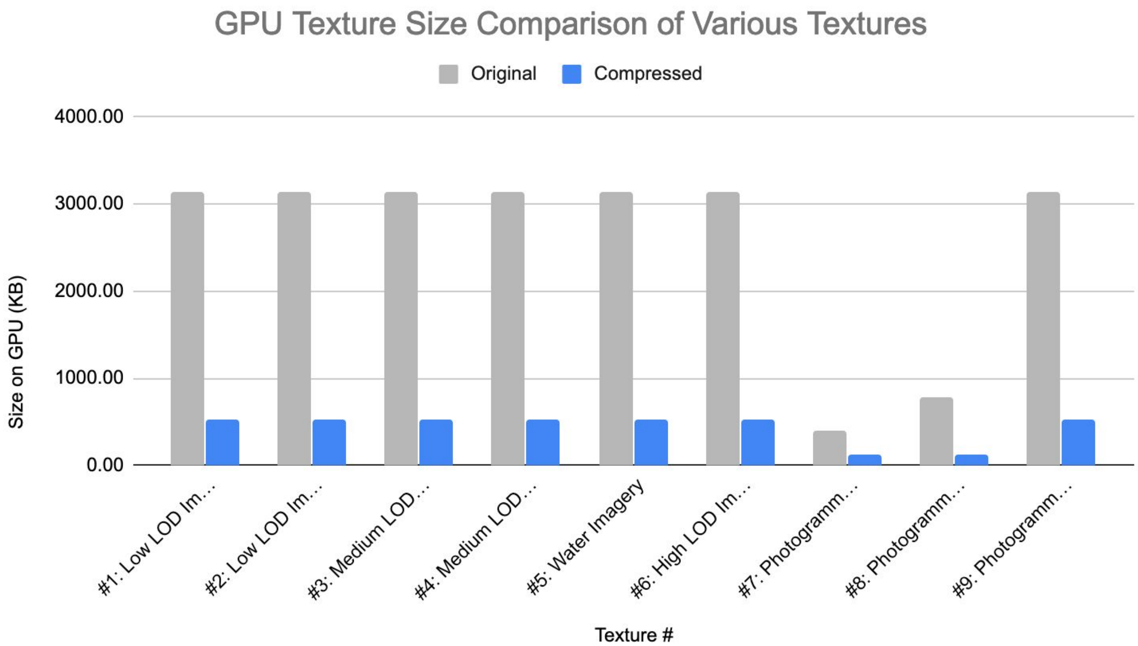


Figure 7. GPU Texture Sizes for each of our nine sample textures

Content	Size on GPU (Original)	Size on GPU (Compressed)	Compression Ratio (Original:Compressed)
#1: Low LOD Imagery	3145.73	524.29	6:1
#2: Low LOD Imagery	3145.73	524.29	6:1
#3: Medium LOD Imagery	3145.73	524.29	6:1
#4: Medium LOD Imagery	3145.73	524.29	6:1
#5: Water Imagery	3145.73	524.29	6:1
#6: High LOD Imagery	3145.73	524.29	6:1
#7: Photogrammetry Texture	393.22	131.07	3:1
#8: Photogrammetry Texture	786.43	131.07	6:1
#9: Photogrammetry Texture	3145.73	524.29	6:1

Table 5. GPU Texture Sizes for each of our nine sample textures

5. Future Work

Based on these initial results, we believe that KTX2 has a large role to play in representing images in geospatial data both for storage and runtime applications. KTX2's robust format allows it to also be applied to metadata textures as well that may store data not necessarily restricted to RGB texture images.

KTX2 in glTF: KTX2 is being developed as part of the 3D Formats Group at Khronos, the group responsible for glTF. Khronos is working on integrating KTX2 as a texture format for glTF, which can then be deployed as 3D models in geospatial applications.

KTX2 in 3D Tiles: 3D Tiles uses glTF as the 3D geometry payload and thus can also take advantage of KTX2 image compression. This would mean that massive heterogenous 3D data can take full advantage of the 3D Tiles specification, along with Draco mesh compression and KTX2 image compression results in a highly-optimized dataset for streaming and offline use-cases.

KTX2 in 3D Tiles for Tiled Imagery: 3D Tiles was designed for efficient streaming of heterogeneous 3D data. The structure of 3D Tiles however is conducive to streaming 2D imagery tiles as well. Using KTX2 as an imagery payload, 3D Tiles could be used to stream tiled imagery, thus bringing optimized texture compression to streaming imagery tiles, which is commonly done using JPEG or PNG files with specifications like TMS and WMTS.

KTX2 in CDB: As KTX2 provides compression comparable to common image formats such as JPEG, JPEG2000, and PNG, it is feasible that KTX2 could be used as the default image format in geospatial and simulation datasets like CDB. KTX2 can be used for imagery, terrain, textures for 3D and moving models, thus improving the disk footprint and runtime optimization.

6. Test Plan

We propose that USSOCOM engage a test plan to further research the application of KTX2 to use-cases in geospatial data. Images in geospatial data are used for various use-cases including imagery, terrain, photogrammetry, 3D models. A test plan that evaluates compression standards and settings available in KTX2 and how these apply to each use-case would be beneficial to producing datasets that would be optimized for the use-cases.

Objectives: The objective of the test should be to deliver a best practices report that defines the set of options to use for each use-case and the expected performance and storage improvements that can be expected. Based on the results presented in this report, delivering images as KTX2 to operators and warfighters in both streaming and offline use-cases can provide significantly improved performance.

Scope: The planned scope for the test should be to evaluate a matrix of options - source image format (JPEG, JPEG2000, PNG, GeoTIFF), device platform (desktop, web, mobile), and KTX2 compression options as have been discussed earlier in results. This test matrix would be able to objectively evaluate and compare the best combination for each use-case to deliver the most optimized performance to warfighters and operators.

Metrics: The metrics for evaluating the efficiency and effectiveness of KTX2 as an image compression standard would be:

- *Storage size* - The storage footprint of KTX2 images compared to the source format.
- *GPU memory usage* - The runtime GPU memory used by images during rendering.
- Encoding cost - The time and computing resources required to encode images into KTX2.
- *Decoding cost* - The time and computing resources required to transcode KTX2 into GPU encoded format, compared to decoding source images into uncompressed GPU memory.
- *Precision loss* - The precision in image data lost by compressing images as KTX2. For images used for visual representation, the error might have higher tolerance compared to images used for analytics and computation.

As mentioned in the results section of this report, formats like JPEG2000 and GeoTIFF have internal pyramid representations. As part of this test plan, we recommend that efforts be taken to possibly remove any pyramid representations to ensure that disk size is a representation of the full image only. This will allow a fair storage size comparison between KTX2 and the source image formats.

Risks: The KTX2 standard is still being developed by the Khronos Group and will undergo standardization in the near future. As a result, the standard is being updated frequently.

7. References:

- KTX File Specification: http://github.khronos.org/KTX-Specification/
- KTX Software Documentation: https://github.khronos.org/KTX-Software/
- KTX Software Repository: https://github.com/KhronosGroup/KTX-Software
- San Diego CDB Dataset used for test data was made available to Cesium via USSOCOM and CAE.