**HelioStreamer prototype**

Inhalt

[Abstract 2](#_Toc506527547)

[Feasibility study 3](#_Toc506527548)

[Videos 4](#_Toc506527549)

[Investigated alternatives & future work 4](#_Toc506527550)

[Media Source Extensions API 4](#_Toc506527551)

[Caching with IndexedDB 4](#_Toc506527552)

[HTML5 workers 4](#_Toc506527553)

[GL\_RED instead of Luminance 4](#_Toc506527554)

[Software video decoders like Broadway.js 5](#_Toc506527555)

[notes 6](#_Toc506527556)

[Answer from helioviewer.org 6](#_Toc506527557)

# Abstract

The HelioStreamer is a website attempting to stream movies onto a 3D sphere, initially with AIA images. The movies streamed are preprocessed MP4 files, with a pyramidal buildup both temporally (POT ratio) and spatially (512px tiles).

In the current implementation the required videos are determined in the client (position, zoom level, time), and the required videos are requested from the server and downloaded. Then each of the MP4s are displayed in a HMLT5 video element. In each frame, each required video is passed to the GPU. This last step is the performance bottleneck for this implementation. On the GPU side, one single 4k x 4k texture is initialized at start, then sub regions are updated with video frames.

An online demo of a recent prototype state is available here: <http://86.119.40.9/Viewer/2>

# Feasibility study

Quick info on current choices & feasibility information

|  |  |
| --- | --- |
| GPU <-> CPU communication | Currently main bottleneck. texSubImage2d has a specialized API for handling html5 videos. Currently this API is used (see Investigated alternatives) |
| Video support | The most supported video format by current web browsers is MP4 H264 (<https://caniuse.com/#feat=mpeg4>, including «fragmented» versions for streaming). All tested platforms support HTML5 video playback of grayscale 8bit interlaced mp4 videos. |
| Storage | For the proof of concept, a video tree was built from AIA images covering one day, with 4 different pixel resolutions (full res 4k down to 512) and 6 different time resolutions i.e. playback speeds (1x, 2x, 4x, 8x, 16x, 32x).  Helioviewer.org provided 2379 JPEG2000 images for 1 day of AIA 131, from which 6'035 videos were built. With the current video format we expect **50TB** of videos, compared to the **57TB** of JPEG2000 images they cover (see Results.xlsx). |
| Bandwidth | In full resolution streaming mode, i.e. streaming the full 4k images, only 7MB/sec are required. Yet the tool is built to download with region of interests, with a realistic worst case of **2.5MB/sec** bandwidth required. For mobile devices, bandwidth usage can be throttled down to **0.15MB/sec**. |
| WebGL | The proof of concept projects up to 64 videos (8 512px videos in width and height) onto a 3D sphere.  Practically all devices support WebGL and 4k textures (<https://webglstats.com/webgl/parameter/MAX_TEXTURE_SIZE>) |
| Decoding | The decoding of MP4 videos is currently done hardware accelerated in HMTL5 video tags. Its speed currently poses no performance bottleneck, but can become one with frequent seeking e.g. for constant synchronization. |
| Synchronisation | The video playbacks quickly fall out of sync, therefore they are regularly synced. A sync is done with a video seek and can therefore punish performance heavily if done regularly and for many videos. Currently, a lead video element has the time to which all other videos adjust. |
| Web Worker | Currently, web workers are not being used. All browsers support web workers (<https://caniuse.com/#feat=webworkers>) |

The proof of concept showed that no criterion from this first evaluation set hinders further investigation into video streaming of AIA images.

# Videos

All videos of the video tree have the following specs:

* Bitrate chosen arbitrarily by example of YouTube (TODO: tune)
* Grayscale (since data is only (EUV) intensities)
* 8bit, as from JPEG2000 helioviewer.org images (and also sufficient for a movie)
* 512px x 512px x 64 frames chosen for convenient file size
* 32 FPS 🡺 2 seconds per video

# Investigated alternatives & future work

## Media Source Extensions API

There is growing support for the Media Source Extensions API (<https://caniuse.com/#feat=mediasource>), with which close control can be exercised over video playback, chunking and streaming.

First attempts to use MSE showed that its use is non-trivial. There entire JavaScript players dedicated to use e.g. MSE with MPEG-DASH (e.g. Shaka player).

## Caching with IndexedDB

For caching IndexedDB seems suitable (<https://caniuse.com/#search=indexeddb>).

## HTML5 workers

An initial technical prototype had a separate worker thread creating random 4k x 4k textures, and passing this texture every frame to the main thread. While achieving a stable 15-30 fps across browsers, the following issues showed:

* No DOM tree is available in workers. This poses an issue as the only way to profit from hardware accelerated decoding seems to be over HTML5 videos (which are part of the DOM).   
  There is an OffscreenCanvas featured specified by W3C, but it barely has any browser support.
* Workers have no access to GPUs

JavaScript decoding of the MP4s in the worker (e.g. with Broadway.js) wasn’t investigated into, as it doesn’t seem promising.

This prototype is still available under <http://86.119.40.9/Viewer/1>

## GL\_RED instead of Luminance

Only WebGL2, and 50% slower than LUMINANCE

## Software video decoders like Broadway.js

JavaScript video decoders like Broadway.js are significantly slower than hardware accelerated video decoding, like in HTML5 video elements. The initial student prototype did not make use of hardware accelerated decoding and was therefore very slow.

# notes

### Answer from helioviewer.org

Roman,

In nominal operations, we have one AIA image in each of the higher temperature channels (94, 131, 171, 193, 211, 304, 335) once every 36 seconds. We have AIA 4500 images once every hour. We have AIA 1600 and 1700 images once every 36 seconds, very roughly speaking. HMI continuum and magnetogram images are about once every 45 seconds each.

We have 57 TB of AIA images and 8 TB of HMI images.

Let me know if you would like any more information.

Thanks,

Jack

### Helioviewer api alternative URL

**Expéditeur:** Eric Buchlin <[eric.buchlin@ias.u-psud.fr](mailto:eric.buchlin@ias.u-psud.fr)>  
**Date:** 23 janvier 2018 à 15:53:55 UTC+1  
**Destinataire:** Andre Csillaghy <[andre.csillaghy@fhnw.ch](mailto:andre.csillaghy@fhnw.ch)>  
**Objet:** **API helioviewer a MEDOC**

Bonjour,  
  
L'API v2 est accessible de  
<https://helioviewer-api.ias.u-psud.fr/>  
(redirection vers la documentation si pas d'argument)  
À bientôt,  
--   
Éric