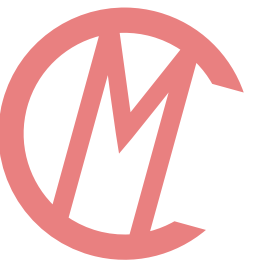






# HDR contents via IIIF

Next steps



# Introduction: What is HDR?

## Incomplete and greatly simplified

- 👉 HDR images using tone mapping are around since 3 decades
  - 👉 Composed of different exposure levels
  - 👉 Images for display still have 8 bit colors (colors are „warped“)
- 👉 We are talking about HDR display using a 10 bit color space
  - 👉 Tone mapping might be still applied
- 👉 Generated either by HDR capable imaging sensors, composed or artificial.



# Introduction: Current status

## High level (user / consumer) view

### 👉 HDR video

- 👉 Currently best known and supported in the browser - for example on YouTube
- 👉 Supported by TV sets
- 👉 Can be created by mobile phones

### 👉 HDR images

- 👉 Can be created by modern mobile phones
- 👉 Supported by some browsers
- 👉 Different level of support by different formats

### 👉 HDR images as Textures (for 3D models)

- 👉 Supported by modern game engines
- 👉 Browser support experimental (Three.js)



# Introduction: Limitations

## Soft- and hardware (as of November 2024)

- 👉 You need a HDR-capable monitor / screen
- 👉 Operating system level support inconsistent
  - 👉 Mac OS is certainly best
  - 👉 Windows support need to be enabled explicitly (but it's solid enough for gaming)
  - 👉 Linux support currently experimental
- 👉 Browser support (~95% for HDR images)
  - 👉 Firefox only has partial support for HDR video
  - 👉 Chromium-based browsers might need additional flag for all features (see demo)
- 👉 OSS support in development / experimental
  - 👉 VIPS, ImageMagick etc.



# Use cases

See <https://christianmahnke.de/en/post/iiif-online-meeting-2024-slides>

- 👉 Presenting artifacts with enhanced colors
  - 👉 Immersive effects
- 👉 Image analysis
- 👉 Mixing different channels (wave lengths) as gain maps
- 👉 Scientific visualization
  - 👉 Like bioluminescence, multispectral / false-colour images etc.
- 👉 Highlighting / contrast enhancements (as optional image operations) in VREs
- 👉 Not necessarily limited to web browsers



# Technical Background

**Not all browsers support all possibilities yet!**

- 👉 Different technical methods depending on media type (image, video, 3D)
- 👉 Currently focused on Images, but shouldn't be limited
- 👉 Currently advanced browser settings might be required



# Technical Background: Image formats (gain maps)

These formats offer a fallback by storing the lighting information separately

- 👉 Separation between SDR and HDR components
- 👉 Gain map holds „differences“ to create HDR image
- 👉 HDR gain map saved in image metadata
- 👉 Available for JPEG, AVIF, JXL and HEIF - specification (ISO 21496-1) currently settling, multiple vendors have implementations (November 2024)
- 👉 Benefits
  - 👉 Fallback for SDR-only soft- and hardware
  - 👉 Efficient storage by subsampled gain map





# Technical Background: Videos

Limited only to browser based presentation

- 👉 Well know from sites like YouTube (since November 2016!)
- 👉 Works in Chrome
- 👉 Works on Firefox for Mac since May 2022
  - 👉 Complete Windows and Linux support still missing
- 👉 Provided via HDR10(+), Advanced HDR or Dolby Vision - 10Bit (or more) color spaces
- 👉 Encoded in H.264, H.265, AV1, VP9
- 👉 Usually encapsulated in MP4 (others possible)



# Technical Background: 3D Models

Limited only to browser based presentation

- 👉 Currently only textures are supported, since those are just images
- 👉 „Real“ HDR rendering not yet supported by browsers, if there is any lighting calculation tone mapping will be applied



# Open questions I

## Short summary of the discussion in IIF/api#2312

- 👉 Started as a discussion on Image API
- 👉 Indication of HDR using `supports`
- 👉 Maybe add another `format` like `ultrahdr` or `jpg+hdr`
- 👉 But there are other HDR capable image formats, should everyone get it's own `format` or a `+hdr` suffix?
- 👉 Should colorspace / bit depth be covered / included?
  - 👉 Might require additional metadata fields
- 👉 `quality` can be considered as an abstraction of colorspace (like `color`, `gray`, `bitonal`), so `extraQualities` might be a good fit - favoured in the discussion
- 👉 How generic should it be? Presentation API (for 3D) as well?
- 👉 Presentation API: Provide metadata as part of the `service`





# Open questions II (high level)

## What should a HDR related API change achieve?

- 👉 Indicate HDR content to an agent (viewer) to be able to check / hint technical requirements / possibilities
- 👉 Might cover the complete range of possible contents
  - 👉 Image API related
    - 👉 Would HDR be client side pull or server push?
    - 👉 Should it be legal to supply HDR image as `format jpg`?
    - 👉 Or should the user be required to explicitly require HDR (as additional `format`)?
  - 👉 Presentation API related
    - 👉 Might be necessary for other HDR capable media, like Video or 3D



# Proposal (Short term)

## More usage convention than specification

- 👉 HDR images as `format jpg` (and `quality color`) should be considered legal if there is a transparent SDR fallback, for backwards compatibility.
- 👉 Provide a URI as `supports` hint
- 👉 No API change required, as a first shot, just register the URI and let viewer implementations handle required steps



# Proposal (long term)

## Next major revision of involved APIs

### 👉 Image API

- 👉 Address the question on bit depth / colorspace
- 👉 Define `extraQualities` - not only limited to HDR use cases

### 👉 Presentation API

- 👉 Add an indication for HDR contents - independent of specific media type
- 👉 Can this be done using a `profile` technical property?
- 👉 Should a dedicated technical property be added?



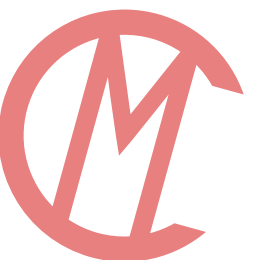


# Discussion

**Please comment at [IIIF/api#2312](https://www.idolmaster.com/forums/thread/2312)**

cmahnke@gmail.com

**<https://christianmahnke.de/en/>**



<https://christianmahnke.de>