

W3C Media Capture Depth Stream Extensions

aka mediacapture-depth

W3C TPAC 2014

What

- Make 3D camera a first-class citizen on the Web Platform
- Integrate with the existing platform APIs and primitives: gUM, Canvas 2D, WebGL
- [You saw live demos of this technology in action on the stage if you attended the W3C20 Anniversary Symposium on Wed.]

Current Status

- Use cases and requirements
- First Public Working Draft published
- Experimental Chromium implementation

Use Cases

- Video games
- 3D object scanning
- Video conferencing
- Augmented reality
- ... and more

IDL

New interfaces

- **DepthData, CameraParameters**

Extensions

- **MediaStreamConstraints**
 - boolean depth
- **MediaStream**
 - getDepthTracks()
- **MediaStreamTrack**
 - kind === “depth”
- **CanvasRenderingContext2D**
 - {create|get|put}DepthData()

Usage Example

```
navigator.getUserMedia({ video: true, depth: true }, success, failure);

function success(s) {
    // wire the RGB + depth stream into a <video> element for playback
    var video = document.querySelector('#video');
    video.src = URL.createObjectURL(s);
    video.play();

    // construct a new MediaStream out of the existing depth track(s)
    var depthStream = new MediaStream(s.getDepthTracks());

    // send the newly created depth stream over a RTCPeerConnection
    var peerConnection = new RTCPeerConnection(config);
    peerConnection.addStream(depthStream);

    // wire the depth stream into another <video> element for playback
    var depthVideo = document.querySelector('#depthVideo');
    depthVideo.src = URL.createObjectURL(depthStream);
    depthVideo.play();
}
```

Next Steps

❑ WebGL extension (Khronos)

<https://www.khronos.org/registry/webgl/extensions/>

❑ Address open issues (W3C)

<https://github.com/w3c/mediacapture-depth/issues>

- ❑ What is the minimum set of metadata CameraParameters should expose
- ❑ Define the algorithms for the createDepthData(), getDepthData(), and putDepthData() methods
- ❑ Change DepthData.data to Float32Array
- ❑ What happens in addTrack(mst) if mst.kind == 'depth'?
- ❑ Specify the depth stream transmission via WebRTC PeerConnection
- ❑ Specify the depth value encoding
- ❑ Need to define in more detail how rgb & d streams are calibrated
- ❑ Specify nature and settings of the depth track constraints
- ❑ Specify the depth stream to WebGL uploading
- ❑ Add color and depth camera coordinate projection support
- ❑ Add depth unit and range capabilities

Demos

- The Magic Xylophone https://www.youtube.com/watch?v=cZiSPsaZ_Dc
- The Fruit Ninja https://www.youtube.com/watch?v=mjU_uF6bS5E
- RGB+Depth with WebGL <https://www.youtube.com/watch?v=LSaI2TaAxYc>

References

- Specification

<http://w3c.github.io/mediacapture-depth/>

[see the spec header for repo, open issues, commit history]

- Use Cases and Requirements

https://www.w3.org/wiki/Media_Capture_Depth_Stream_Extension

- Experimental Chromium Implementation

https://github.com/huningxin/chromium-crosswalk/tree/depth_stream