

Mark Foltz (Google)
Elad Alon (Google)
Guido Urdaneta (Google)
Riju Bhaumik (Intel)
Steven Becker (Microsoft)
Youenn Fablet (Apple)
Eric Carlson (Apple)
Anssi Kostiaainen (Intel)
Xiaohan Wang (Google)
Eero Hakkinen (Intel)
Brachel Zachernuk (Apple)
Jan-Ivar Bruaroey (Mozilla)
Sunggook Chue (Microsoft)

Introduction of topic, effects including browser and OSes.

Reminder of W3C code of ethics, etc.

Mentions some previous work done (by Intel's Riju?) through constraints.

Mentions how over the last few years, OSes started offering camera effects.
Mentions audio effects, says won't be covered, mentions there might be some overlap.
Goes over Windows, macOS, reactions ("another set of features")
Mentions higher end chromebooks and their effects.

Video conferencing apps offer their own effects. Gives the example of Meet.
Mentions Teams.
Mentions Zoom.

Mentions the same effect or interfering effects being applied.

Youenn mentions that mute was solved; Elad disagrees. Discussions to follow.

Mark will refer to blur generally, but it generalizes to other effects.

API surfaces for checking (1) support for blur and (2) state of the effect.

Mentions events, mentions per-frame metadata.

Mentions how the app could notify the user, offer their own effect, etc.

Another strategy - let apps know how much control over the effect.

Mark apologizes for his appearance in screenshot. :-P

Explains how UA could show a dialog to cancel blur.

Mentions limitations in the app-OS interaction over effects, and how changing the OS-level state could affect all other native apps and Web apps.

Youenn asks how many times users use a single camera on multiple sites simultaneously.

Mentions exposing the background segmentation mark and letting apps do their own effects.

Mark says decided not to focus on controlling platform effects.

Mark shows code:

- * Capability detection
- * State query
- * Event handler

Mark demonstrates reading the same info on individual frames. Gives an example of a site that only wants to transmit blur or unblurred frames, as the case may be.

Constraint-based approach:

Setting an object on the track, check if blur is supported, onconfigurationchange, etc.

Discusses detecting capabilities.

Youenn discusses the mute case.

Eric asks about background-replacement from the OS, and how that is to be exposed to the app.

Mark says it depends on what set of effects we have, and the granularity it allows in exposing info.

Discussion of whether the two things are different from the app's POV.

Eric thinks may be the same, Elad and Mark think that not.

Discussion of permanent vs. transitory effects.

Discussion of bucketing effects. Eric mentions technical limitations in telling effects apart, or even knowing that an effect is active.

Discussion of state leaking cross-origin.

Elad mentions software fallback.

Youenn mentions the ability to programmatically add reactions.

Brandel mentions proctoring, ability to guarantee the source of a frame and that it is unmodified. Elad mentions virtual cameras and questions if a guarantee to that effect could be made from a technical pov.

Discussion of when applyConstraints's promise resolves, and whether that should affect whether to go with constraints or not.

Youenn: It would be good if people produced PRs.

Action Items

- ☐ Elad to file an issue about applyConstraints' Promise resolution time.
- ☐ Mark to fix how the Chrome sample code uses applyConstraints()
- ☐ Mark to file an issue / draft a PR to expose effect status on VideoFrame to continue discussion
- ☐ Mark to work with Riju to make x-platform behavior consistent for blur control