VP8 for RTCWEB Mandatory To Implement

Technical arguments and issues
draft-alvestrand-rtcweb-vp8-00

VP8

- Widely useful
  - One profile. All implementations interoperate.
  - Free and best of breed hw implementation design available for free.
  - Real time capable.

- Widely implemented
  - Hardware and Software
  - 50+ SOC have VP8 hw in production with real time capable implementations.

- Widely deployed
  - WebRTC in Chrome and Firefox, ooVoo, QQ, others

VP8 Is Well Defined

- RFC 6386 - with source code
- Submitted to ISO SC29/WG11 (MPEG)
- No decoder profiles
- No known interoperability issues

Comparing Picture Quality

- A codec's output is no better than its platform
  - although it may be considerably worse
- Open source allows anyone to compare
  - if they can agree on what to compare against.
- Comparing is hard. PSNR is a metric.
- Command lines to compare:

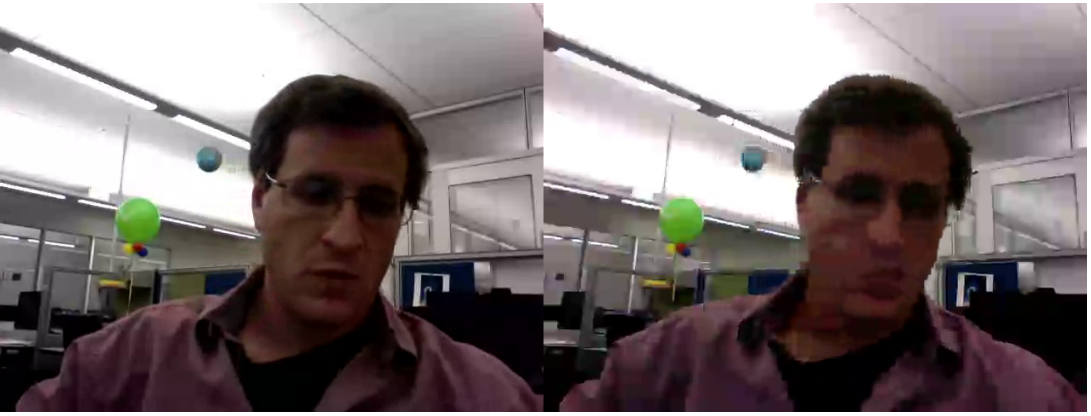
```
vpxenc --lag-in-frames=0 --target-bitrate=$5 --kf-min-dist=3000 --kf-max-dist=3000 --cpu-used=-2 --fps=$4 --static-thresh=1 --token-parts=1 --drop-frame=0 --end-usage=cbr --min-q=2 --max-q=56 --undershoot-pct=100 --overshoot-pct=15 --buf-sz=1000 --buf-initial-sz=5000 --buf-optimal-sz=600 --max-intra-rate=1200 --resize-allowed=0 --passes=1 --rt --noise-sensitivity=0 -w $2 -h $3 $1.yuv -o $1-$5.webm
```

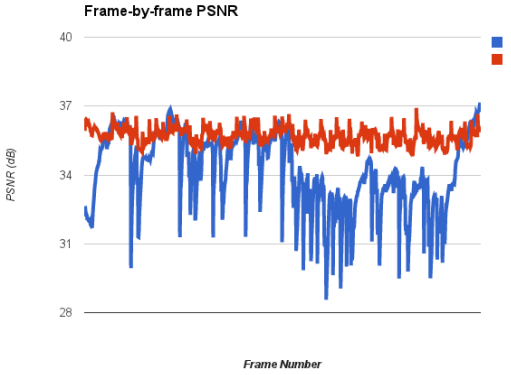
```
x264 --vbv-bufsize $5 --bitrate $2 --fps $3 --profile baseline --no-scenecut --keyint infinite --input-res $4 -o ./$1_$2.mkv $1
```

- Scripts and test data are made public.

VP8 Wins on Quality

- Conferencing test: Less bits, same PSNR as H.264 constrained baseline
  - Google has made test script public
- MPEG references from IVC project: Significantly better than AVC "anchors"
  - Study done by independent contributors





VP8 Wins on Performance

Tests run on difficult 720p material

- Software Encode: 720p 48-96 fps, 1 core
- Software Decode: 720p 200 fps on PC hardware (1 core)
  - H.264 Baseline: 100 fps
- Details in the internet-draft

H.264 Hardware Has Issues

- Wild variations in what profiles are supported
- Decoders in devices far outnumber encoders
- Codecs are tuned for a particular application
  - High resolution = low compression (photo)
  - Low bandwidth = no real-time (video)
  - Outside target parameters = bad quality
- H.264 HW behind private APIs
  - iOS is an example of this

Note: Most devices with video bandwidth have CPUs powerful enough for software codecs.

VP8 Hardware

- More than 50 manufacturers
- Freely available hardware IPR **and** RTL
- Consistent capability sets
- Real time capable.

Performance is good

- 1080p decode in 25 mW
- > 10 SD stream decode on a single chip
- More frugal in chip area and memory bandwidth than H.264

VP8 Is Maintained

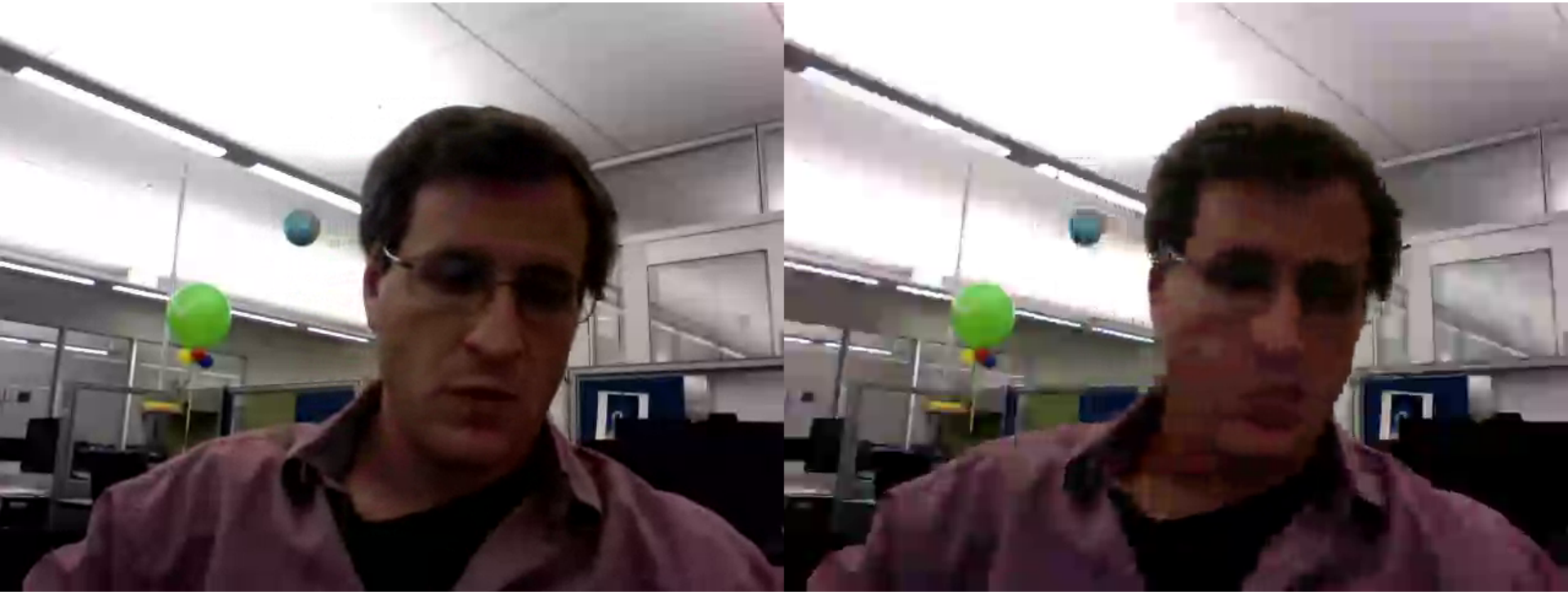
- Every change at webmproject.org automatically tested
  - Linux (manual tests: Windows / Android)
  - Unit / System / Input fuzz testing
  - Dashboards track quality metrics per commit
- Every change at webrtc.org automatically tested
  - Win / OSX / Linux / Android
  - Valgrind / asan / tsan / memcheck
  - Input fuzz testing
  - End-to-end video quality testing
- Chrome testing
  - Manual release testing
  - Automated fuzz-testing

Summary

- VP8 can meet or beat the performance of all proposed alternatives, on any metric.
  - If the test set is reasonably large & diverse
- VP8 is suitable for and used for real time.
- VP8 is available now. The reference platform is the one people use.
- VP8 is good enough to make interworking using the MTI viable for RTCWEB.
- VP8 should be chosen for RTCWEB MTI.

Addendum: Frames, bigger

VP8 vs H.264 at 146/184 kbps



Addendum: PSNR - bigger