Adaptive Streaming of Interactive Free Viewpoint Videos to Heterogeneous Clients

Ahmed Hamza[1] and Mohamed Hefeeda[1,2]

[1]Simon Fraser University, Canada

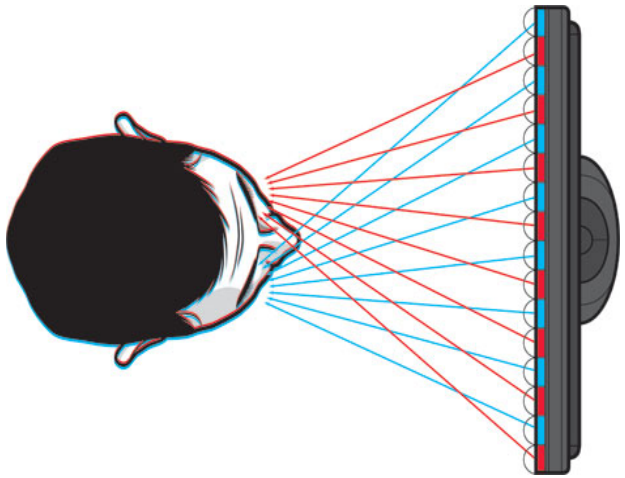
[2]Qatar Computing Research Institute, Qatar

12 May 2016

SFU

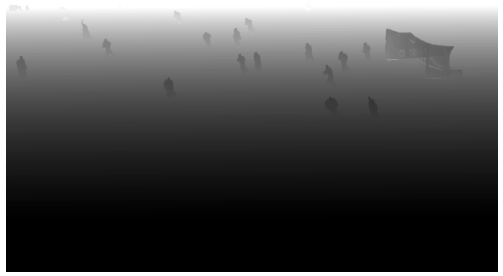
معهد قطر لبحوث الحوسبة
Qatar Computing Research Institute
Member of Qatar Foundation

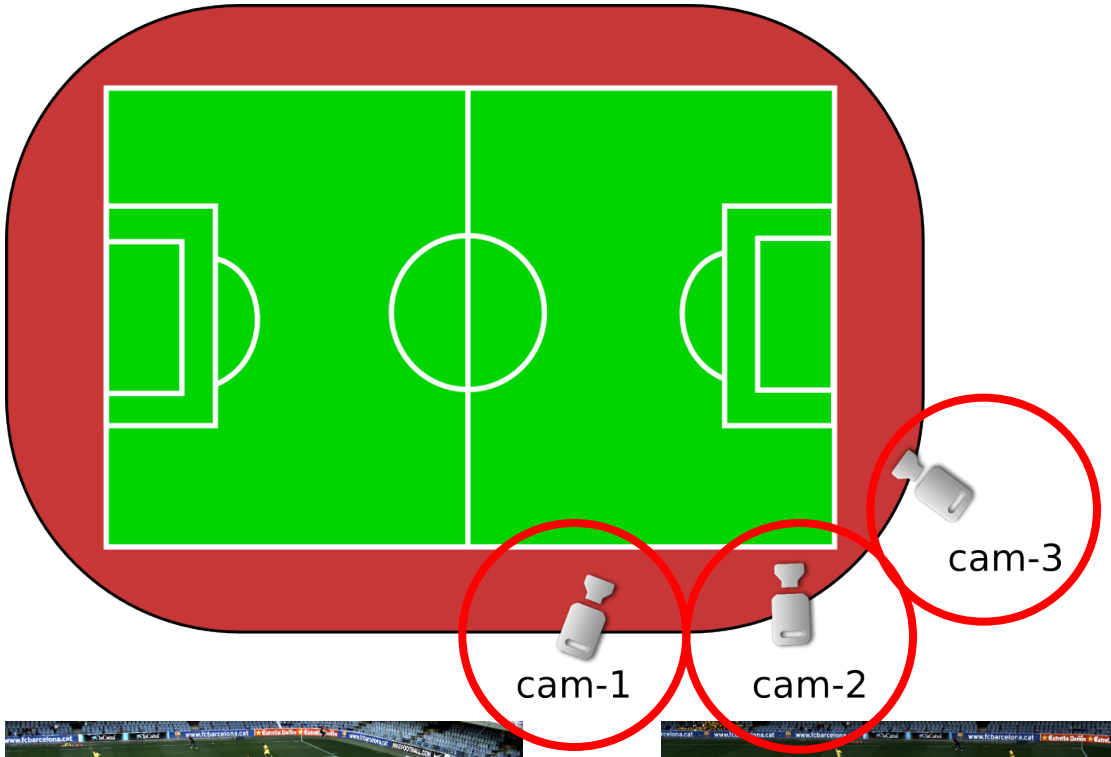
Introduction



Free-viewpoint Video

Depth Streams





Multi-view Plus Depth (MVD)

Left Reference View
(Texture + Depth)

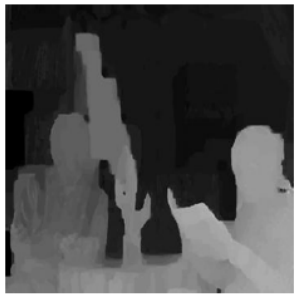
Example: 2-view plus depth

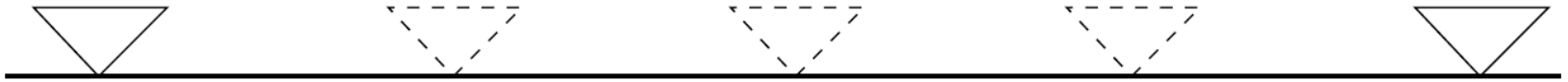
Synthesized View 1

Synthesized View 2

Synthesized View 3

Right Reference View
(Texture + Depth)





FVV Streaming is Challenging

- **FVV streaming**
  - multiple video streams (multiple views, multiple components)
  - rendered frames are the result of a view synthesis process from received components

- **Complex rate adaptation**
  - quality of rendered video stream is dependent on the qualities of component streams used as references in the view synthesis process
  - changes in components' bit rates do not equally contribute to the quality of the synthesized video

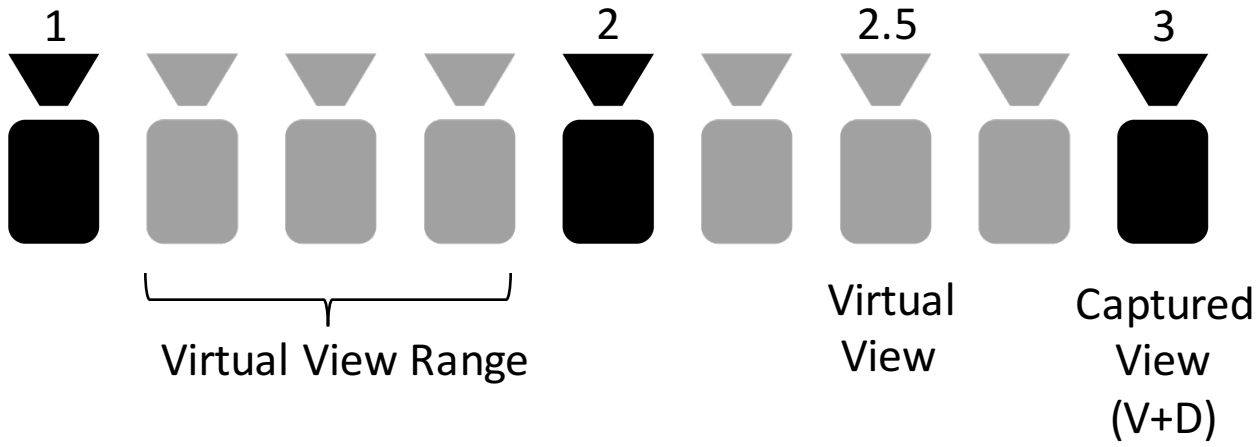
Problem

- **Given current viewpoint position and available network bandwidth**
  - **which reference views should be requested?**
  - **which representations for each (texture and depth) component should be downloaded?**

- **Objective:**
  - **Maximize quality of rendered virtual views at the client side**

Proposed Solution

- **Two-step approach**
  - **Determine set of reference views to be requested from server in order to render target viewpoint**
  - **Decide on the representations for the segments of the scheduled views' components**
- **Terminology**



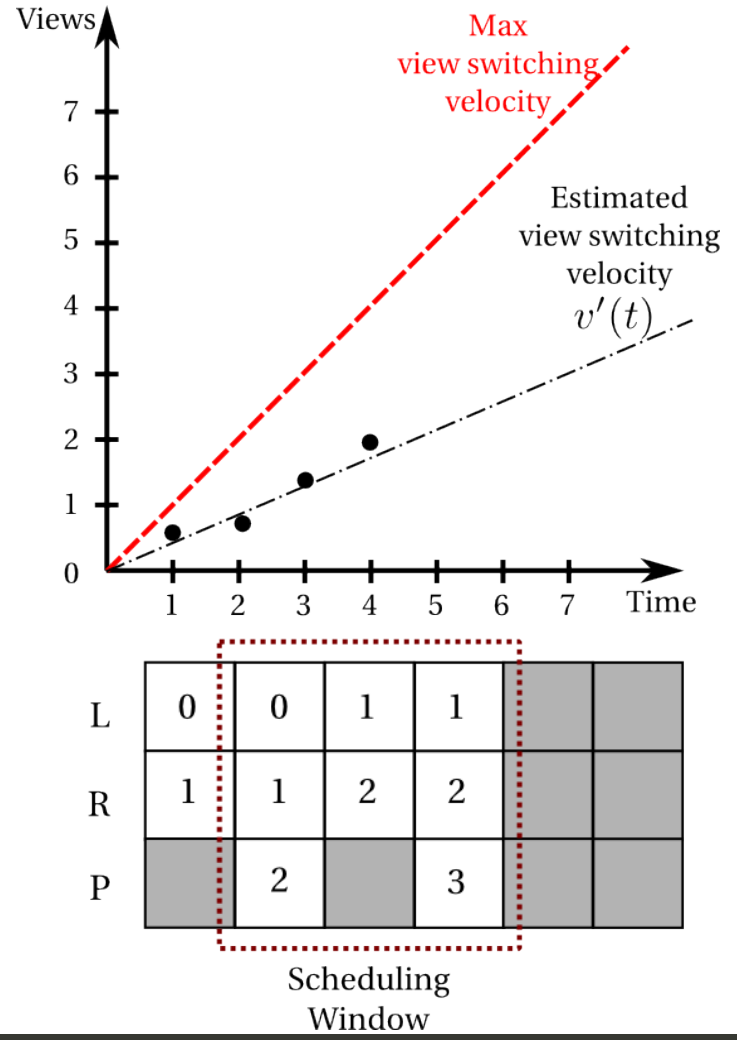
Reference View Scheduling

- **Predict + Pre-fetch**
  - **periodically record user's viewpoint position**
  - **use navigation path prediction techniques to extrapolate future position**
  - **pre-fetch additional reference view if necessary**

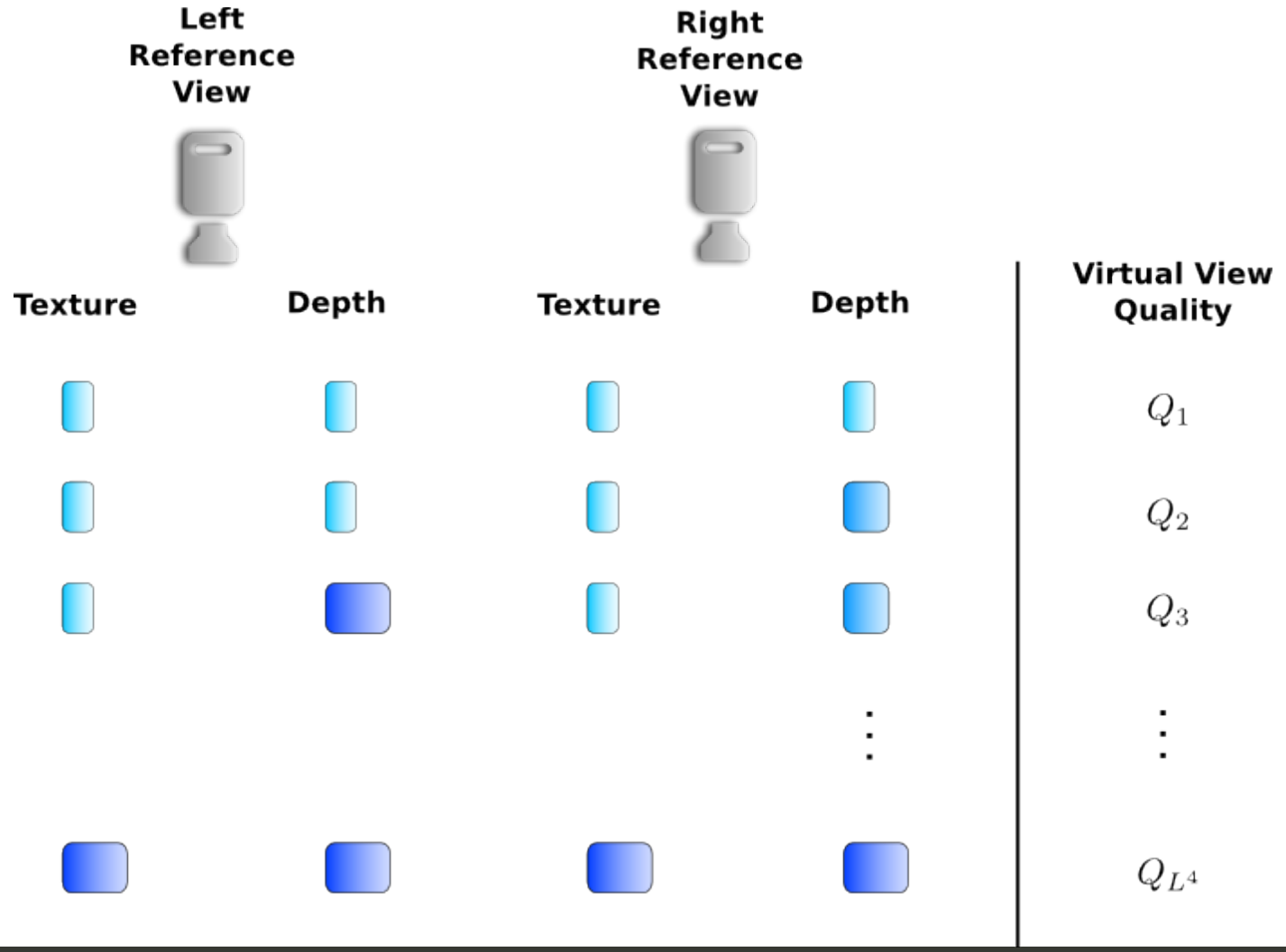
Reference View Scheduling

- Viewpoint position prediction
  - Dead reckoning

- Steps:
  - View switching velocity
    $$v(t) = (x(t) - x(t - \Delta))/\Delta$$
  - Smoothing
    $$v'(t) = \theta \cdot v(t) + (1 - \theta) \cdot v'(t - \Delta)$$
  - Prediction
    $$x(t + \tau) = x(t) + v(t) \cdot \tau$$



Virtual View Distortion Model



Virtual View Quality-Aware Rate Adaptation

- **Use virtual view quality models to guide the rate adaptation process**
  - **Empirical models → (M – 1)KL^4 decode-synthesize iterations**
  - **Analytical models → faster to obtain, less overhead, near optimal quality**

- **Relation between reference views quality/bitrate and quality of synthesized virtual view**

$$D_v = \lambda D_t^L + \mu D_d^L + \nu D_t^R + \xi D_d^R + c$$

Virtual View Quality-Aware Rate Adaptation

- **For each supported virtual view position**
  - **Solve system of linear equations to obtain model coefficients**

$$\begin{pmatrix} D^L_{t\,1} & D^R_{t\,1} & D^L_{d\,1} & D^R_{d\,1} & 1 \\ D^L_{t\,2} & D^R_{t\,2} & D^L_{d\,2} & D^R_{d\,2} & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ D^L_{t\,n} & D^R_{t\,n} & D^L_{d\,n} & D^R_{d\,n} & 1 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \end{pmatrix} = \begin{pmatrix} D_{v1} \\ D_{v2} \\ \vdots \\ D_{vn} \end{pmatrix}$$

  - **Signal model coefficients in extended MPD file**

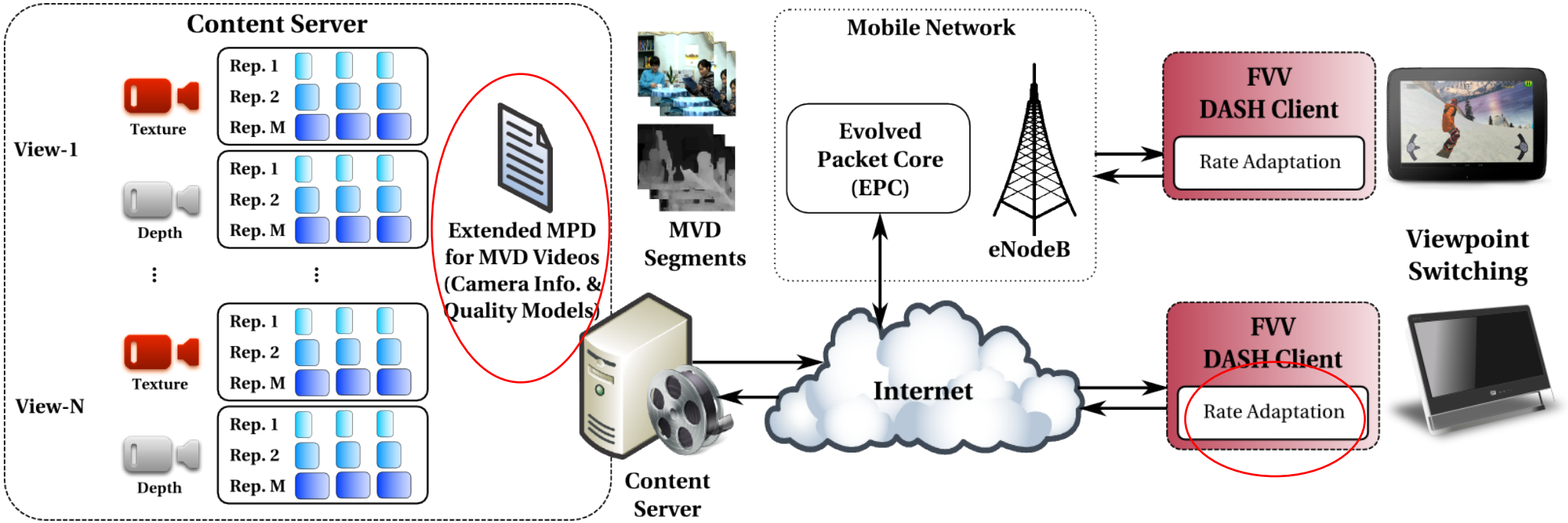
Rate Adaptation

- **Given:**
  - **Estimated channel bandwidth**
  - **Set of virtual viewpoint positions for scheduled virtual view range(s)**

- **Find optimal operating point p^* which minimizes average distortion over all virtual viewpoint positions**
  - **such that** $R(p^*) \leq R_c$

```
1  p* ← φ, D_min ← ∞
2  foreach p ∈ P do
3      if R(p) ≤ R_c then
4          D_sum ← 0
5          for i ← 0 to K do
6              α ← A(i)
7              D_sum ← D_sum + D(p, α)
8          D_avg(p) ← D_sum / K
9          if (D_avg(p) < D_min) then
10             D_min ← D_avg(p)
11             p* ← p
12 if p* == φ then
13     p* ← lowest quality representations
14 return p*
```

System Architecture



MVD Signaling

- **Extended MPD file**

Camera Parameters

```
<CameraParameters …>

</CameraParameters>
```

Per segment index virtual view quality models

```
<VVRDModel …>

</VVRDModel>
```

Components of captured (reference) views

```
<Period>
  <AdaptationSet …>
  </AdaptationSet>
  <AdaptationSet …>
  </AdaptationSet>
</Period>
```

Extended MPD

- **Camera Parameters**

```
<CameraParameters>
<View id="0">
 <IntrinsicParam fx="2241.26" fy="2241.26" cx="701.5" cy="514.5" />
 <ExtrinsicParam rotation="1,0,0,0,1,0,0,0,1" translation="5,0,0" />
 <ZRange zNear="448.2512" zFar="11206.2803" />
</View>
</CameraParamters>
```

Extended MPD

- **Virtual view quality models in MPD**

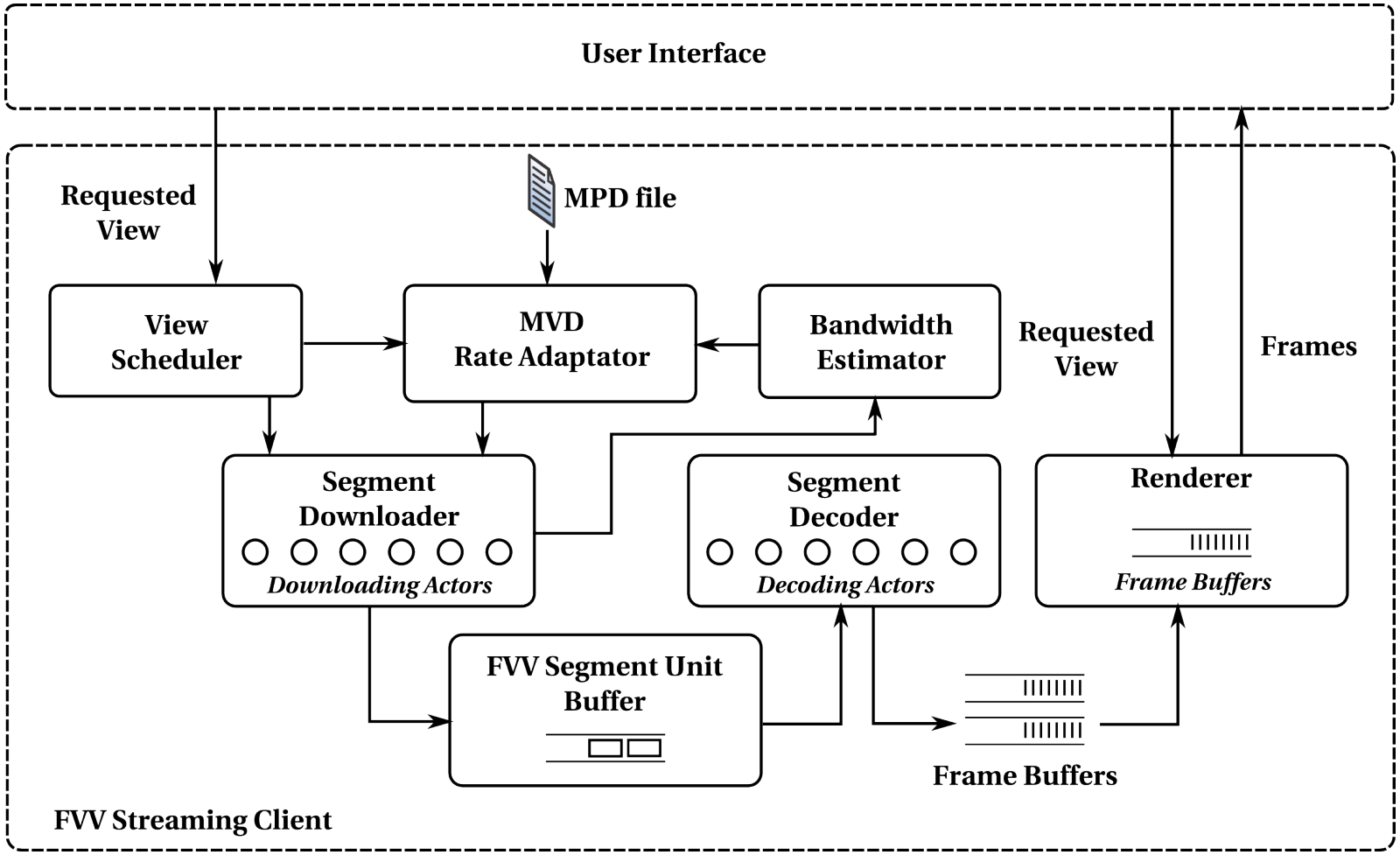
```
<VVRDModel metric="psnr">
<SegmentVVRDModel segId="11">
  <VVRange id="1" l="3" r="5">
    <VVParam alpha="0.25" a0="0.24" a1="0.27" a2="0.08" a3="0.03"
        a4="11.99" />
    <VVParam alpha="0.50" a0="0.23" a1="0.22" a2="0.05" a3="0.05"
        a4="14.07" />
    <VVParam alpha="0.75" a0="0.24" a1="0.19" a2="0.02" a3="0.07"
        a4="14.06" />
  </VVRange>
</SegmentVVRDModel>
</VVRDModel>
```

Extended MPD

- **Reference Streams Quality**

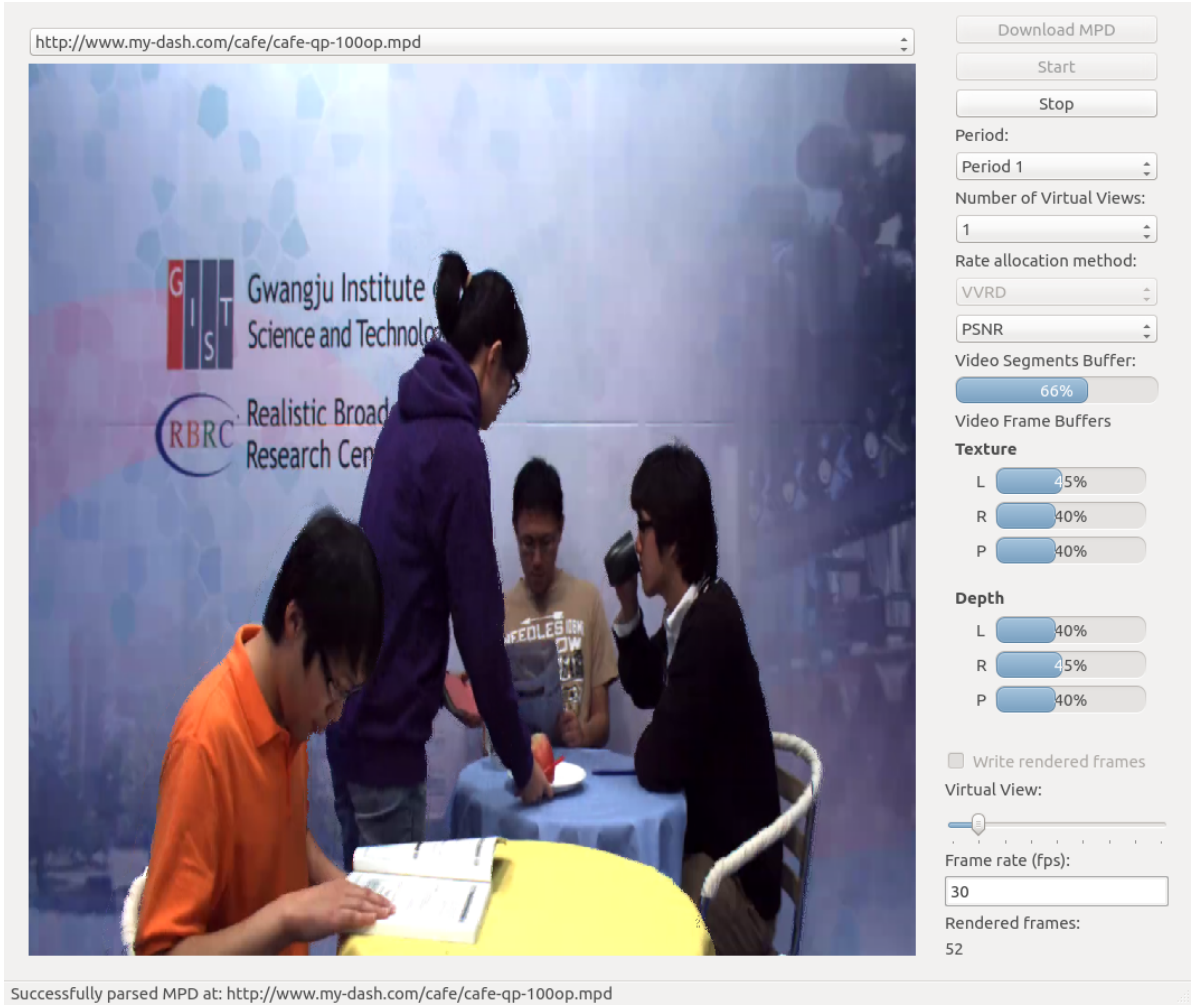
```
<AdaptationSet mimeType="video/mp4" codecs="avc1.640828">
  <Viewpoint schemeIdUri="urn:mpeg:dash:mvv:2014" value="0"/>
  <Role schemeIdUri="urn:mpeg:dash:v+d:2014" value="t"/>
   <Representation bandwidth="128000" avgPSNR="34.1" avgSSIM="0.959">
    <SegmentList duration="1">
         <Initialization sourceURL="oblivion_128_t0_init.mp4"/>
         <SegmentURL media="oblivion_128_t0_seg1.m4s"/>
        </SegmentList>
   </Representation>
</AdaptationSet>
```

Streaming Client Components



Screenshot

- **Implemented using C++**
  - *libdash*
  - *FFmpeg*
  - *GPAC*
- **Actor-based concurrency**
  - message passing
- **Indicators:**
  - Segment and frame buffer levels
  - Viewpoint position



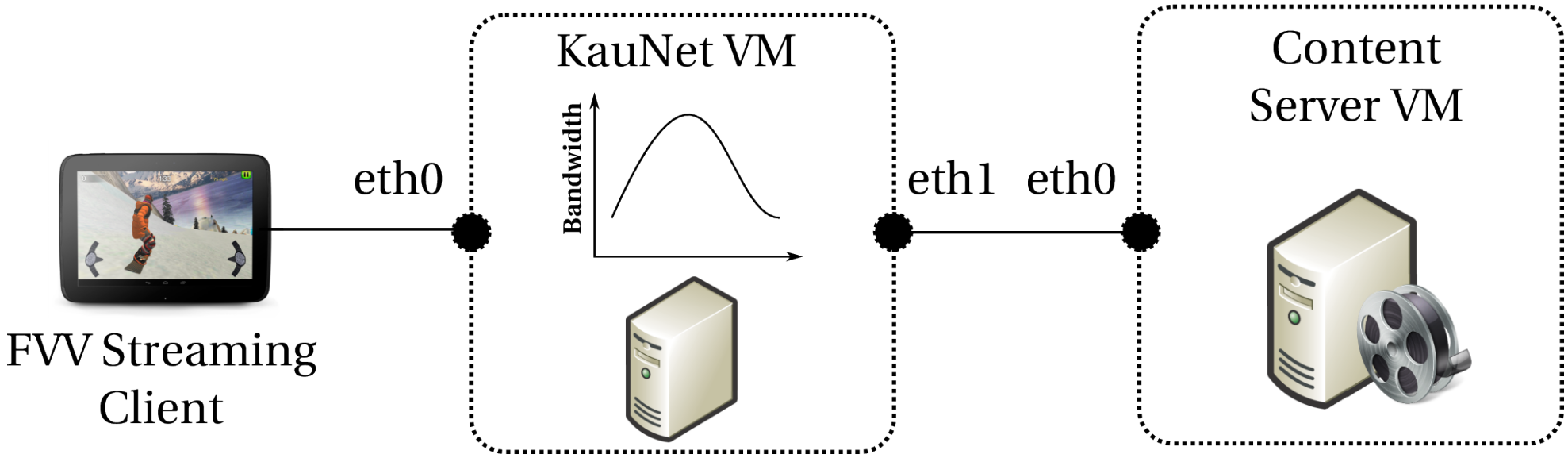
Evaluation

- **Three MVD video sequences: Kendo, Balloons, and Café**

- **For each MVD video**
  - Three cameras from the set of captured views (texture and depth)
  - Component streams encoded using CBR and VBR at different quality levels
  - Three virtual view positions within each virtual view range
  - Virtual view quality models for all supported virtual view positions
  - Two quality models for each virtual view position (*100* and *40* OPs)

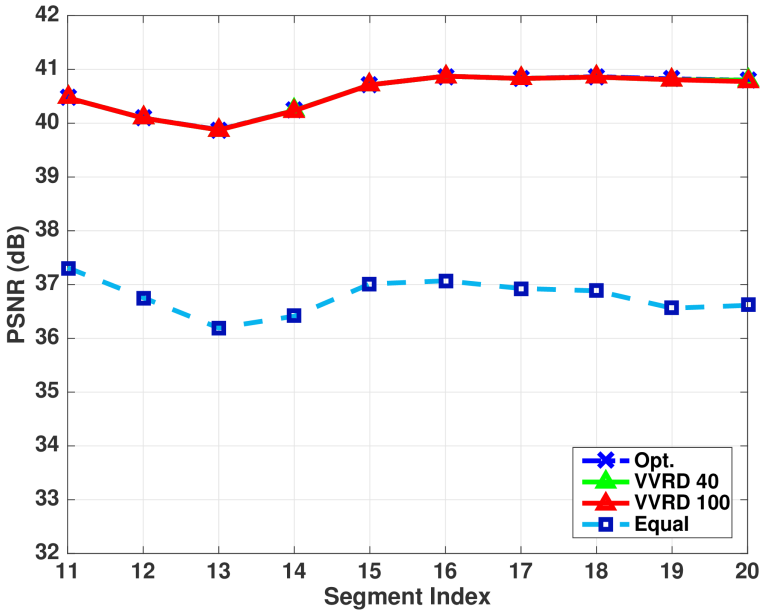
- **Subjective and objective evaluation experiments**
  - Proposed rate adaptation vs. equal allocation [Su et al. '15]

Evaluation Testbed



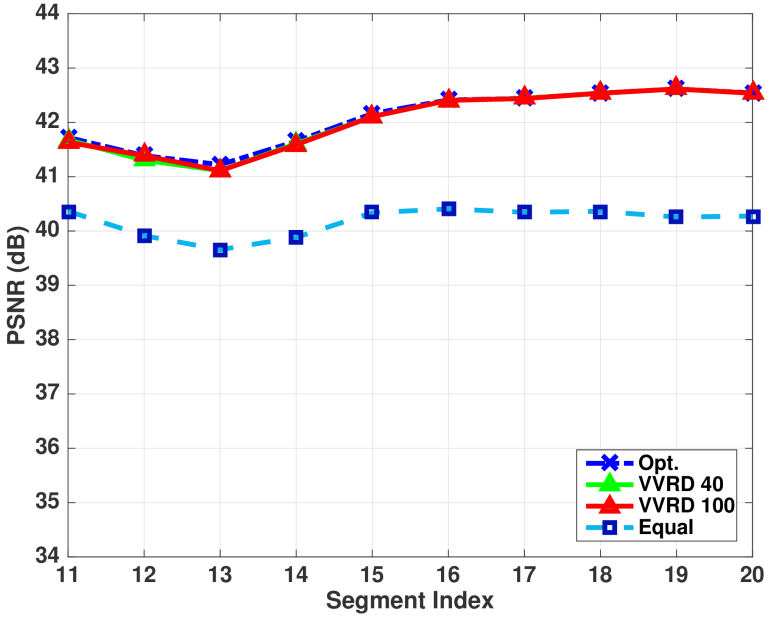
Results: Fixed Network Bandwidth

- Balloons (view 2) - CBR



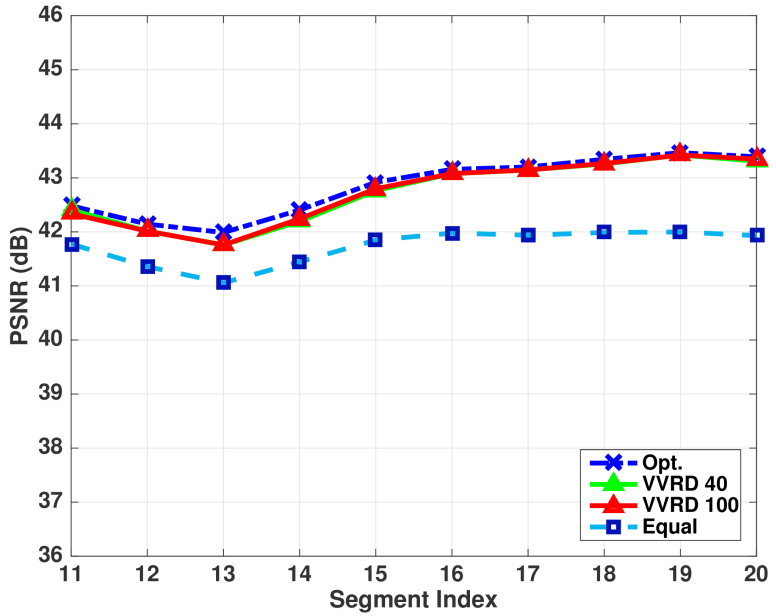
2 Mbps

≈ 4 dB



3 Mbps

≈ 2 dB

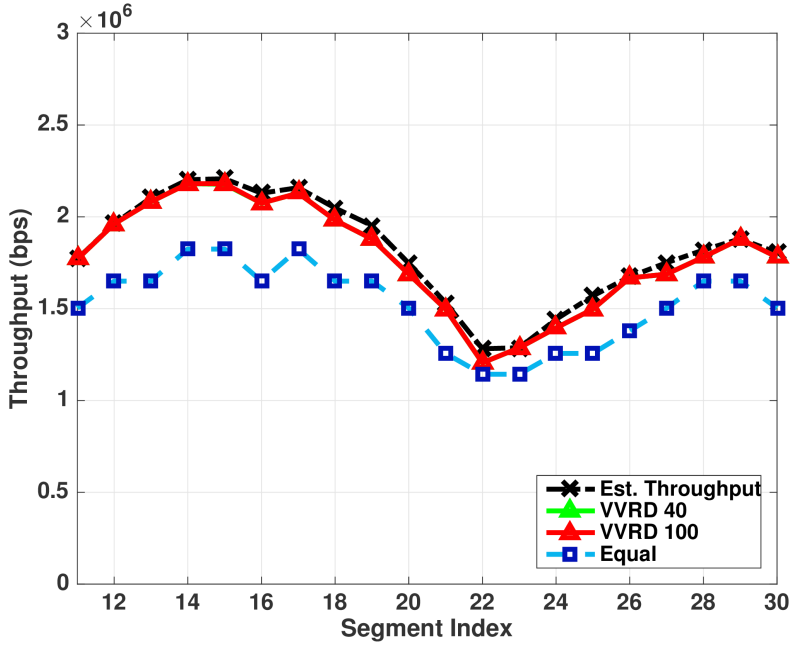


4 Mbps

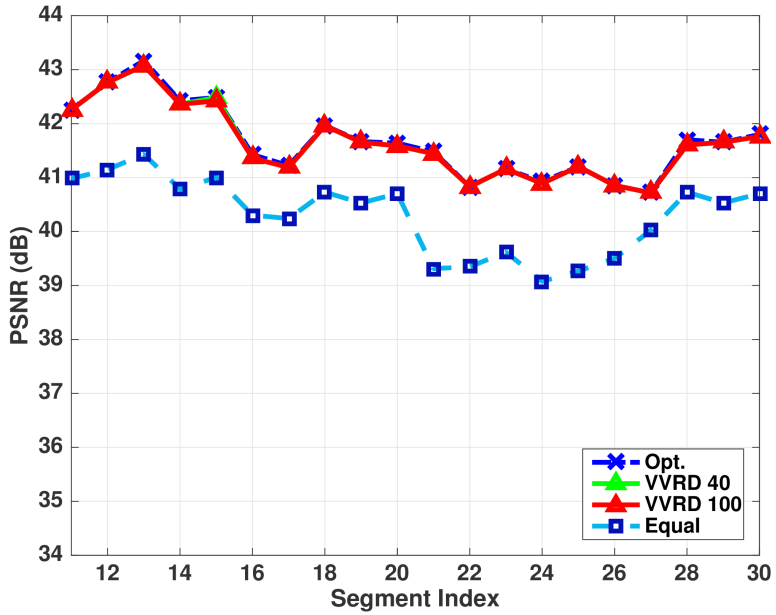
≈ 1.2 dB

Results: Variable Network Bandwidth

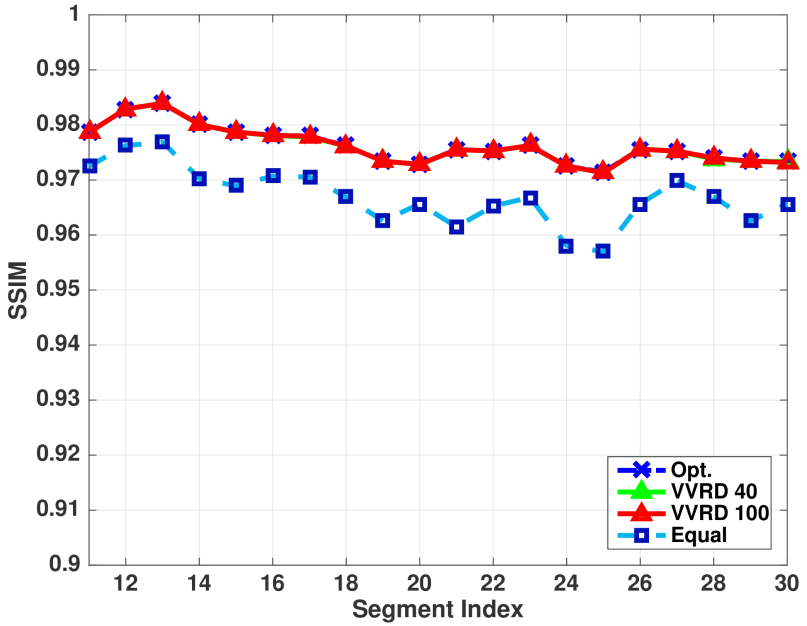
- Kendo (view 2) - VBR



Throughput



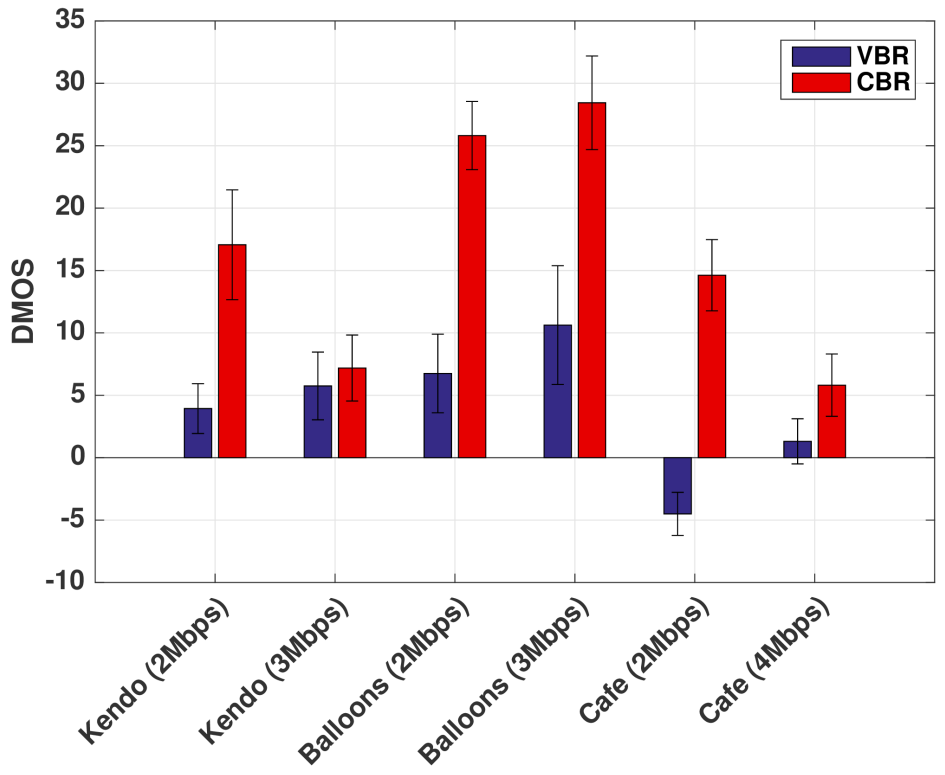
PSNR



SSIM

Subjective Assessment

- ***Double-stimulus continuous quality-scale* (DSCQS)**
  - **17 participants (12 males and 5 females)**
  - **23-33 years old**
  - **12 test conditions**
    - **3 video content**
    - **2 encoding configurations**
    - **2 bandwidth capacities**
  - **60” LG 4K Ultra HD 240Hz display**

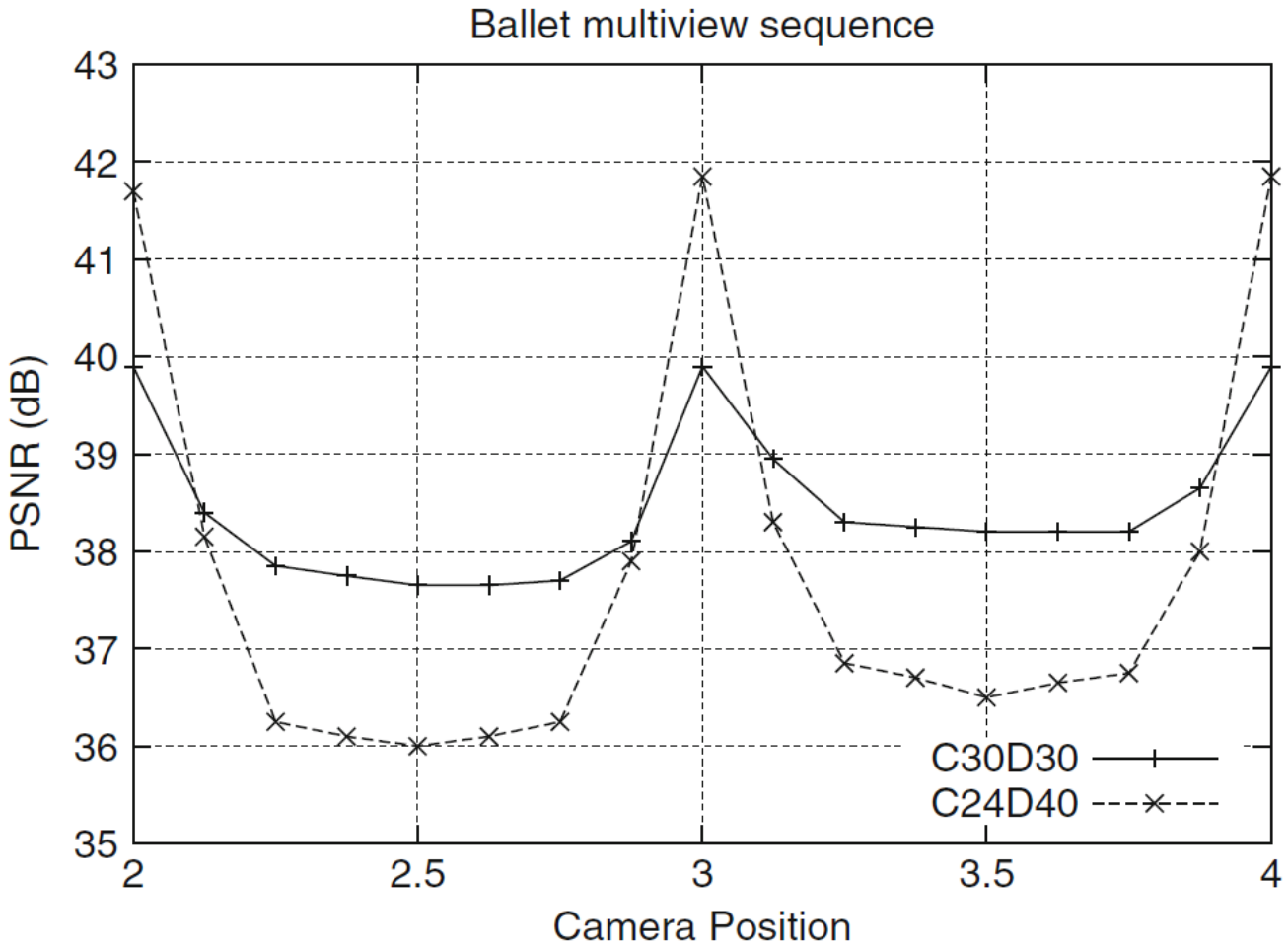


Conclusions

- **FVV streaming is interesting, but challenging to implement!**
  - **Need to efficiently utilize available bandwidth to maximize quality**
- **Virtual view quality-aware rate adaptation**
  - **Analytical quality models to reduce signaling overhead**
- **Complete system for FVV streaming and empirical results**

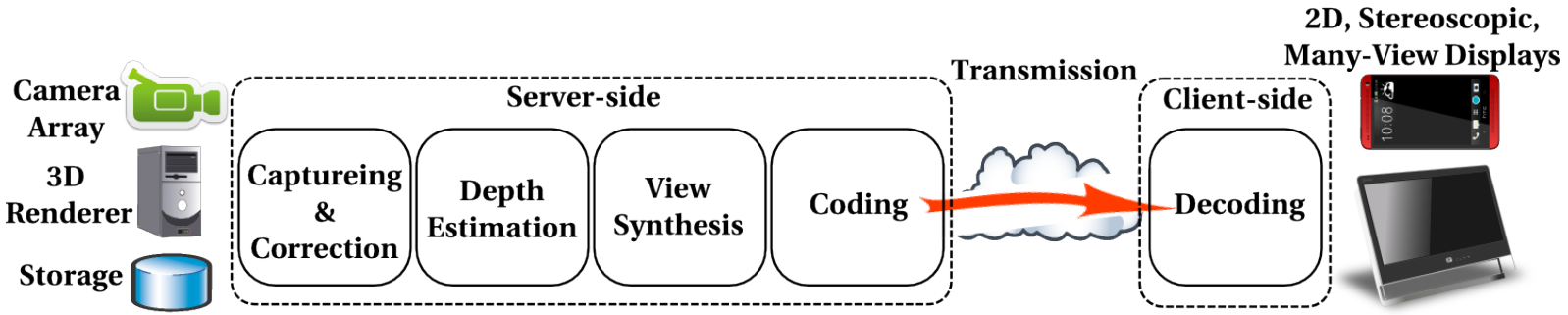
Questions?

Virtual View Quality

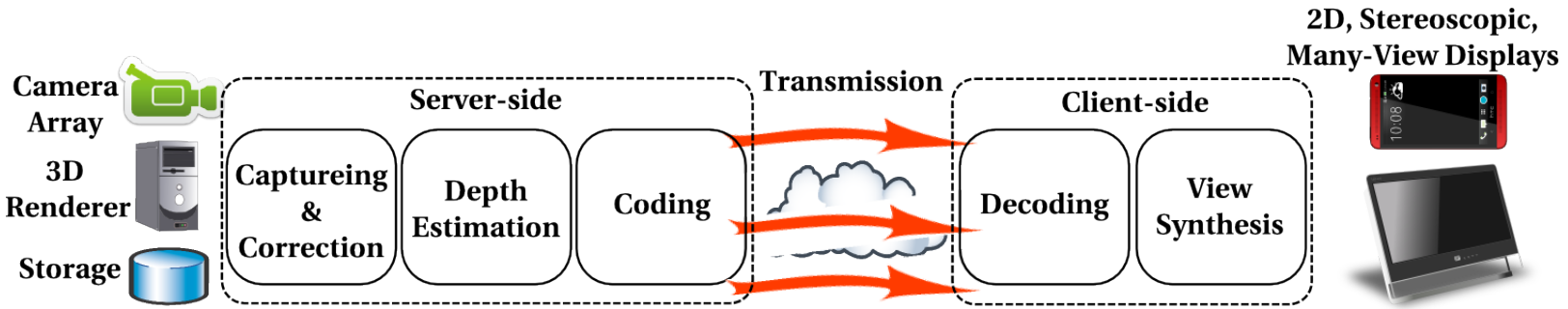


* A. Vetro, A. Tourapis, K. Müller, and T. Chen, "3D-TV content storage and transmission", IEEE transactions on broadcasting, vol 57, no 2, pp. 384–394, June 2011

FVV Streaming



Server-side rendering



Client-side rendering

References

- [AMR] http://www.alliedmarketresearch.com/3d-display-market
- [Gartner] http://www.digitaltrends.com/cool-tech/gartner-predicts-vr-growth-over-2016-and-2017
- [IDC] http://www.idc.com/getdoc.jsp?containerId=prUS41199616
- [Su et al. '15] T. Su, A. Sobhani, A. Yassine, S. Shirmohammadi, and A. Javadtalab, "A DASH-based HEVC multi-view video streaming system," *Journal of Real-Time Image Processing*, pages 1–14, 2015.