Distribution Systems for 3D Teleimmersive and Video 360 Content: Similarities and Differences

Klara Nahrstedt
Department of Computer Science
University of Illinois at Urbana-Champaign
klara@illinois.edu

ACM Multimedia Systems, June 12, 2018, Amsterdam, Netherlands
Overview

• Motivation
• 3D Teleimmersive Video Representation
• Video 360 Representation
• Similarities and Differences in Content Representation
• Distribution of 3DTI Video
• Distribution of Video 360
• Similarities and Differences in Content Distribution
• Conclusion
3D Teleimmersive (3DTI) Systems

High-End Tele-Presence Environments

Cisco Tele-presence

HP Halo

HP Colesium

UNC
Multi-Camera Live Broadcast Systems

Multi-Camera Broadcast Systems

https://www.myslingstudio.com/

https://www.spiideo.com/sports/

https://www.cinfo.es/our-products/synthetrick/multicam
360-Degree Video

3D Teleimmersive Video Representation
3D Teleimmersive Stereo Video and Free Viewpoint Video Capture
3DTI Viewing
3D Stereo Video Representation

Free-Viewpoint 3D Video Representation
View Model

source: http://zing.ncsl.nist.gov/~gseidman/vrml/
3DTI Data Model

• **3D frame** for camera $i$ at time $t$: $f_{i,t}$
  
  • Each pixel in the frame carries color+depth data and can be independently rendered

• **Stream** for camera $i$
  
  • $S_i = \{ f_{i,t1}, f_{i,t2}, \ldots \}$

• **Macro-frame**
  
  • $F_t = \{ f_{1,t}, f_{2,t}, \ldots, f_{n,t} \}$
360-Degree Video Representation
360-Degree Video

Generation of 360-Degree Video
• Capturing of multiple 2D videos together with their metadata
• Stitching videos together and further editing them in spherical video
• Encoding spherical video considering projection, interactivity, storage and delivery formats (this will impact decoding and rendering processes)
Video 360 Viewing and Navigation

Example of HDM (Head-Mounted Displays) – Oculus Rift, Samsung Gear VR, HTC Vive,

https://en.wikipedia.org/wiki/Head-mounted_display
360-Degree Video Data Model

• **Field-of-View** or **Viewport** – display region on the Head-Mounted Display
  • Fraction of omnidirectional view of the scene
  • Viewport defined by a device-specific viewing angle (typically 120 degrees) which delimits horizontally scene from head direction center, called viewport center

• **Viewport Resolution** – 4K (3840x2160) pixels
  • Resolution of full 360-degree video – at least 12K (11520x6480)

• **Video Framerate** – order of HMD refresh rate 100Hz – 100 fps

• **Motion-to-Photon Latency** requirement
  • Less than 20 ms for VR – much smaller than Internet request-reply delay
  • Need viewport prediction

• **Bitrate** – Video 360 vs HEVC (8K video at 60fps is approx. 100 Mbps)

• **Tiling** - Spatial divide of spherical video into independent tiles
Issues with Spherical Mapping to Tiles
• Viewport distortion
• Spatial quality variance

Considerations of sphere-to-plane mapping and viewing probability of tiles are IMPORTANT
• Overall spherical distortion of segment is the sum of distortion over all pixels the segment covers

Video 360 Spherical-to-Plane Projections

Equirectangular Projection – stretches poles and reduces efficiency of coding
Pyramid Projection – sees degradation on sides
Cubemap – maps 90 degree FOV to sides of cube and provides hence less degradation

Carbillon, Simon, Devlic, Chakareski, “Viewport-Adaptive Navigable 360-Degree Video delivery”, May 2017
Nasrabadi et al. “Adaptive 360-Degree Video Streaming using Scalable Video Coding”, ACM Multimedia 2017
Encoding and Delivery Formats

• **Codecs**
  • AVC/H.264, HEVC/H.265
  • VP8, VP9

• **Delivery Formats**
  • DASH/HLS (Dynamic Adaptive HTTP)

• **MPEG-DASH Standard considers tiling**
  • MPD (Media Presentation Description) – Modified for Video 360
  • SRD (Spatial Relation Description) integrated into MPD

• HEVC considers video tiles

• **MPEG – Immersive media standard ISO/IEC 23090**
  • Part 1: Use cases
  • Part 2: OMAF (Omnidirectional Media Application Format)
    • Description of equirectangular projection format
    • Metadata for interoperable rendering of 360-degree monoscopic and stereoscopic audio-visual data
    • Storage format (ISO base media file format/MP4)
    • Codecs: HEVC, MPEG0H 3D audio
  • Part 3: Immersive video
  • Part 4: Immersive Audio

Similarities and Differences of Representations
<table>
<thead>
<tr>
<th>Similarity Parameter</th>
<th>3DTI Video</th>
<th>360-Degree Video</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multi-camera Views</td>
<td>Yes (view)</td>
<td>Yes (viewport)</td>
</tr>
<tr>
<td>Joint coordinate system</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Bitrate consideration</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>View change</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Difference Parameter</th>
<th>3DTI Video</th>
<th>360-Degree Video</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Video Format</strong></td>
<td>Color-Plus-Depth</td>
<td>Color</td>
</tr>
<tr>
<td><strong>Smallest item to adapt</strong></td>
<td>3DTI frame</td>
<td>tile</td>
</tr>
<tr>
<td><strong>Frame Representation</strong></td>
<td>Frame manipulation at Pixel level (RGB, Depth, Polygons)</td>
<td>Frame manipulation at tiles and Region of Interest level</td>
</tr>
<tr>
<td><strong>Coding</strong></td>
<td>Simple zip</td>
<td>Complex HVEC</td>
</tr>
<tr>
<td><strong>Resolution</strong></td>
<td>640x480 or 1080p</td>
<td>4K to 16K</td>
</tr>
<tr>
<td><strong>Resolution for diverse devices</strong></td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Format for diverse navigation</strong></td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Distribution Systems of 3DTI Video
Multi-Camera 3DTI Transmission System

C = camera  
A = microphone  
G = gateway  
R = renderer
Approach: Multi-stream Hierarchical Adaptation
Multi-stream Adaptation (Stream Selection)

- Camera orientation: $\vec{O}_i$
- User view orientation: $\vec{O}_u$
  \[ \cos \theta = (\vec{O}_i \cdot \vec{O}_u) \]
  where $\theta$ is the angle between camera and user view

- Selection (SI) – View-Centric Stream Selection
  \[ SI = \{ i : (\vec{O}_i \cdot \vec{O}_u) \geq T, 1 \leq i \leq N \} \]
  where $T$ is a user specified parameter

View-Centric Stream Differentiation

3D capturing

3D camera

streams contributing more to user view

transmission

3D rendering

less important streams

user view

streams contributing more to user view
Timing Performance Validation

- Macro-Frame Delay at Sender side
- Macro-frame Completion Interval at Receiver Side (End-to-End Delay UIUC-UCB)
Immersive View-Centric Multi-View Multi-Party 3DTI

Multi-Party Multi-View Telepresence

- Multi-stream contents
- High resource demand
- Multi-view environment
- Multi-stream dependency
- Real-time interactivity

Example of 3D representation captured by 4 cameras
Telepresence Session Control

Decoupled control and data plane
- Hierarchical control
- Global session controller
- Local session controllers at G

Coordinated global control plane
- Monitor data plane
- Configure data plane

Data plane at TI participants
- Session routing table (SRT)
- Stream forwarding

<table>
<thead>
<tr>
<th>Matching Field (ID)</th>
<th>Forwarding Action</th>
<th>Bitrate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Site-X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Site-Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Site-Z</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

C = camera
A = microphone
G = gateway
R = renderer

Global Session Controller

Matching Field (ID) | Forwarding Action | Bitrate
ViewCast: Middleware (Overlay) Framework

A three-layer multi-party/multi-stream management framework
3D capturing 3D camera
less important streams
transmission
3D rendering
User/node’s view request

streams contributing more to user view

3D capturing

3D camera
less important streams
transmission
3D rendering
User/node’s view request

session controller

user view

request streams contributing more to user view

less important streams

3D camera
less important streams
transmission
3D rendering
User/node’s view request

3D capturing

3D camera
less important streams
transmission
3D rendering
User/node’s view request

session controller

user view

request streams contributing more to user view

less important streams

3D camera
less important streams
transmission
3D rendering
User/node’s view request

3D capturing

3D camera
less important streams
transmission
3D rendering
User/node’s view request

session controller

user view

request streams contributing more to user view

less important streams

3D camera
less important streams
transmission
3D rendering
User/node’s view request

3D capturing

3D camera
less important streams
transmission
3D rendering
User/node’s view request

session controller

user view

request streams contributing more to user view

less important streams

3D camera
less important streams
transmission
3D rendering
User/node’s view request

3D capturing

3D camera
less important streams
transmission
3D rendering
User/node’s view request

session controller

user view

request streams contributing more to user view

less important streams

3D camera
less important streams
transmission
3D rendering
User/node’s view request

3D capturing

3D camera
less important streams
transmission
3D rendering
User/node’s view request

session controller

user view

request streams contributing more to user view

less important streams

3D camera
less important streams
transmission
3D rendering
User/node’s view request

3D capturing

3D camera
less important streams
transmission
3D rendering
User/node’s view request

session controller

user view

request streams contributing more to user view

less important streams

3D camera
less important streams
transmission
3D rendering
User/node’s view request

3D capturing

3D camera
less important streams
transmission
3D rendering
User/node’s view request

session controller

user view

request streams contributing more to user view

less important streams

3D camera
less important streams
transmission
3D rendering
User/node’s view request

3D capturing

3D camera
less important streams
transmission
3D rendering
User/node’s view request

session controller

user view

request streams contributing more to user view

less important streams

3D camera
less important streams
transmission
3D rendering
User/node’s view request

3D capturing

3D camera
less important streams
transmission
3D rendering
User/node’s view request

session controller

user view

request streams contributing more to user view

less important streams

3D camera
less important streams
transmission
3D rendering
User/node’s view request

3D capturing

3D camera
less important streams
transmission
3D rendering
User/node’s view request

session controller

user view

request streams contributing more to user view

less important streams

3D camera
less important streams
transmission
3D rendering
User/node’s view request

3D capturing

3D camera
less important streams
transmission
3D rendering
User/node’s view request

session controller

user view

request streams contributing more to user view

less important streams
Why view change a problem?
Streams/View

GC = 100%, $I_i (O_i) = 24$

average 3.2 better than MC–3 performance but with 22% less rejection ratio
Immersive and Non-Immersive Multi-Party Multi-View (Live Broadcast) Systems

TI Components & Participants

- Immersive Participants
- Tight Interactivity
- Limited Scale

Producers

C = camera
A = microphone
G = gateway
R = renderer
S = sensors

Producers

Non-immersive Participants
Large Scale
View/Stream Concepts among Immersive Participants

3D capturing

3D camera

less important streams

streams contributing more to user view

3D rendering

transmission

Content Producer (Immersive Participant)

Content Producer (Immersive Participant)
View/Stream Concept among Non-Immersive Participants

3D streams → 3D streams

4D Content

\[ v_1 = [ 6 \ 5 \ 4 \ 7 \ 6 \ 5 ] \]

6 > 7 > 5 > 6 > 4 > 5
Multi-View Video among Non-Immersive Participants

Site-A

Producers

Site-B

Camera

Display

Viewers

Producers

Viewers

v_2 = \begin{bmatrix}
4 & 3 & 5 & 6 & 7 & 5
\end{bmatrix}

4 > 6 > 3 > 7 > 5 > 5
Approach: **4D TeleCast**

![Diagram of 4D TeleCast architecture]

- **Producer Tier**
  - Site-A
  - Site-B
  - Site-C

- **Viewer Tier**
  - GSC – Global Session Controller
  - LSC – Local Session Controller
  - Camera
  - Communication Gateway
  - Viewer

**Internet**
Infrastructure Management

CDN-P2P
[CDN Assisted Peer]
Wang’08, Liu’10, Chang’09
4D TeleCast

Request view: $V_1 = \{S_1, S_2, S_3\}$

Request view: $V_1 = \{S_1, S_2, S_3\}$

Request view: $V_1 = \{S_1, S_2, S_3\}$

Request view: $V_1 = \{S_1, S_2, S_3\}$
Multi-stream Dependency (Problem Description)

Victim stream

Maximum allowed delay bound = $d_{\text{buff}}$

Violation of delay bound by $d_{\text{buff}}$  →  Victim streams  →  Waste of bandwidth
Understanding E2E Delay

- Use **Delay Layer Hierarchy**

\[
\tau = \text{layer size} \\
\Delta = \text{distance from source}
\]
Distribution Systems for Video 360
Pipeline of 360-Degree Video

Challenges of 360-degree Video Distribution

• **Real-Time Stitching**

• **Simulator Sickness** in Interactivity Scenarios
  • Enable to react to HMD head movements as fast as the HMD refresh rate (120 Hz)

• **Viewport extraction in real-time**
  • Challenge: difficult to predict user orientation for more than 3 seconds
  • Challenge: if short-term prediction is needed, how do we avoid rebuffering/stall under small playout buffers?

• **Avoidance of bandwidth waste** (if one downloads viewports that are not needed)
  • Tiles prefetching error
MPEG-DASH Video Distribution System for Single 2D Video Stream

MPEG-DASH Video 360 Video Streaming using Tiles

360-Video Streaming Systems

• Tiling for Adaptive Streaming
  • Video divided into tiles
  • Depending on the mapping of spherical video projection, different tiles will be streamed
  • Tiles currently viewed by users are streamed at high quality and the rest with low resolution

• Personalized Viewport-Only Streaming – Asymmetric Panorama viewing
  • Also called asymmetric panorama viewport adaptive streaming
  • Methods: Truncated Pyramid Projection (TSP), Cubemap
  • Video divided into segments
  • When client moves head, the viewport center changes and new viewport must be display
  • Decrease of bitrate without decrease of quality of viewport

Tile-based HTTP Adaptive Streaming and Head Movement Prediction

Tile-based HTTP Adaptive Streaming for 360 Video
ERP – Raw Panoramic Video
• ERP is divided into video chunks
• Each chunk is cropped into N tiles, indexed in raster-scan order
• Each tile is encoded into segments with M bit-rate levels
• MxN optional segments stored at server and ready for pre-fetching and streaming
360ProbDASH Approach

- Pre-fetch Segments by predicting viewport
  - Use probabilistic model for prediction
  - Leverage Linear Regression Prediction of Orientation

\[
\begin{align*}
\hat{\alpha}(t_0 + \delta) &= m_\alpha \delta + \alpha(t_0), \\
\hat{\beta}(t_0 + \delta) &= m_\beta \delta + \beta(t_0), \\
\hat{\gamma}(t_0 + \delta) &= m_\gamma \delta + \gamma(t_0).
\end{align*}
\]

Yaw prediction
Pitch prediction
Roll prediction

- Distribution of Prediction Errors
  - Long-term predictions are hard
  - 5 users data collection for short term prediction error (3 seconds)
Tile-based Adaptive Video Streaming

- **Ochi** et al use tile-based streaming where spherical video is mapped to equirectangular video and video is cut into 8x8 tiles

- **Hosseini and Swaminathan** use hexa-face sphere-based tiling of 360° degree video to take into account projection distortion
  - Description of tiles with MPEG-DASH Spatial Relation Description

- **Quan** et al use prediction of head movement to deliver tiles

**Weaknesses of Tiling systems**
- Time and energy consuming reconstruction
- Coding inefficiency due to independent tiling
- Server management of files is difficult due to large amount of quality levels and large MPD files
- Client selection process is complex
- Mixed bit-rate tiles can result in visible border and quality inconsistence in combined-tiles rendering
- Multiple Decoders


QER Viewport-Adaptive Streaming

Carbillon, Simon, Devlic, Chakareski, “Viewport-Adaptive Navigable 360-Degree Video delivery”, May 2017
Viewport Adaptive Streaming System

Carillon, Simon, Devlic, Chakareski, “Viewport-Adaptive Navigable 360-Degree Video delivery”, May 2017
Approach: QER - Quality Emphasized Region

• Not only bit-rate adaptation but also **QER server adaptation** where different regions have different quality
  • QER – Quality Enhanced Region

• Each QER is represented by **Quality Emphasis Center (QEC)**
  • Full video gets delivered in certain projection representation (equirectangular, cube, ..), but it has different versions of video QEC
  • Client device selects the right representation and extracts viewport

• Viewport-adaptive streaming similar to DASH
  • Client runs adaptation algorithm to select video representation; **selects QER and QEC of available QER**
  • QEC selection is based on **smallest orthodromic distance**
    • Orthodromic distance – shortest distance between two points on surface of sphere, measured along surface of sphere

• **Video segment length**
  • Temporal Chunk sent from server – 1-10 seconds
  • Tradeoff between short and long segments

• **Expanded MPD**
  • MPD file expanded with new information
    • Coordinates of its QEC in degrees
      • Two angles (0,360) degrees and (-90,90) degrees
    • All representations assume the same reference coordinate system

```xml
<?xml version="1.0"?>
<MPD>
  <Representation id="1" gec="90,60" bandwidth="9876" width="1920" height="1080" frameRate="30">
    <EssentialProperty schemedUri="urn:mpeg:dash:vrd:2017" value="0,0"/>
  </Representation>
</AdaptationSet>
</MPD>
```
QER-Based Viewport Adaptive Streaming

Carillon, Simon, Devlic, Chakareski, “Viewport-Adaptive Navigable 360-Degree Video delivery”, May 2017
Examples of Experimental Results

- Metric to extract viewport – (1) MS-SSIM: Multi-Scale Structural Similarity and (2) PSNR
- Original equirectangular video of full quality - 4K video with 1080p resolution
- QEC - in center of face encoded with best quality, other faces at 25% of full quality
- Distance - for \( d = 0 \), QEC and viewport center match 0.98; as \( d \) increases, quality decreases
- QEC numbers - With increased QEC number, quality increases; shorter segments are better
Similarities and Differences of Distribution Systems
<table>
<thead>
<tr>
<th>Similarity Parameter</th>
<th>3DTI Video</th>
<th>360-Degree Video</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dealing with Bandwidth</td>
<td>Adapt Views</td>
<td>Adapt Viewports</td>
</tr>
<tr>
<td>View change</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Navigation</td>
<td>Via mouse yes</td>
<td>Via mouse yes</td>
</tr>
<tr>
<td>Client adaptation</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Streaming Protocols</td>
<td>TCP-based</td>
<td>TCP-based</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Difference Parameter</th>
<th>3DTI Video</th>
<th>360-Degree Video</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dealing with Bandwidth</td>
<td>Adapt Views/Streams</td>
<td>Adapt Viewports/Tiles</td>
</tr>
<tr>
<td>Encoding Standards</td>
<td>zip/some efforts in MPEG/OMAF on 3DTI compression</td>
<td>MPEG-DASH considers omnidirectional video tiles</td>
</tr>
<tr>
<td>Distribution Style</td>
<td>Real-time view-based telepresence style or live view-based broadcast</td>
<td>On-demand DASH-style</td>
</tr>
<tr>
<td>Clients</td>
<td>homogeneous</td>
<td>heterogeneous</td>
</tr>
<tr>
<td>Viewing</td>
<td>Flat 2D or 3D displays</td>
<td>Head-Mounted Displays</td>
</tr>
<tr>
<td>Streaming Protocols</td>
<td>TCP-Based</td>
<td>HTTP-based Standard MPEG-DASH</td>
</tr>
<tr>
<td>Navigation</td>
<td>Via mouse only</td>
<td>Via mouse, head movement, hand movement</td>
</tr>
</tbody>
</table>
Conclusion and Summary

• **360-degree video is becoming possible for**
  - 3D teleimmersive video or
  - Omnidirectional video

• **First solutions are coming up in terms of**
  - capture, encoding and viewing

• **But distribution represents challenge**
  - Real-time live streaming or
  - Near-real-time distribution of 360-degree video

• A lot of presented material will be published in a survey paper
  - “Scalable 36-Degree Video Streaming: Challenges, Solutions and Opportunities”
  - Authors: Michael Zink, Ramesh Sitaraman, Klara Nahrstedt
  - Journal Venue: Proceedings of IEEE Special Issue
  - Editors: Boris Koldehoffe, Ralf Steinmetz, ...
  - Coming up in early 2019