MPEG IMMERSIVE VIDEO (MIV) STANDARD

Jill Boyce and Basel Salahieh

Mile High Video 2020
MPEG Immersive video (MIV)

Under development standard to enable viewing of immersive volumetric video content with viewer 6 degrees of freedom (6DoF)
Immersive content can be consumed using wide variety of devices

- Viewer feels immersed within a remote (or virtual) 3D scene
- Many of these devices are widely available now
- Range of viewer motion limited by range of camera capture
MIV Input Format

- Volumetric video scene captured from multiple cameras, at different positions
- Texture + Depth from each camera
- Any camera arrangement supported, of real or virtual cameras
Example MIV test content
Depth (Geometry): synthetic vs. natural

Depth estimation not standardized, but necessary
MPEG Immersive Video (MIV) standard

- Specification is based upon MPEG V3C/V-PCC (Point Cloud Coding) and extends it
- Uses standard video codecs (AVC, HEVC, VVC, ...)
- Only MIV decoder is normative. Implementations may customize encoder, depth estimation, view synthesis/renderer, video codec.
  - Reference software available for non-normative components (TMIV has enc, renderer)
MIV Encoding & Decoding

Source Views (T+D) → MIV Encoder → MIV Decoder & Renderer → Display
MIV features & benefits

Any number of cameras (real or virtual) and any camera arrangement
- Level limits on pixel rate, atlas size, # of atlases to constrain complexity
- Any projection format: perspective, orthographic, spherical ERP, spherical cube maps...
- Can update camera parameters & depth range per view to handle moving camera rigs & objects

Encoder can select size and number of atlases, patches, groups
- Can cluster cameras into different groups to enhance local coherency and substream processing
- Can support fine details (i.e., hair, smoke, grass) better than point clouds, mesh
- Preserves specularities and efficiently handles patches with fixed depth

Can be used with any video codec: HEVC, HEVC SCC, AV1, VVC, AVC

High level syntax to indicate alignment of atlases with camera views, to enable viewport dependent streaming, substream access for decoding and rendering
MIV Operating Modes

*MIV Atlas*

*MIV View*

*MIV Entity*

*MIV Geometry Absent*
MVIV frame packing

- MIV can combine texture, geometry/depth, and external occupancy (from multiple cameras) in same packed frame, to enable use of single video stream

MVIV w/ explicit occupancy, 2 atlases packed in 1 frame:
- Texture atlases at full resolution
- Depth downscaled by 2x2
- Occupancy downscaled by 4x4
MIV and V3C/V-PCC

• Key similarities:
  • Much shared syntax: MIV normatively references the V3C specification and extends it
  • Patches, atlases
  • Use of any video codec standard
  • Same systems layer mappings

• Key differences:
  • MIV supports flexible camera arrangements, signaling of camera parameters
  • MIV inputs and outputs are videos, not point clouds
  • Occupancy can be embedded in geometry or signaled explicitly
  • Additional syntax to support variety of modes
  • Restrictions to reduce implementation complexity, e.g. requires time alignment across components

• Would be straightforward to support both standards in same implementation
Intel media:
Low power, high performance dedicated fixed-function HEVC video decoder

Intel graphics:
View synthesis algorithms for improved video quality, high performance
Conclusion

• MIV standard designed for easy deployment
  • Uses legacy video codecs
  • Uses GPU rendering
  • Can be viewed on current displays
• Variety of modes supported to target wide range of applications
• Technical challenges
  • Depth estimation
  • Encoding algorithms