# Camera Localization and 3D Mapping for Reference Picture Synthesis in VVC

Hossein Golestani

RWTH Aachen University, Germany

# Outline

- Motivation
- The proposed method
- Simulation results & Analysis
- Conclusion

# Outline

- Motivation
- The proposed method
- Simulation results & Analysis
- Conclusion

# Target Videos

- Videos sequences captured by moving cameras.

# More Videos are Coming …

Credit: Uber


Credit: University of Pennsylvania


Credit: medium.com

Credit: nofilmschool.com


Credit: BookingHunterTV (YTube Channel)

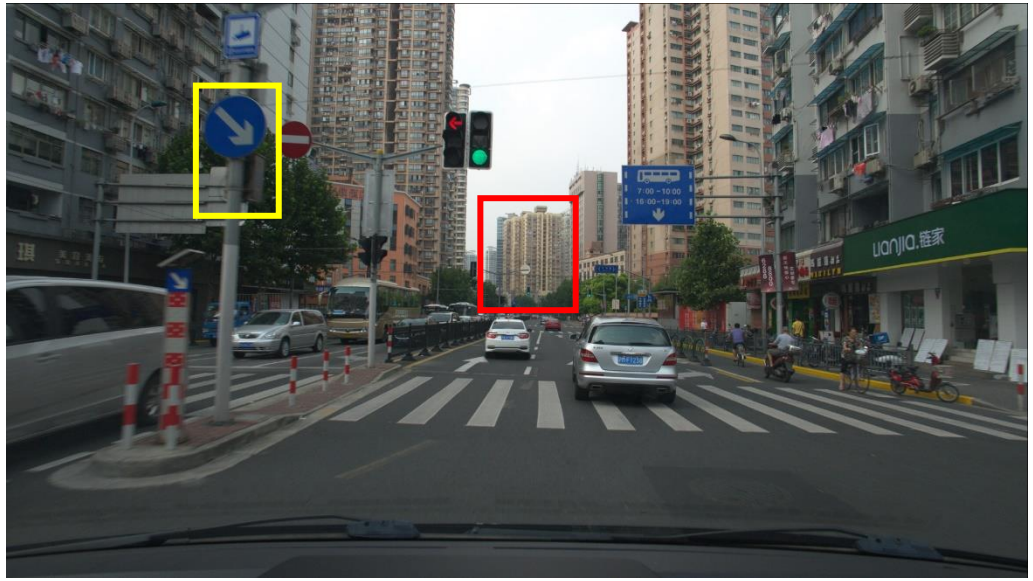
Credit: Bald and Bankrupt (YTube Channel)

# A question …

- Are 2D motion models (translational and affine) efficient enough for those videos?

# Thinking about the original 3D environment …

- If we have a 3D understanding of the scene, we can do a better motion compensation.
- We need
  - 3D scene geometry
  - Camera Parameters (intrinsic and 6DoF-poses)

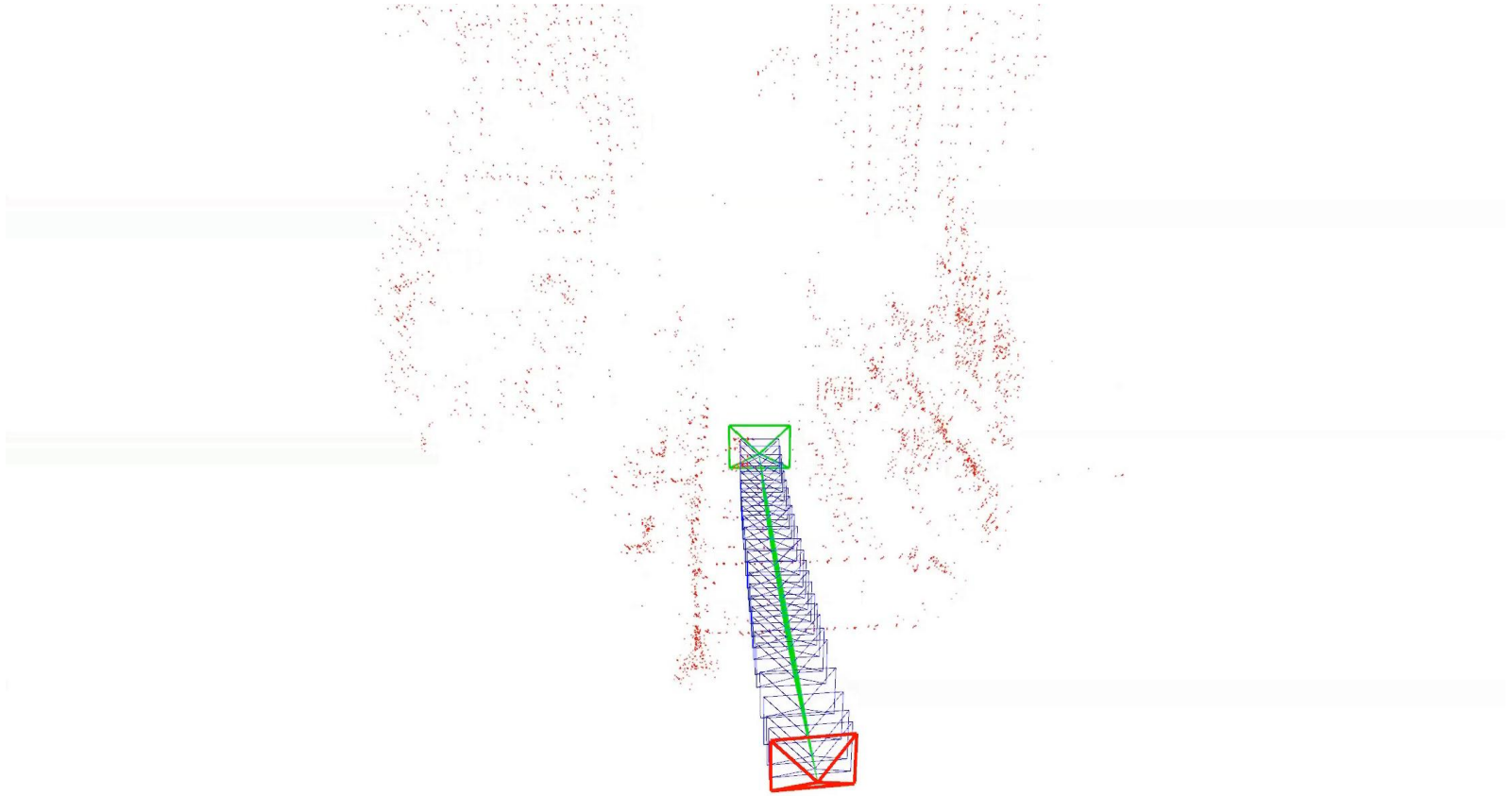


(a) The reference frame, t=T-1

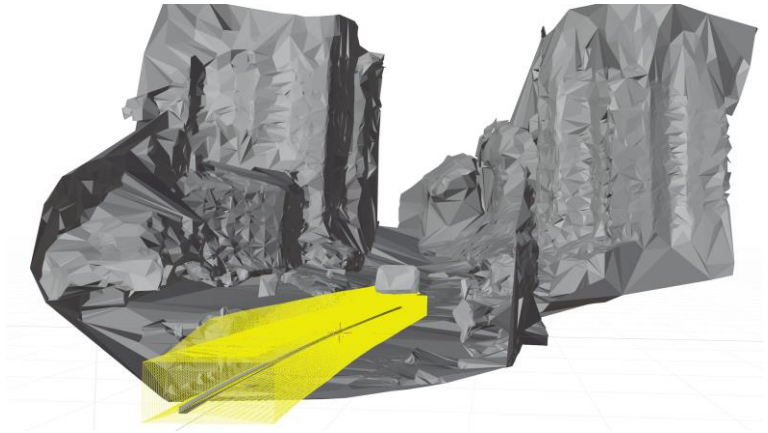
(b) the current frame, t=T

# Extracting 3D data from 2D videos

(a) Input DayLightRoad Video Sequence



(b) Sparse Reconstruction



(c) Dense Reconstruction

# Some Questions and Answers

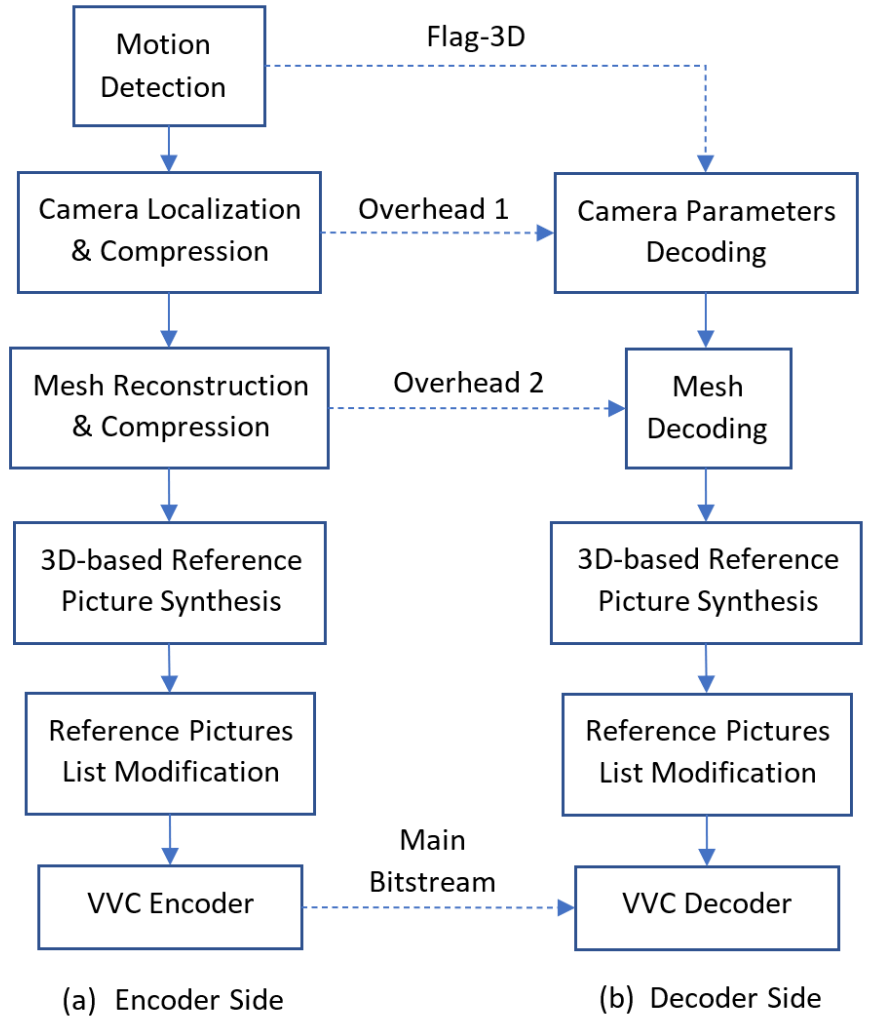
- How to extract 3D data from 2D videos?
- How to use the estimated/captured data for motion compensation?
- How much coding gain can be achieved compared to latest standard?
- How complex is the system?

# Outline

- Motivation
- The proposed method
- Simulation results & Analysis
- Conclusion

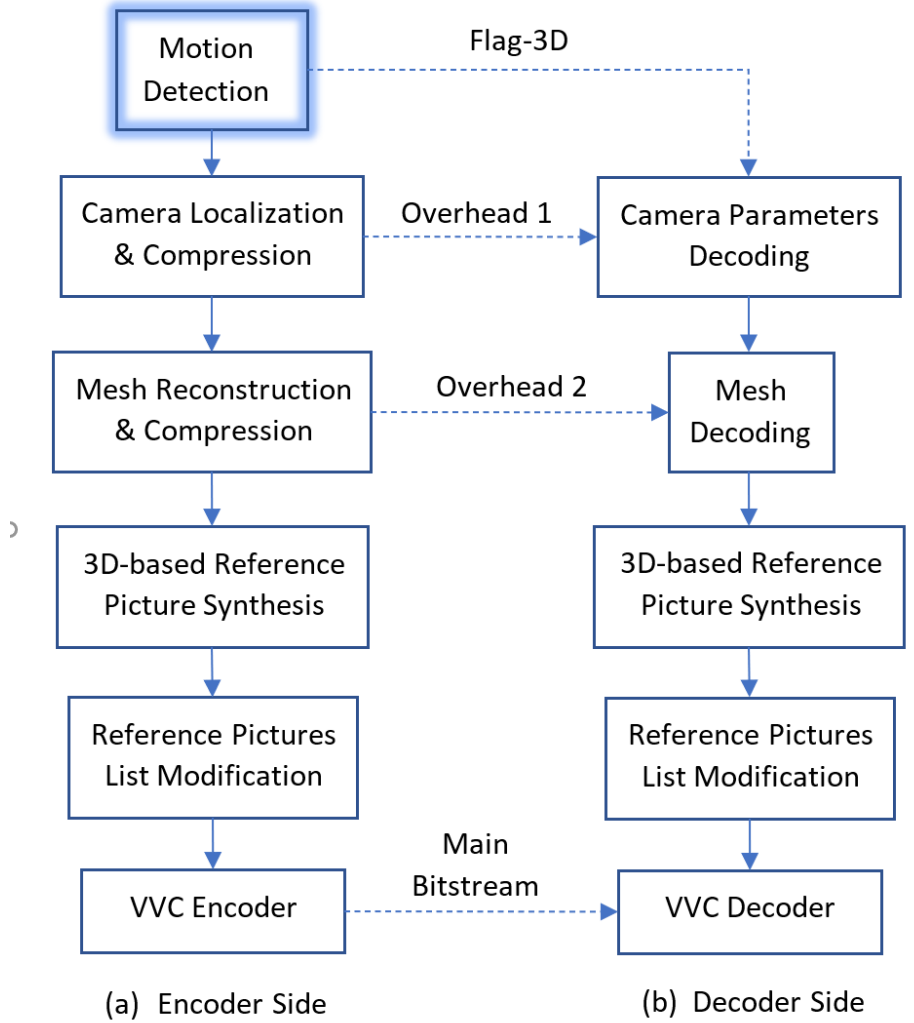
# The proposed method

- Main steps
  - Motion detection (Flag-3D)
  - Camera Localization and Compression (Overhead 1)
  - 3D Mesh Reconstruction and Compression (Overhead 2)
  - 3D-based Reference Picture Synthesis
  - VVC Reference Picture Lists Modifications
  - VVC Encoding



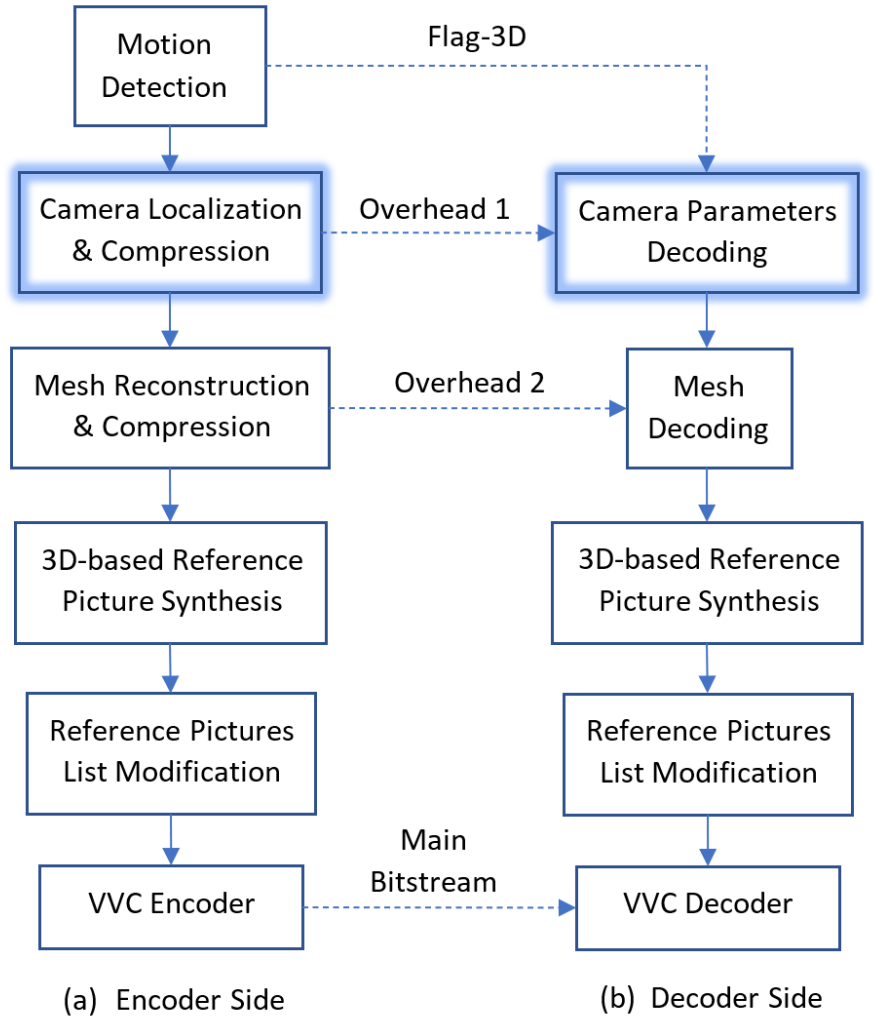
# The proposed method

- Main steps
  - Motion detection (Flag-3D)
  - Camera Localization and Compression (Overhead 1)
  - 3D Mesh Reconstruction and Compression (Overhead 2)
  - 3D-based Reference Picture Synthesis
  - VVC Reference Picture Lists Modifications
  - VVC Encoding



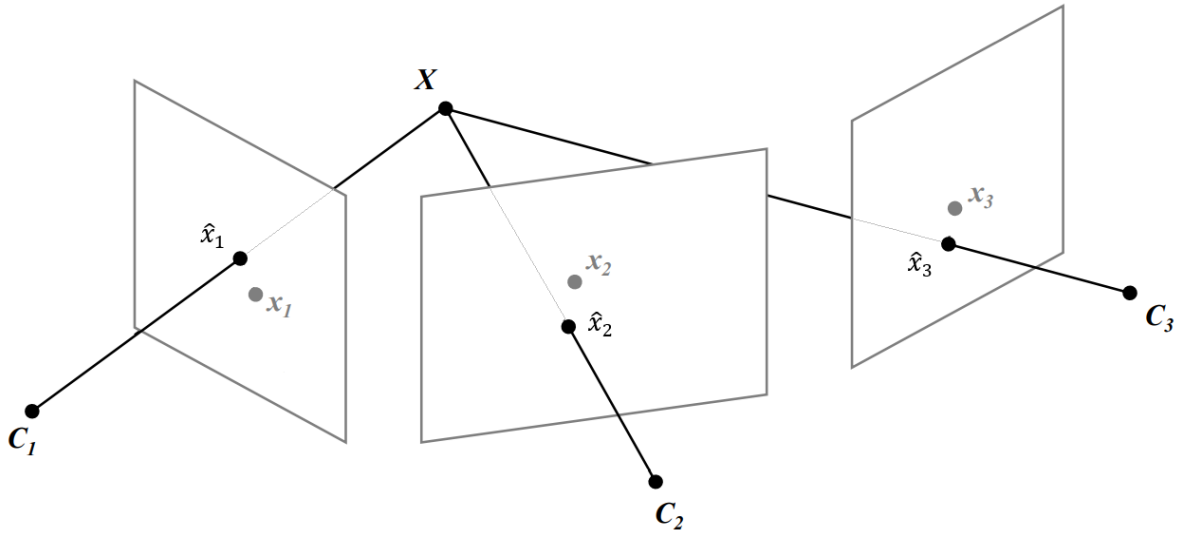
# The proposed method

- Main steps
  - Motion detection (Flag-3D)
  - Camera calibration and Compression (Overhead 1)
  - 3D Mesh Reconstruction and Compression (Overhead 2)
  - 3D-based Reference Picture Synthesis
  - VVC Reference Picture Lists Modifications
  - VVC Encoding



# Camera Calibration using incremental Structure-from-Motion

- Incremental Structure from Motion (SfM) [1]
  - Input: Images $\mathcal{I} = \{I_i \mid i = 1,2,\dots,N_I\}$
  - Output:
    - poses $\mathcal{P} = \{P_c \in \mathrm{SE}(3) \mid c = 1,2,\dots,N_P\}$
    - Reconstructed points $\mathcal{X} = \{X_k \in \mathbb{R}^3 \mid k = 1,2,\dots,N_X\}$



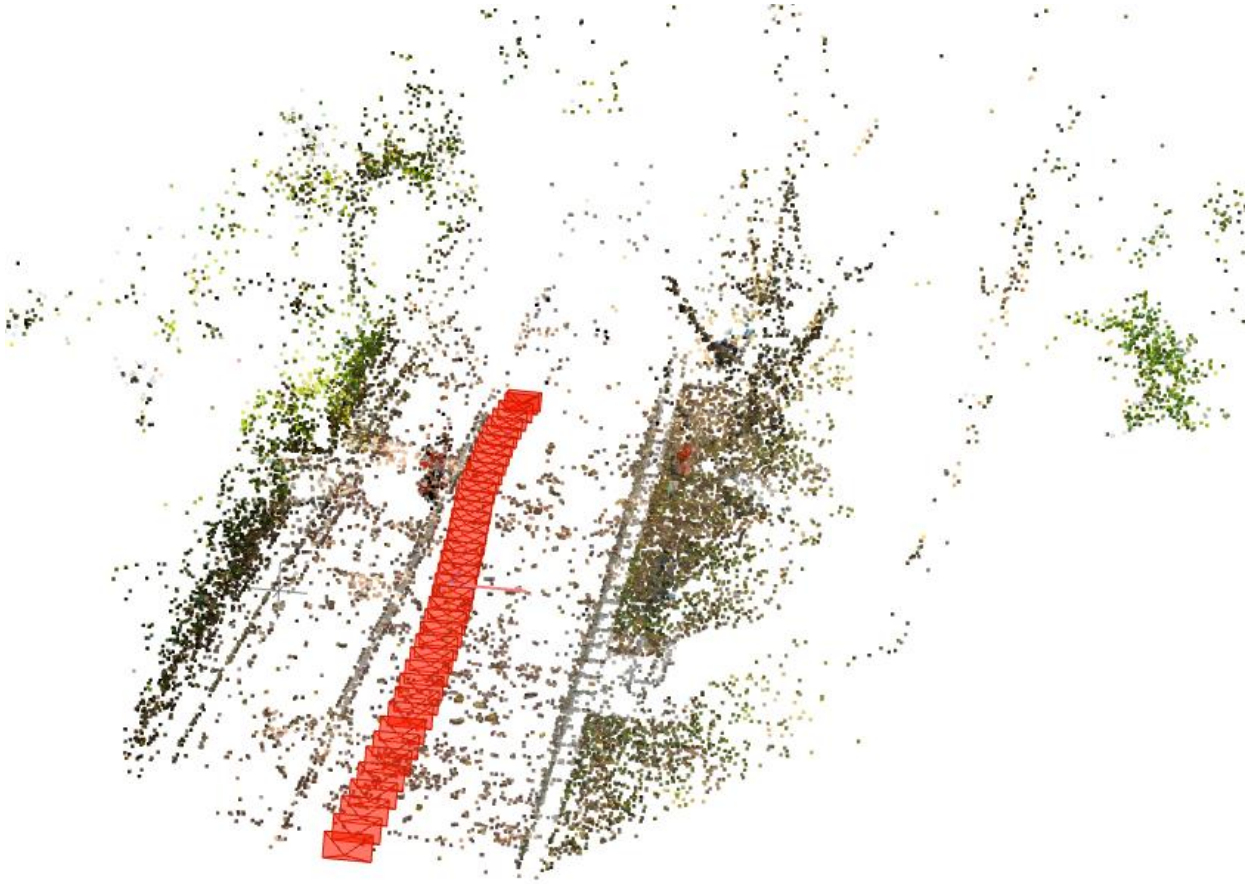

Steps

(1) Correspondence Search

(2) Incremental Reconstruction

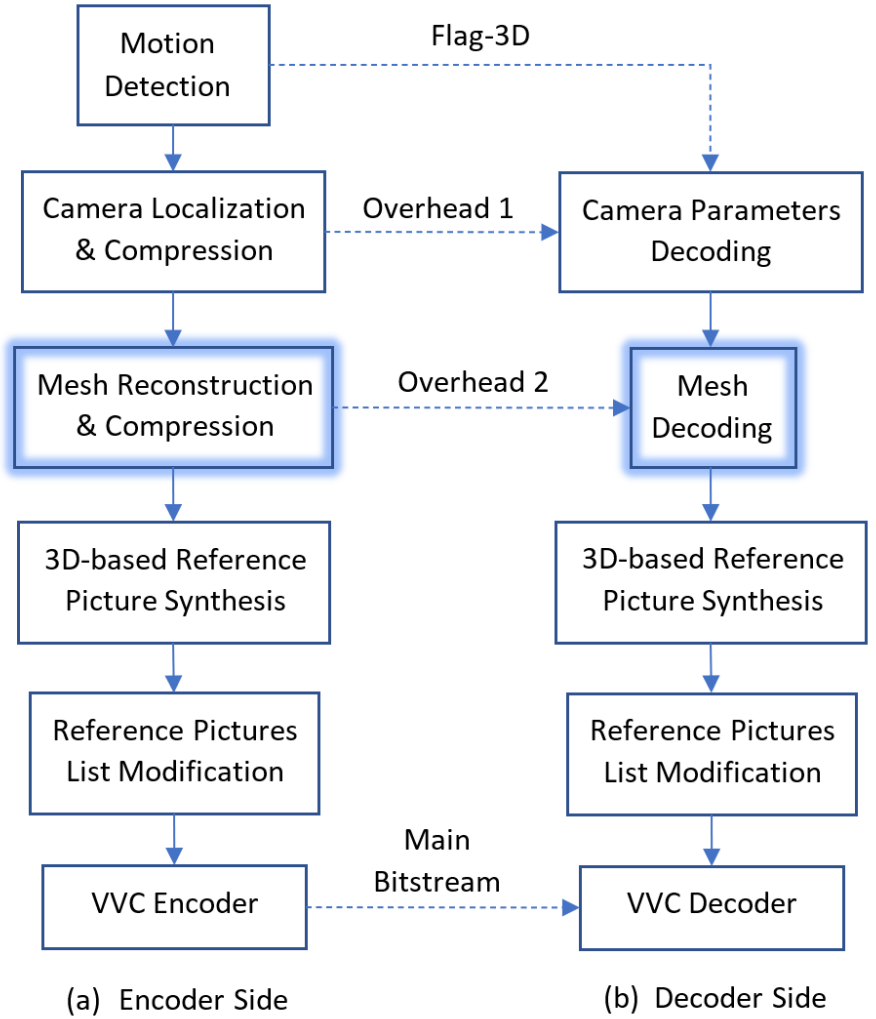

[1] Andrew, Alex M. "Multiple view geometry in computer vision." *Kybernetes* (2001).

# An example …



# The proposed method

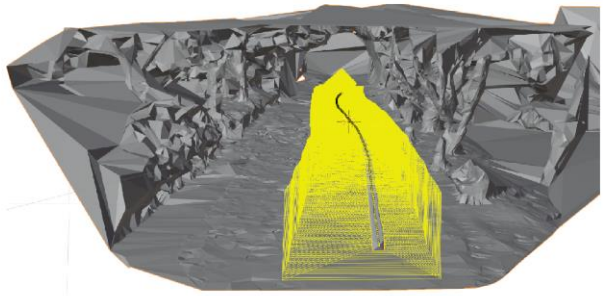
- Main steps
  - Motion detection (Flag-3D)
  - Camera Localization and Compression (Overhead 1)
  - 3D Mesh Reconstruction and Compression (Overhead 2)
  - 3D-based Reference Picture Synthesis
  - VVC Reference Picture Lists Modifications
  - VVC Encoding



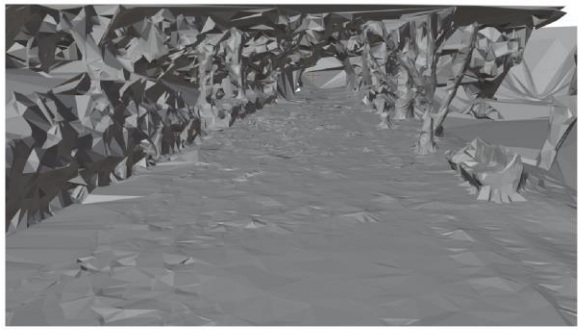
# Mesh Reconstruction and Compression

- Mesh Reconstruction [2]
  - Depth map estimation
  - Filtering and fusion
- Mesh Decimation [3]
  - Reducing the number of vertices with minimal shape changes
- Mesh Compression [4]
  - Geometry compression
  - Connectivity compression
- An important trade-off between
  - Mesh fidelity
  - The overhead of sending mesh

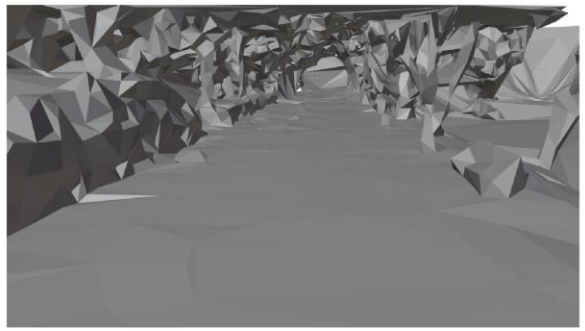
(a) ParkRunning Sequence



(b) 3D Geometry for ParkRunning



(c) Original Mesh (1095 kB)

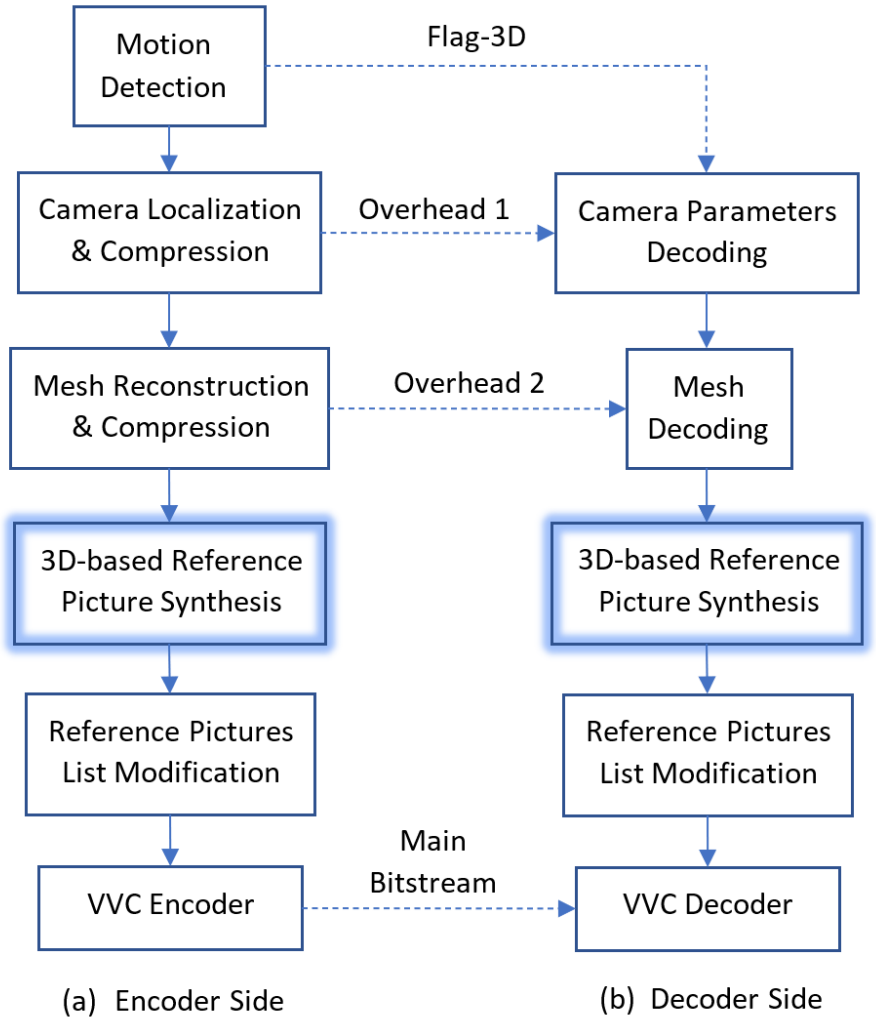


(d) Decimated and Compressed Mesh (14.5 kB)


[2] Zheng, Enliang, et al. "Patchmatch based joint view selection and depthmap estimation." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2014.
[3] M. Garland, Y. Zhou, “Quadric-based simplification in any dimension”, ACM Trans. Graph- 24, 2 (2005), pp: 209-239.
[4] Draco, Accessed 01.10.2020 [Online], Available: https://github.com/google/draco

# The proposed method

- Main steps
  - Motion detection (Flag-3D)
  - Camera Localization and Compression (Overhead 1)
  - 3D Mesh Reconstruction and Compression (Overhead 2)
  - 3D-based Reference Picture Synthesis
  - VVC Reference Picture Lists Modifications
  - VVC Encoding



# Mesh-based Reference Picture Synthesis

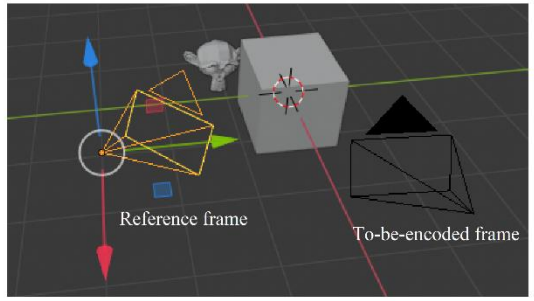
(a) To-be-encoded frame

(b) Reference frame

# Mesh-based Reference Picture Synthesis

(a) To-be-encoded frame

(b) Reference frame

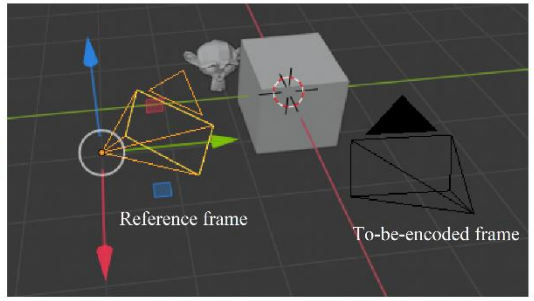



(c) 3D setup

# Mesh-based Reference Picture Synthesis

(a) To-be-encoded frame

(b) Reference frame



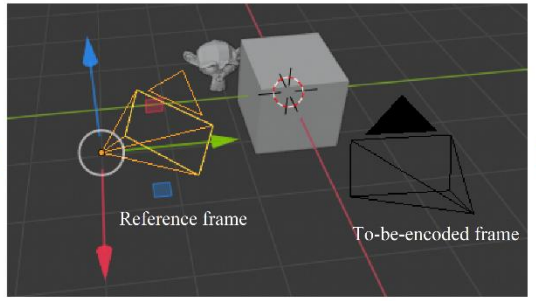

(c) 3D setup

(d) Synthesized reference picture

# Mesh-based Reference Picture Synthesis

(a) To-be-encoded frame

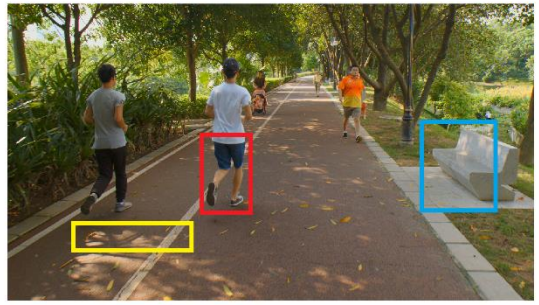
(b) Reference frame



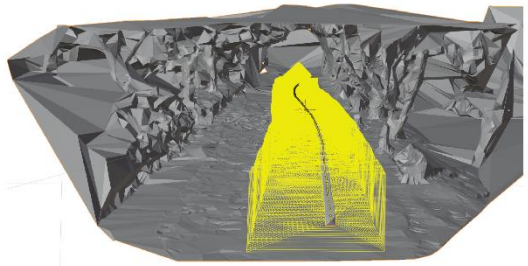

(c) 3D setup

(d) Synthesized reference picture

(a) To-be-encoded frame



(b) Reference frame

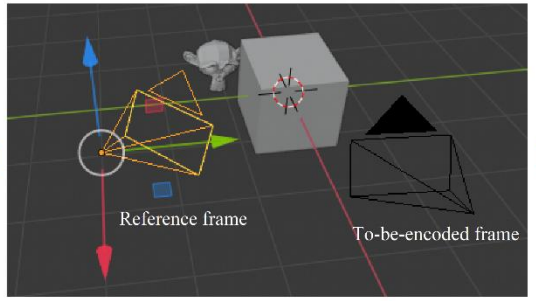


(c) 3D setup

# Mesh-based Reference Picture Synthesis

(a) To-be-encoded frame

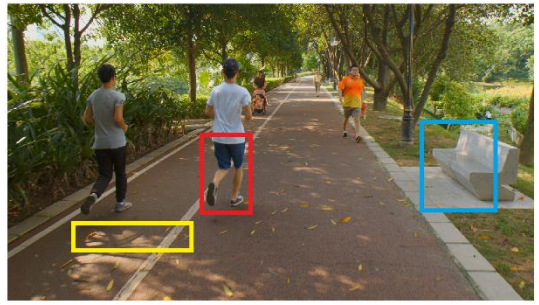
(b) Reference frame



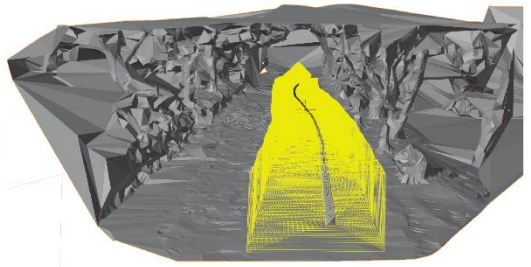

(c) 3D setup

(d) Synthesized reference picture

(a) To-be-encoded frame



(b) Reference frame

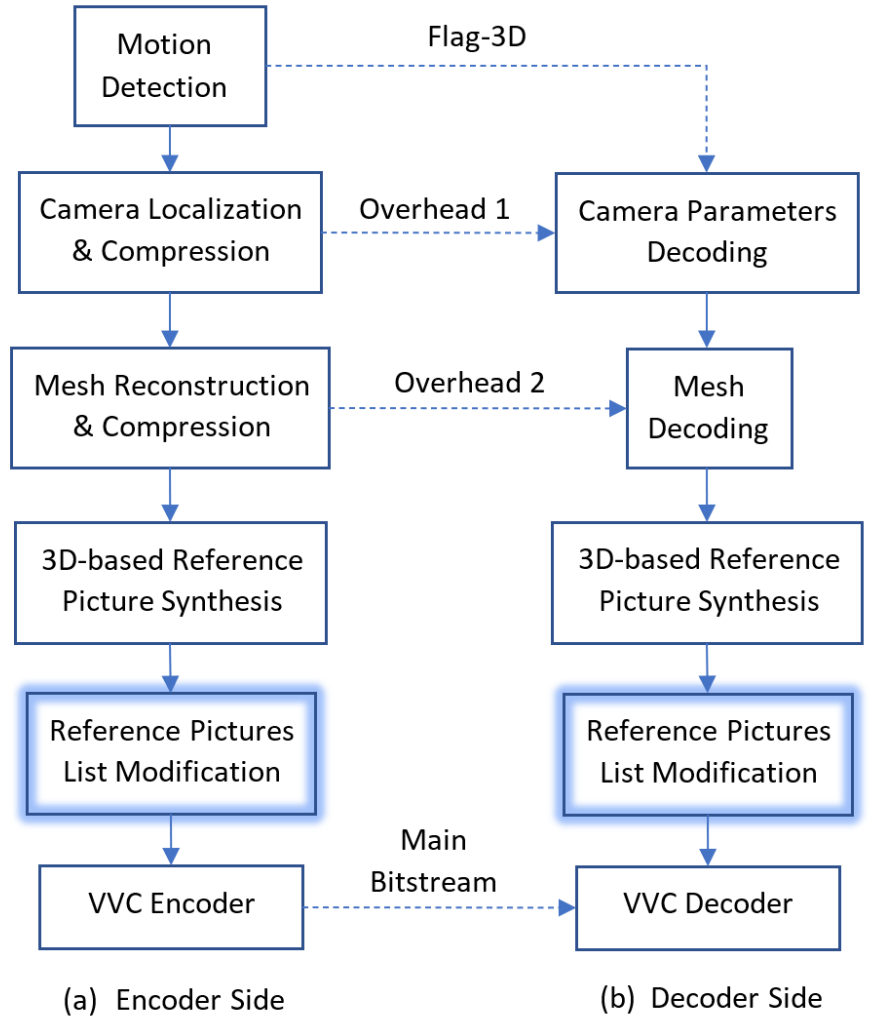


(c) 3D setup

(d) Synthesized reference picture

# The proposed method

- Main steps
  - Motion detection (Flag-3D)
  - Camera Localization and Compression (Overhead 1)
  - 3D Mesh Reconstruction and Compression (Overhead 2)
  - 3D-based Reference Picture Synthesis
  - VVC Reference Picture Lists Modifications
  - VVC Encoding



# Reference Pictures Lists (RPLs) Modifications

- Reference Pictures Lists (RPLs) Modifications
  - Replaced Reference Mode
  - Additional Reference Mode

VTM: POC10 [L0 8 0] [L1 12c 8]

- VVC applies its 2D motion models to all RPs (including the 3D-based RPs) and decides which RP should be selected for each CU through rate-distortion optimization.

# Reference Pictures Lists (RPLs) Modifications

- Reference Pictures Lists (RPLs) Modifications
  - Replaced Reference Mode
  - Additional Reference Mode

| | |
|---|---|
| VTM: | POC10 [L0 8 0] [L1 12c 8] |
| RR-Mode: | POC10 [L0 8 0] [L1 12c 8] |

- VVC applies its 2D motion models to all RPs (including the 3D-based RPs) and decides which RP should be selected for each CU through rate-distortion optimization.

# Reference Pictures Lists (RPLs) Modifications

- Reference Pictures Lists (RPLs) Modifications
  - Replaced Reference Mode
  - Additional Reference Mode

VTM: POC10 [L0 8 0] [L1 12c 8]

RR-Mode: POC10 [L0 8 ~~0~~] [L1 12c ~~8~~]

- VVC applies its 2D motion models to all RPs (including the 3D-based RPs) and decides which RP should be selected for each CU through rate-distortion optimization.

# Reference Pictures Lists (RPLs) Modifications

- Reference Pictures Lists (RPLs) Modifications
  - Replaced Reference Mode
  - Additional Reference Mode

VTM: POC10 [L0 8 0] [L1 12c 8]

RR-Mode: POC10 [L0 8 **3D-Ref**] [L1 12c **3D-Ref**]

- VVC applies its 2D motion models to all RPs (including the 3D-based RPs) and decides which RP should be selected for each CU through rate-distortion optimization.

# Reference Pictures Lists (RPLs) Modifications

- Reference Pictures Lists (RPLs) Modifications
  - Replaced Reference Mode
  - Additional Reference Mode

| | |
|---|---|
| VTM: | POC10 [L0 8 0] [L1 12c 8] |
| RR-Mode: | POC10 [L0 8 **3D-Ref**] [L1 12c **3D-Ref**] |
| AR-Mode: | POC10 [L0 8 0] [L1 12c 8] |

- VVC applies its 2D motion models to all RPs (including the 3D-based RPs) and decides which RP should be selected for each CU through rate-distortion optimization.

# Reference Pictures Lists (RPLs) Modifications

- Reference Pictures Lists (RPLs) Modifications
  - Replaced Reference Mode
  - Additional Reference Mode

VTM: POC10 [L0 8 0] [L1 12c 8]

RR-Mode: POC10 [L0 8 **3D-Ref**] [L1 12c **3D-Ref**]

AR-Mode: POC10 [L0 8 0 ] [L1 12c 8 ]

- VVC applies its 2D motion models to all RPs (including the 3D-based RPs) and decides which RP should be selected for each CU through rate-distortion optimization.

# Reference Pictures Lists (RPLs) Modifications

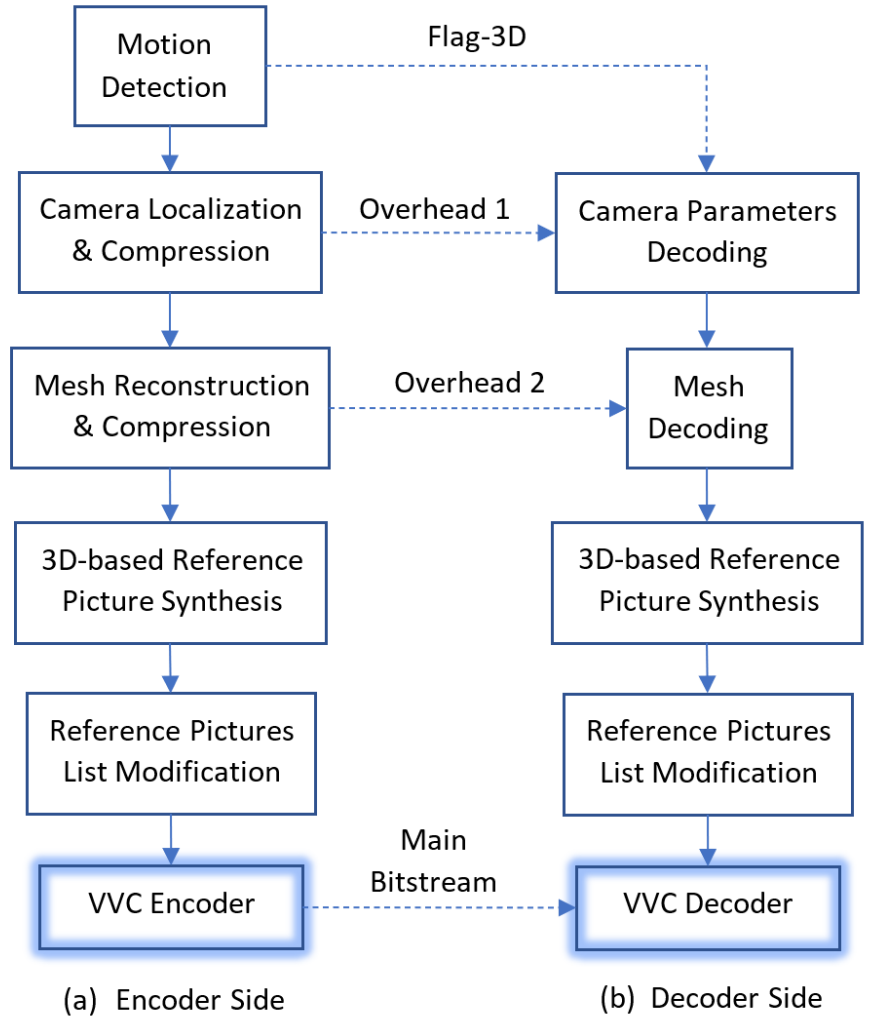
- Reference Pictures Lists (RPLs) Modifications
  - Replaced Reference Mode
  - Additional Reference Mode

| | |
|---|---|
| VTM: | POC10 [L0 8 0] [L1 12c 8] |
| RR-Mode: | POC10 [L0 8 **3D-Ref**] [L1 12c **3D-Ref**] |
| AR-Mode: | POC10 [L0 8 0 **3D-Ref**] [L1 12c 8 **3D-Ref**] |

- VVC applies its 2D motion models to all RPs (including the 3D-based RPs) and decides which RP should be selected for each CU through rate-distortion optimization.

# The proposed method

- Main steps
  - Motion detection (Flag-3D)
  - Camera Localization and Compression (Overhead 1)
  - 3D Mesh Reconstruction and Compression (Overhead 2)
  - 3D-based Reference Picture Synthesis
  - VVC Reference Picture Lists Modifications
  - VVC Encoding

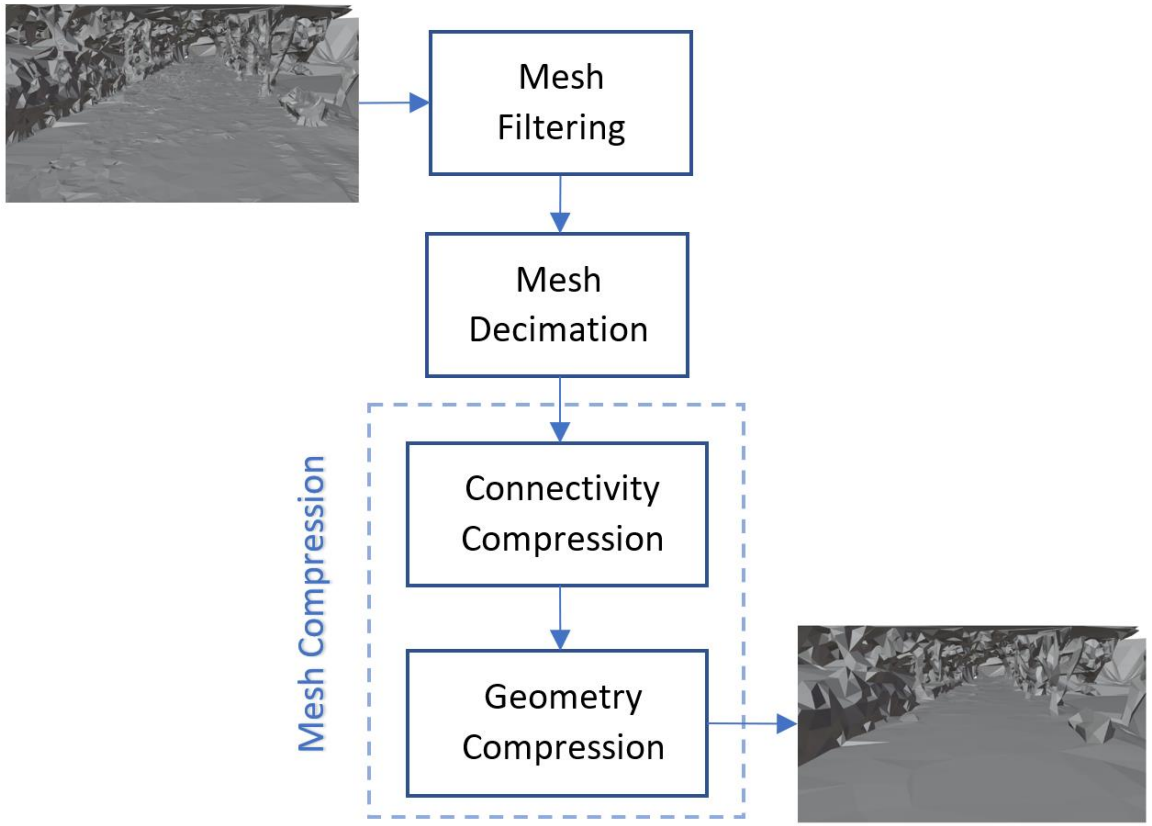


# Outline

- Motivation
- The proposed method
- Simulation results & Analysis
- Conclusion

# Important Mesh Processing Parameters

- Mesh Filtering
  - Bounding Box Size (BB)
- Mesh Decimation
  - Decimation Factor (DF)
- Mesh Compression
  - Geometry Quantization Parameter (G-QP)

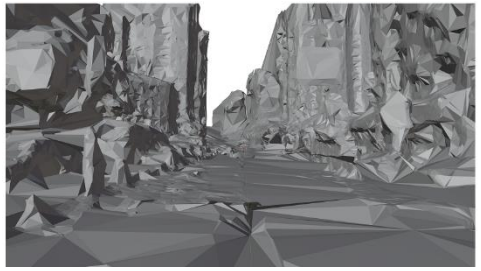


# The Impact of Decimation Factor (DF)

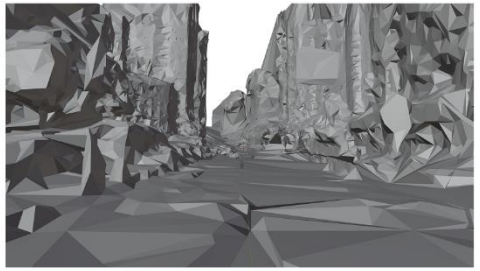
- Decimation Factor (DF)

$$DF = \frac{Number\ of\ Remaining\ \ Vertices\ after\ Decimation}{Number\ of\ Original\ Vertices}\ \%$$

(a) DayLightRoad



(b) Original Mesh
size=697 kB



(c) Decimated and Compressed
DF=0.5, size=26 kB

(d) Decimated and Compressed
DF=0.1, size=6.9 kB

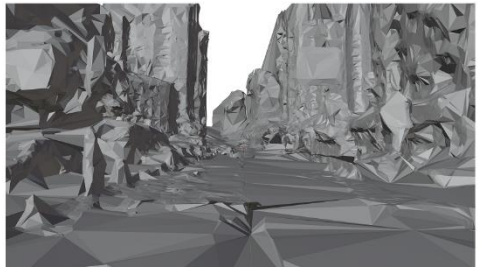
(e) Decimated and Compressed
DF=0.05, size=3.7 kB

# The Impact of Decimation Factor (DF)

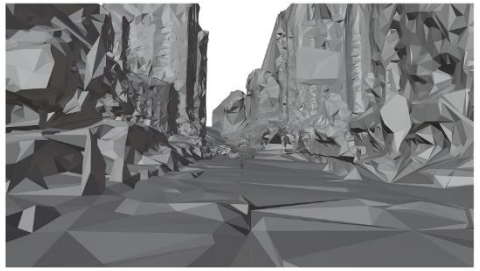
- Decimation Factor (DF)

$$DF = \frac{Number\ of\ Remaining\ \ Vertices\ after\ Decimation}{Number\ of\ Original\ Vertices}\ \%$$

(a) DayLightRoad



(b) Original Mesh size=697 kB



(c) Decimated and Compressed DF=0.5, size=26 kB

(d) Decimated and Compressed DF=0.1, size=6.9 kB

(e) Decimated and Compressed DF=0.05, size=3.7 kB

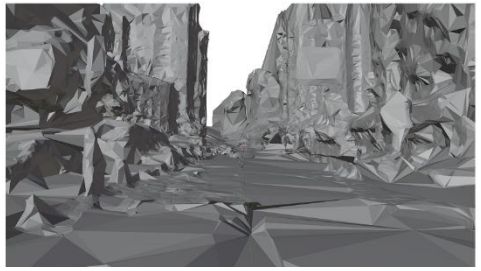
| | | | | |
|---|---|---|---|---|
| BD-Rate without the overhead | -3.04 % | -2. 94 % | -2.61 % | -2.46 % |

# The Impact of Decimation Factor (DF)

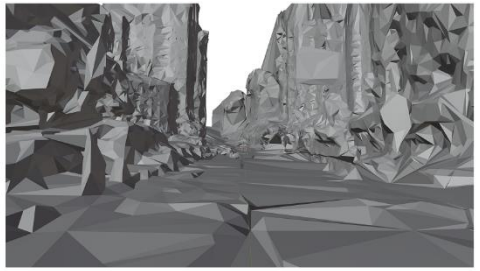
- Decimation Factor (DF)

$$DF = \frac{Number\ of\ Remaining\ \ Vertices\ after\ Decimation}{Number\ of\ Original\ Vertices}\ \%$$

(a) DayLightRoad



(b) Original Mesh size=697 kB



(c) Decimated and Compressed DF=0.5, size=26 kB

(d) Decimated and Compressed DF=0.1, size=6.9 kB

(e) Decimated and Compressed DF=0.05, size=3.7 kB

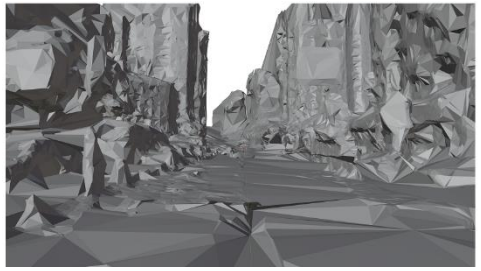
| | | | | |
|---|---|---|---|---|
| BD-Rate without the overhead | -3.04 % | -2. 94 % | -2.61 % | -2.46 % |

# The Impact of Decimation Factor (DF)

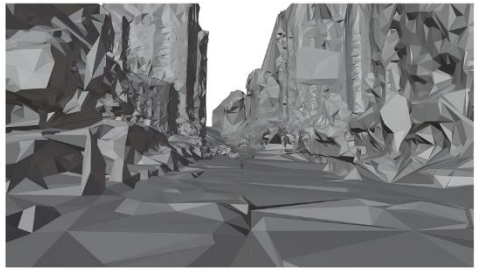
- Decimation Factor (DF)

$$DF = \frac{Number\ of\ Remaining\ \ Vertices\ after\ Decimation}{Number\ of\ Original\ Vertices}\ \%$$

(a) DayLightRoad



(b) Original Mesh size=697 kB



(c) Decimated and Compressed DF=0.5, size=26 kB

(d) Decimated and Compressed DF=0.1, size=6.9 kB

(e) Decimated and Compressed DF=0.05, size=3.7 kB

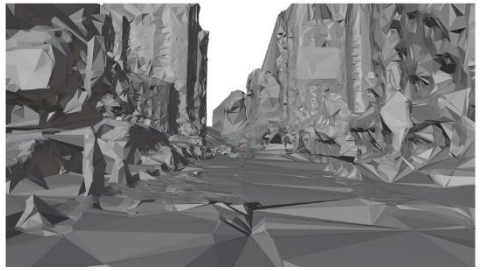
| | | | | |
|---|---|---|---|---|
| BD-Rate without the overhead | -3.04 % | -2. 94 % | -2.61 % | -2.46 % |
| BD-Rate with the overhead | +21.31% | -1.92 % | -2.28 % | -2.24 % |

# The Impact of Decimation Factor (DF)

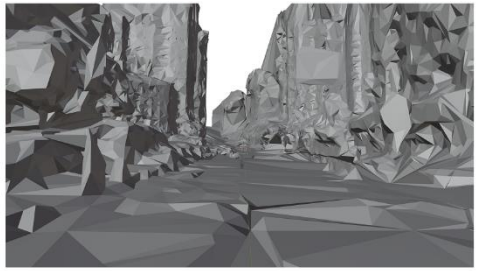
- Decimation Factor (DF)

$$DF = \frac{Number\ of\ Remaining\ \ Vertices\ after\ Decimation}{Number\ of\ Original\ Vertices}\ \%$$

(a) DayLightRoad



(b) Original Mesh size=697 kB



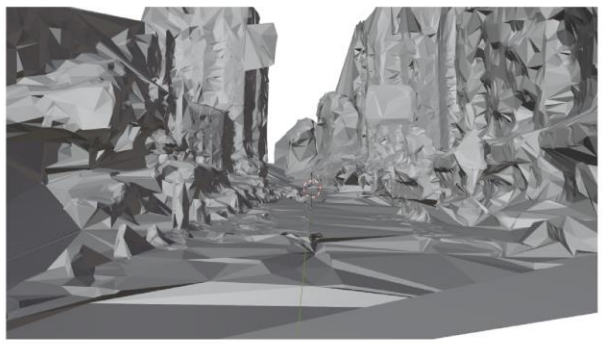
(c) Decimated and Compressed DF=0.5, size=26 kB

(d) Decimated and Compressed DF=0.1, size=6.9 kB

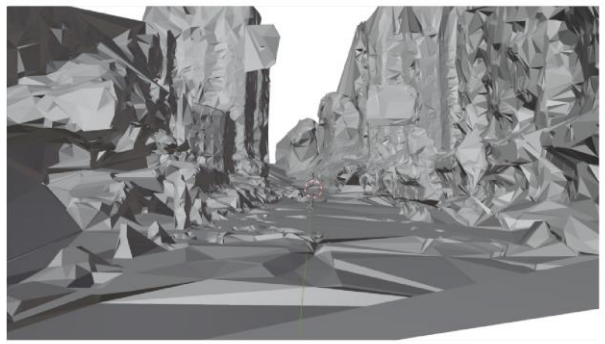
(e) Decimated and Compressed DF=0.05, size=3.7 kB

| | | | | |
|---|---|---|---|---|
| BD-Rate without the overhead | -3.04 % | -2. 94 % | -2.61 % | -2.46 % |
| BD-Rate with the overhead | +21.31% | -1.92 % | -2.28 % | -2.24 % |

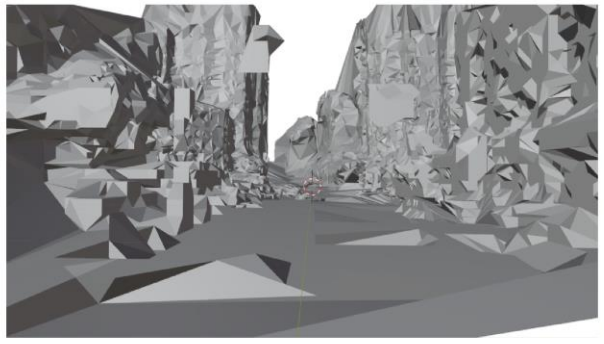
# The Impact of Geometry Quantization Parameter (G-QP)



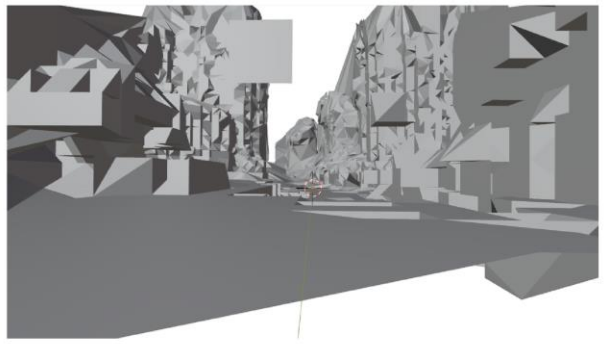
(a) G-QP=14, Size=53kB



(b) G-QP=12, Size=39kB



(c) G-QP=10, Size=27kB



(d) G-QP=8, Size=17kB

- R1: BD-Rate without the overhead
- R2: BD-Rate with the overhead
- Mesh Distortion: d-metric PCC tool [5]

| Seq. | Geometry Quantization | DF-0.1 | | | |
|---|---|---|---|---|---|
| | | *Mesh Dist.* | *Overhead* | *R1* | *R2* |
| Avg. of all seq. | 14-bit | 0.0023 | 14.33 | -2.85 | -2.51 |
| | 12-bit | 0.0041 | 11.93 | -2.79 | **-2.53** |
| | 10-bit | 0.3407 | 9.55 | -2.58 | -2.35 |

[5] G-PCC MPEG-I point cloud comparison tool, accessed 01.02.2021 [Online], available: http://mpegx.int-evry.fr/software/MPEG/PCC/mpegpcc-dmetric.

# Simulation results

- Coding results
  - BD-Rate Y
  - Anchor: VTM 10.0
  - Random Access Profile

VTM: POC10 [L0 8 0] [L1 12c 8]

RR-Mode: POC10 [L0 8 **3D-Ref**] [L1 12c **3D-Ref**]

AR-Mode: POC10 [L0 8 0 **3D-Ref**] [L1 12c 8 **3D-Ref**]

TABLE I. BD-RATE (IN%) AND OVERHEAD (IN KB) FOR DIFFERENT CONFIGURATIONS – ANCHOR: VTM10.0

| Seq. | Configuration | DF-0.5 | | | DF-0.3 | | | DF-0.1 | | | DF-0.05 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | *Overhead* | *R1* | *R2* | *Overhead* | *R1* | *R2* | *Overhead* | *R1* | *R2* | *Overhead* | *R1* | *R2* |
| DayLightRoad | Replaced Ref. | 28.0 | -2.03 | -1.00 | 19.7 | -2.01 | -1.29 | 8.9 | -1.64 | -1.32 | 5.7 | -1.46 | -1.25 |
| | Additional Ref. | | -2.94 | -1.92 | | -2.93 | -2.21 | | -2.61 | **-2.28** | | -2.46 | -2.24 |
| ParkRunning | Replaced Ref. | 42.7 | -1.94 | -1.66 | 30.2 | -1.93 | -1.73 | 14.5 | -1.94 | -1.84 | 9.2 | -1.92 | -1.86 |
| | Additional Ref. | | -2.20 | -1.93 | | -2.21 | -2.01 | | -2.24 | **-2.14** | | -2.20 | **-2.14** |
| MarketPlace | Replaced Ref. | 34.9 | -4.52 | -3.57 | 24.9 | -4.57 | -3.89 | 11.8 | -4.27 | -3.94 | 7.8 | -3.74 | -3.52 |
| | Additional Ref. | | -5.61 | -4.66 | | -5.60 | -4.92 | | -5.34 | **-5.01** | | -4.79 | -4.57 |
| NetFlix-DrivingPOV | Replaced Ref. | 47.9 | -2.91 | -1.61 | 34.1 | -3.03 | -2.11 | 15.9 | -3.01 | -2.58 | 9.8 | -2.82 | -2.55 |
| | Additional Ref. | | -3.66 | -2.37 | | -3.76 | -2.84 | | -3.70 | **-3.27** | | -3.23 | -2.96 |
| MountainBay | Replaced Ref. | 23.5 | -3.37 | -2.57 | 17.2 | -3.37 | -2.79 | 8.7 | -3.42 | -3.12 | 6.0 | -3.40 | **-3.20** |
| | Additional Ref. | | -3.39 | -2.59 | | -3.38 | -2.80 | | -3.41 | -3.11 | | -3.36 | -3.15 |
| NetFlix-Aerial | Replaced Ref. | 34.6 | -2.64 | -2.01 | 24.7 | -2.64 | -2.19 | 11.8 | -2.58 | -2.37 | 7.7 | -2.39 | -2.25 |
| | Additional Ref. | | -2.70 | -2.07 | | -2.71 | -2.26 | | -2.60 | **-2.38** | | -2.39 | -2.25 |

# Encoding/Decoding run-time

- The run-time of all blocks are considered (camera localization, mesh generation, mesh decimation, mesh compression, RP synthesis, ….)

TABLE II. RUNNING-TIME COMPARISON – REFERENCE: VTM 10.0

| | VTM 10.0 | Mesh-based + RR | Mesh-based + AR |
|---|---|---|---|
| Encoding | 100% | 95% | 103% |
| Decoding | 100% | 2668% | 2674% |

- Encoding time is almost equal to or less than the anchor
- Some ideas on how to reduce the decoding time …

# Outline

- Motivation
- The proposed method
- Simulation results & Analysis
- Conclusion

# Conclusion

- A better 3D understanding of the scene results in a better motion compensation
- 3D data can be extracted from 2D video sequences captured by moving cameras
- The proposed method achieves ~*3*% BD-Rate over VTM 10.0 (on average)

- The encoding/decoding run-time
  - The encoding run-time is comparable with the anchor
  - The decoding time is almost 27x

- Sensors like LiDAR, IMU, and ToF can accelerate and enhance the results.

Source: Apple

Source: Samsung


Source:Volkswagen

# Any Questions?

- H. B. Golestani and J.-R. Ohm, “Exploiting the 3D Structures Observed in 2D Video Sequences for Motion Compensation,” in *Picture Coding Symposium (PCS'21)*, (Bristol, UK), IEEE, June 2021.

- Any further questions? Please contact me.

- http://www.ient.rwth-aachen.de
- golestani@ient.rwth-aachen.de