MPEG Video-Based Dynamic Mesh Coding (V-DMC)

Marius Preda, Lukasz Kondrad, Danillo B Graziosi, Wenjie Zou

ICIP Tutorial, September 2025

Presenters

Danillo B Graziosi

Lukasz Kondrad

Wenjie Zou

Marius Preda

Agenda

- **Motivations for V-DMC and overall architecture** (Marius Preda – 15 minutes)
- **Core technologies and coding features** (Danillo Graziosi – 1 hour)
- **Syntax and semantics** (Lukasz Kondrad – 30 minutes)
- **Reconstruction techniques** (Lukasz Kondrad + Danillo Graziosi – 30 minutes)
- **Performance metrics and evaluation** (Wenjie Zou – 25 minutes)
- **Conclusions and future directions** (Marius Preda – 10 minutes)

Agenda

- **Motivations for V-DMC and overall architecture** (Marius Preda – 15 minutes)
- **Core technologies and coding features** (Danillo Graziosi – 1 hour)
- **Syntax and semantics** (Lukasz Kondrad – 30 minutes)
- **Reconstruction techniques** (Lukasz Kondrad + Danillo Graziosi – 30 minutes)
- **Performance metrics and evaluation** (Wenjie Zou – 25 minutes)
- **Conclusions and future directions** (Marius Preda – 10 minutes)

3D Graphics Compression

- Hard problem
  - Data heterogeneity
    - Scene
    - Geometry
    - Attributes
  - Different Representation Methods
  - Different Application Requirements

3DG Compression, a N dimensional problem

Image and Video coding is “easy”, only colors

- A good compression method is agnostic to the ~~signal~~ but exploits the nature of the signal

Dynamic 3D Graphics Compression

- Harder problem
  - Data heterogeneity
    - Scene
    - Time-varying Geometry
    - Time-varying Attributes
    - Animation
  - Different Representation Methods
  - Different Application Requirements

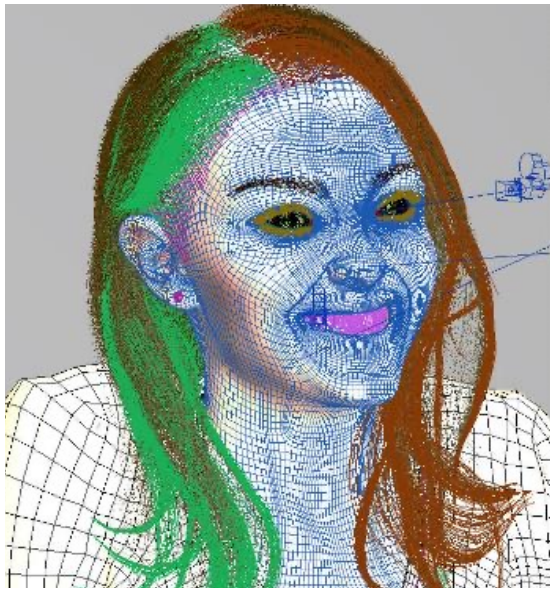
Dynamic 3DG Compression, a N+1 dimensional problem

- A good compression method is agnostic to the signal but exploits the nature of the signal

A mesh object

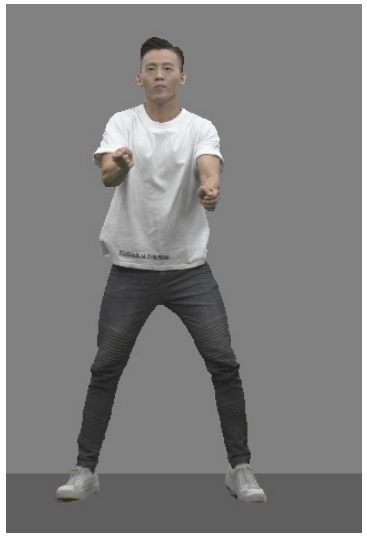
- A set of 3D points
  - ordered,
  - connected to form polygons

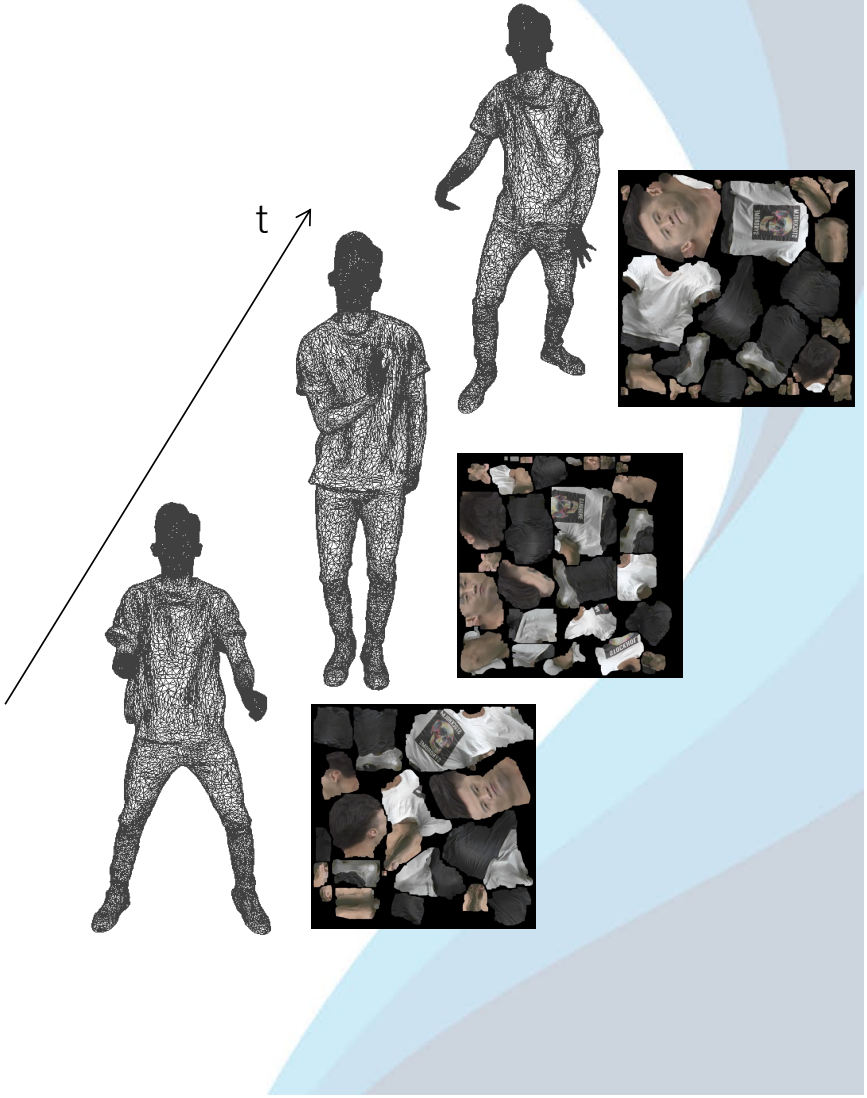
- An animated mesh is defined by
  - $(X_t, Y_t, Z_t)_n$
  - $(v_1, v_2, v_3)_m$
  - (R, G, B) – still image
  - a mapping from texture to geometry
  - reflectance, transparency, …



A dynamic mesh object

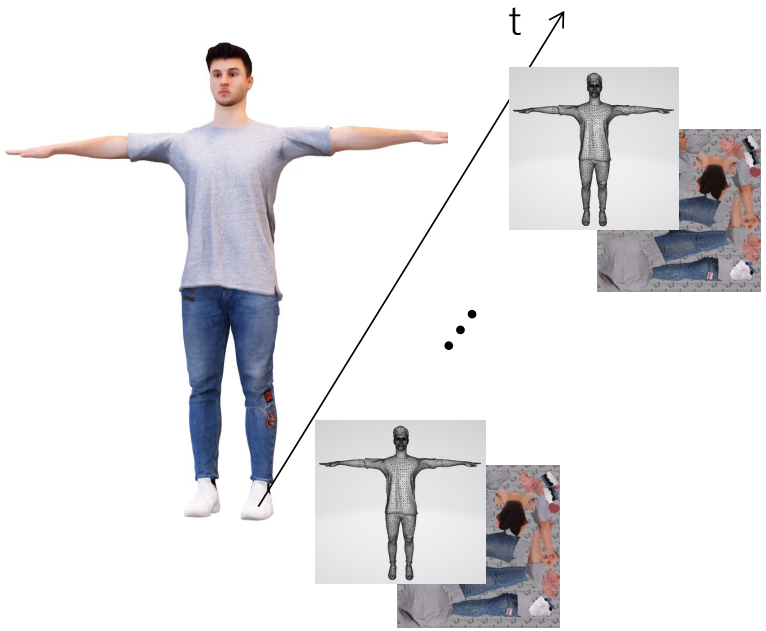
- A set of 3D points
  - ordered,
  - connected to form polygons varying in time
- A dynamic mesh is defined by
  - $(X_t, Y_t, Z_t)_{n(t)}$
  - $(v_1, v_2, v_3)_{m(t)}$
  - (R, G, B) – video
  - a dynamic mapping from texture to geometry
  - dynamic reflectance, transparency, …





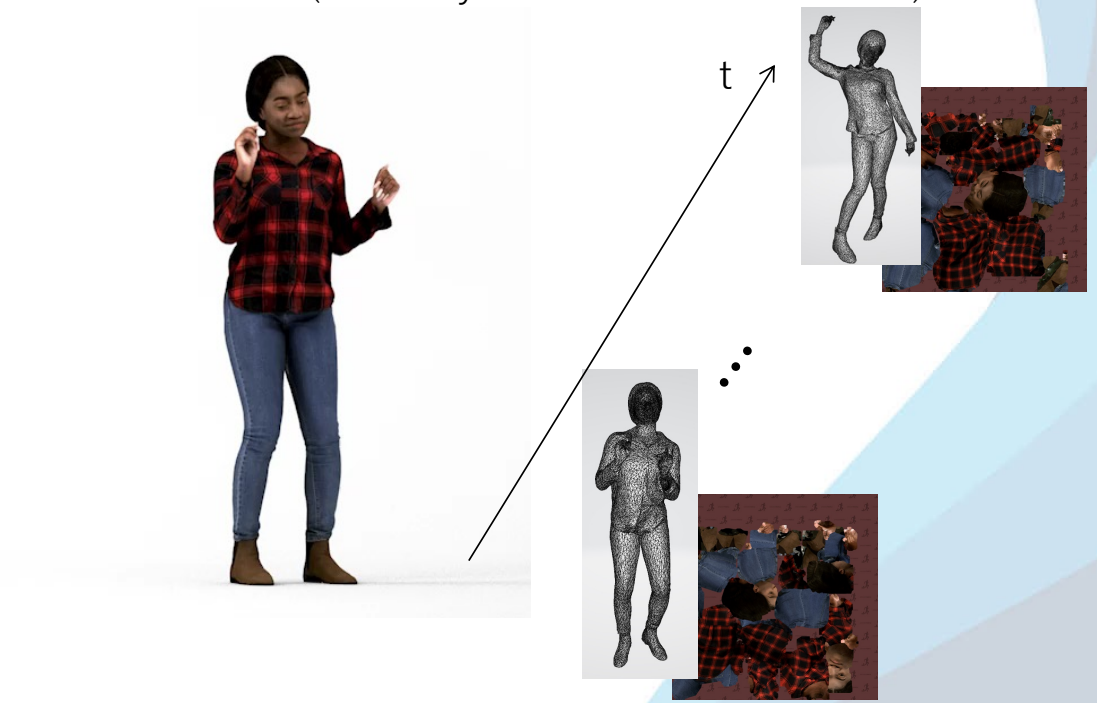
Dynamic meshes vs. animated meshes

Animated Mesh
(Frame by Frame Vertex reconstruction)



~181 MB with animation quality

Dynamic Mesh
(Frame by Frame Mesh construction)



~3.46 GB with realistic quality

Free model from
renderpeople.com

Dynamic meshes – a type of volumetric media

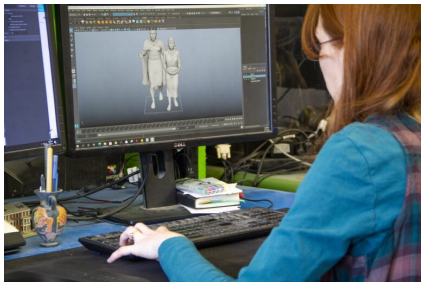
"Volumetric media" refers to a type of digital media that captures a three-dimensional space or object in such a way that it can be interacted with or viewed from any angle.

Capture and Creation	Created by capturing real-world environments using multiple cameras from different angles or/and through computer graphics to generate 3D digital models.
Interactivity	Allows viewers to manipulate and view the media from any angle within the 3D space, enhancing the sense of immersion and interactivity.
Technologies Involved	Involves 3D modeling, computer vision, computational photography, and sometimes artificial intelligence to enhance model accuracy and detail.
Applications	Used in VR and AR for entertainment, immersive learning environments in education, and telepresence in professional settings for interactive 3D holographic communications.
Advantages	Provides realistic representations of environments, enhancing user engagement and emotional connection; beneficial for experiential learning and storytelling.
Challenges	Requires high data bandwidth for storage and transmission, significant computational resources for processing, and specialized hardware for creation and viewing.

Why volumetric media?

Traditional video technologies - since 1930’

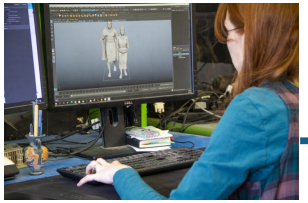
The best of the two worlds, however, still in an embryonic phase

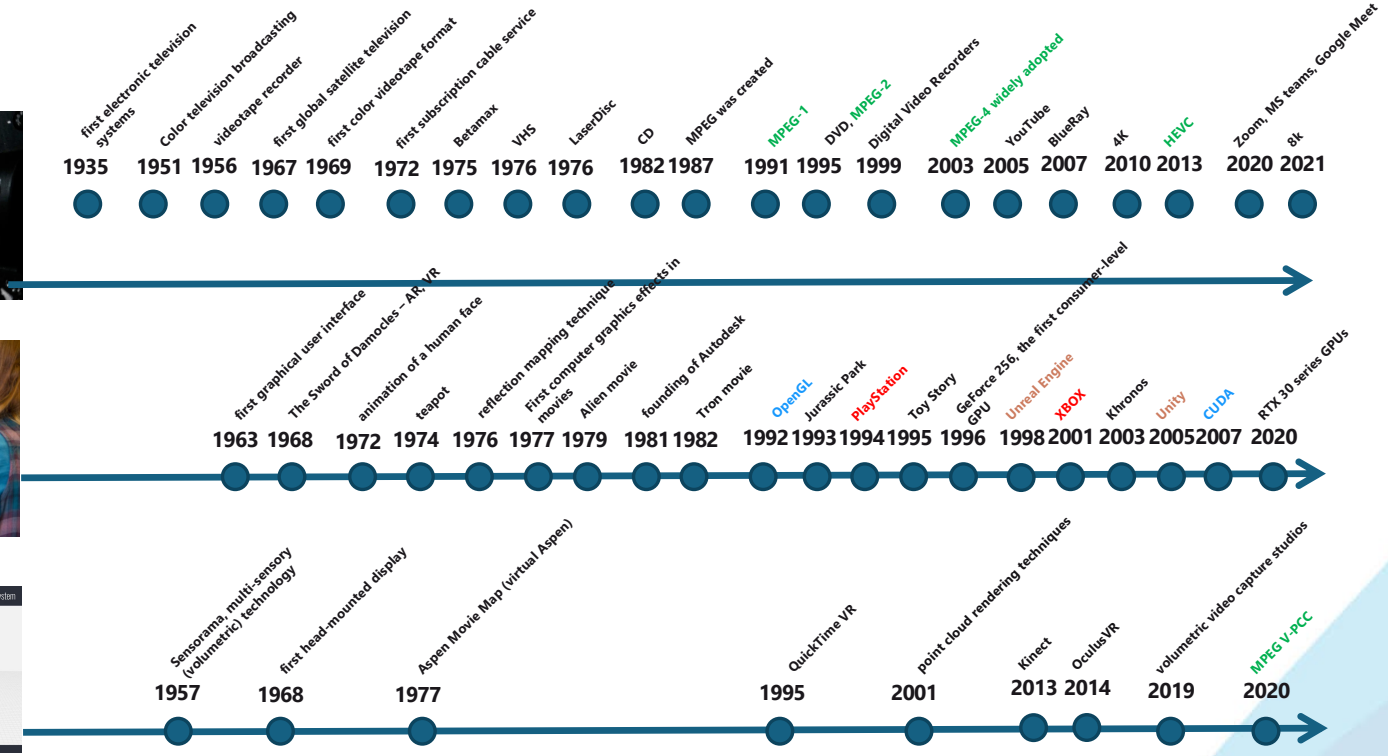


Traditional 3D graphics technologies - since 1960’

Why volumetric media?

3 timelines





The video standardization was successful, but the 3D graphics situation could have been better in terms of interoperability

Synergies by the confluence of two worlds of technologies

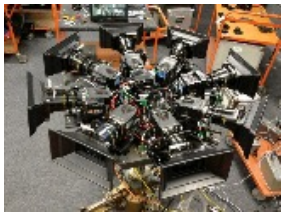
Visual capture

Visual synthesis

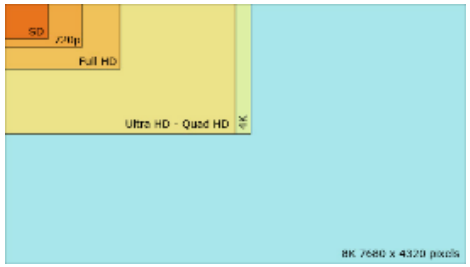
Synergies by the confluence of two worlds of technologies

Visual capture

LDR, HDR



Multi-camera



HD, Full HD, 4K, 8K

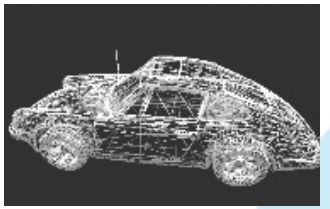
Stereoscopy

Visual synthesis

Synergies by the confluence of two worlds of technologies

Visual capture

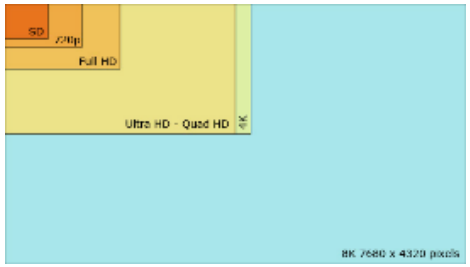
Geometric primitives



Visual synthesis

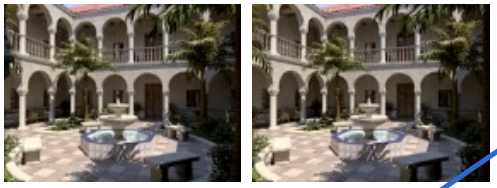
Synergies by the confluence of two worlds of technologies

Visual capture

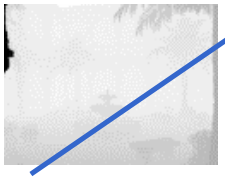


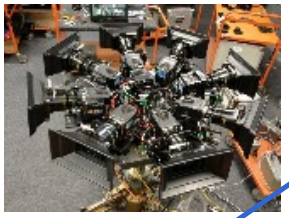
HD, Full HD, 4K, 8K

LDR, HDR



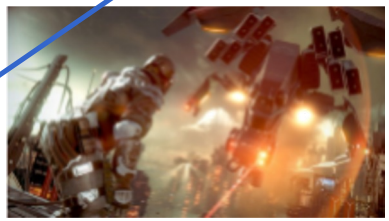
Stereoscopy

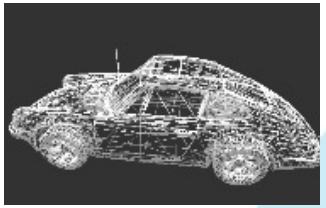




Multi-camera

Geometric primitives

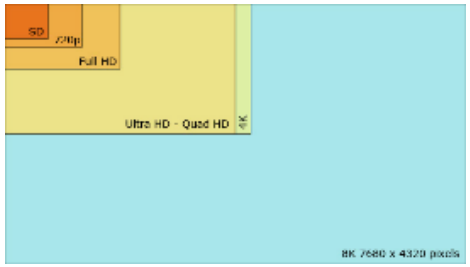


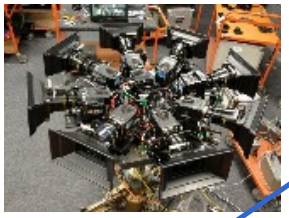


Visual synthesis

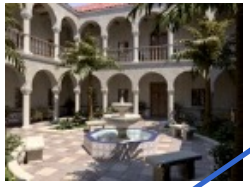
Synergies by the confluence of two worlds of technologies

Visual capture

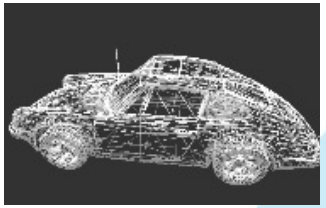


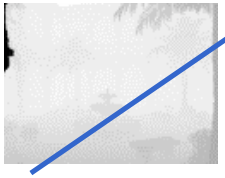


Easy to produce
High quality



Interactivity
Immersion

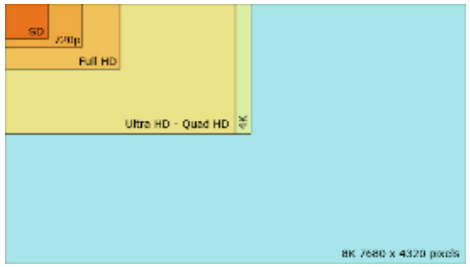


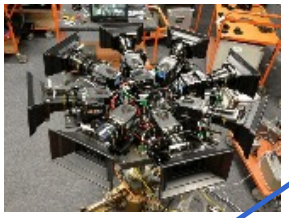


Visual synthesis

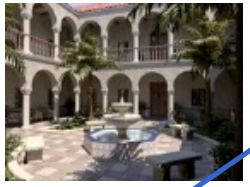
Synergies by the confluence of two worlds of technologies

Visual capture

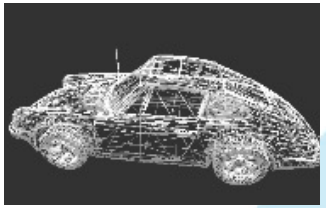


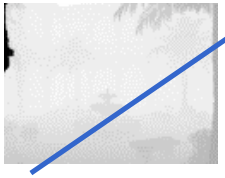


Easy to produce
High quality



Interactivity
Immersion





Visual synthesis

What do we need?

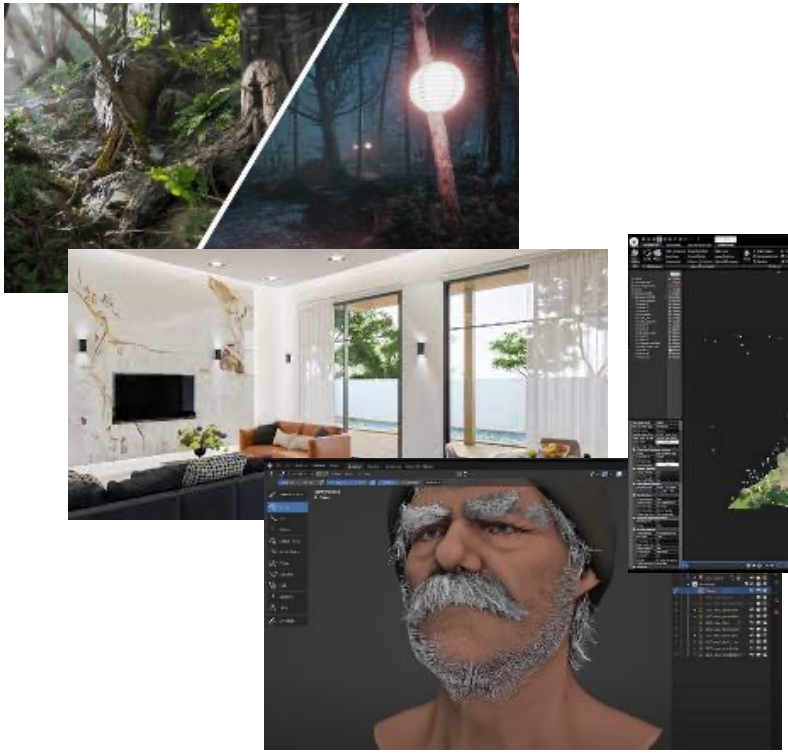
Visual capture

Easy to produce, high quality, realistic, interactive
CONTENT

Visual synthesis

What do we need?

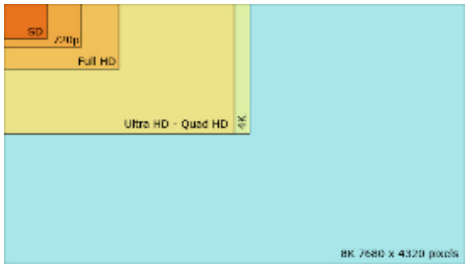
Easy to produce, high quality, realistic, interactive
CONTENT



This is cool! … but heavy

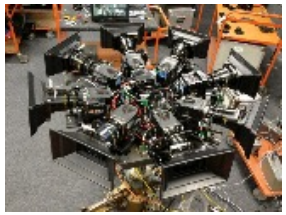
COMPRESSION is the keystone

Capturing more and more from the reality



HD, Full HD, 4K, 8K

LDR, HDR



Multi-camera

Stereoscopy

What is the most appropriate representation format?

- Video + depth
- Many videos + depths

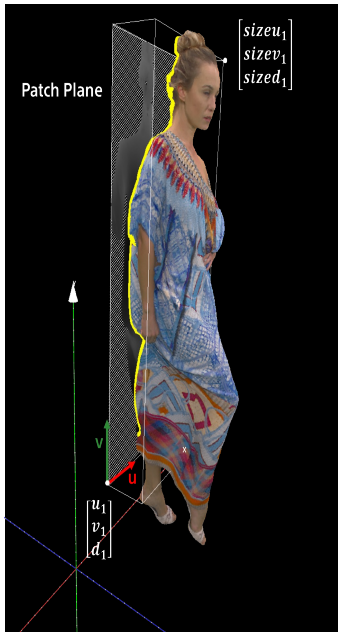
MPEG Video

- Point clouds
- Meshes
- Gaussian Splatting

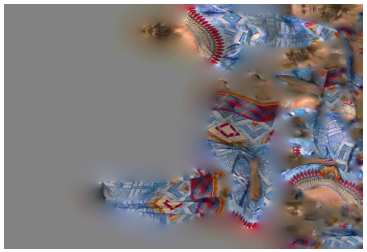
MPEG Graphics

Looking for an integrated framework

Encoding the 3D representation as a set of 2D projections



Geometry



Attributes

What is the most appropriate representation format?

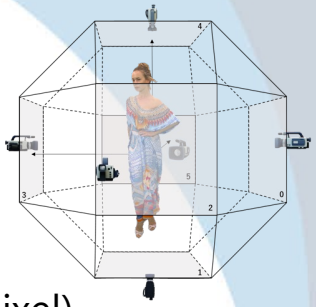
- Video + depth
- Many videos + depths

MPEG Video

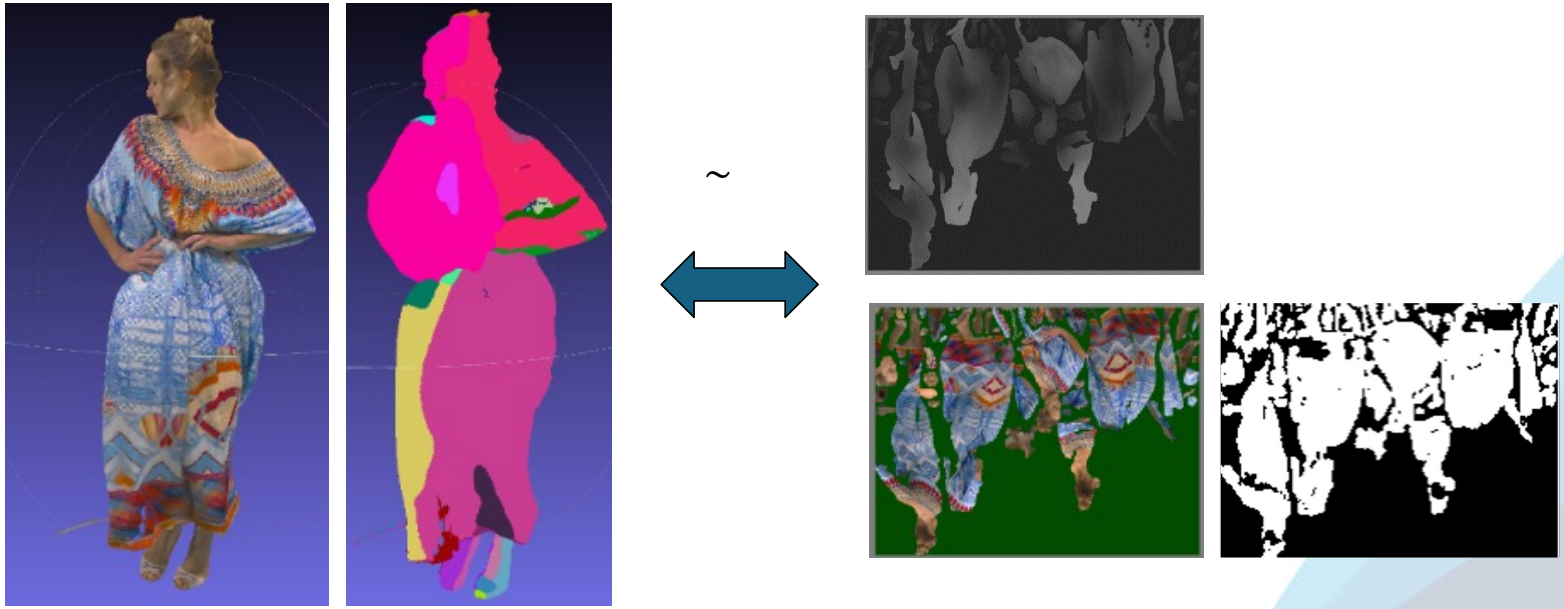
- Point clouds
- Meshes
- Gaussian Splatting

MPEG Graphics

Video-based Point Clouds Compression

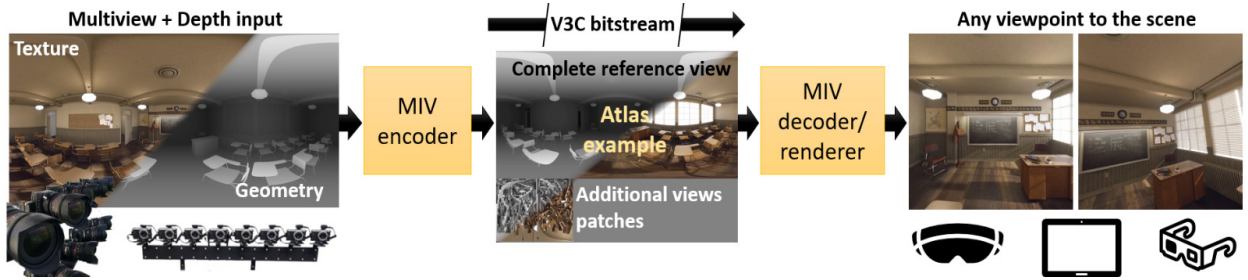


- First successful instantiation of the 3D to 2D projection framework
- Encoding the 3D point clouds as a set of 2D patches (a 3D point is transposed into a 2D pixel)
- Not all the pixels in the image are used for reconstruction, an occupancy map indicates which ones should be used



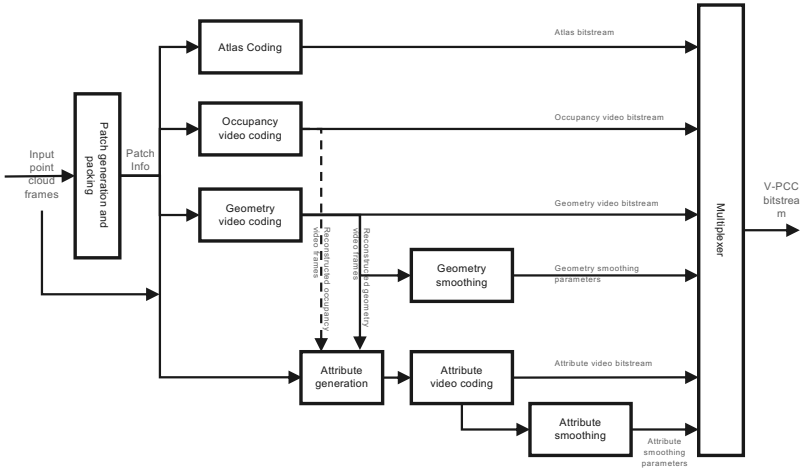
MPEG Immersive video

- A second successful instantiation of the 3D to 2D projection framework
- MIV data representation: reference view + dis-occlusion patches

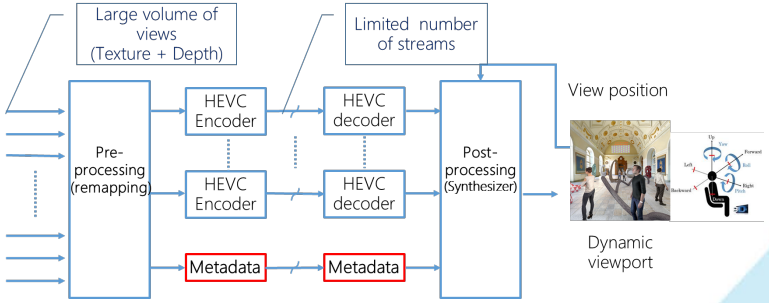


MPEG V3C: Visual Volumetric Video-based Coding

- MPEG V3C : a powerful framework in which a volumetric representation is converted into a set of color + depth videos



V-PCC encoding pipeline



3DoF+ (MIV) encoding pipeline

How V3C principle applies to meshes?

Points are nice but … they should be many ☺ to realistically representing objects and scenes

Even if the compression is solved, it remains the Rendering problem

(graphics cards are optimized for triangles)

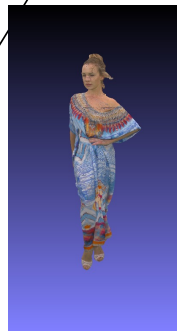
Solution: reconsidering Mesh – based representations within the V3C context

(a surface approximation of the point cloud object)

Dynamic meshes vs. point clouds

Point cloud
(Frame by Frame Point reconstruction)

t

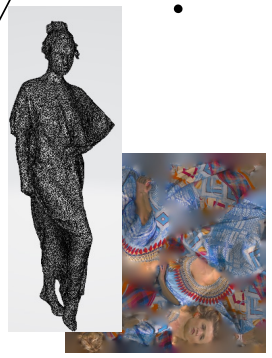


✔ Easy to produce
✖ Difficult to render

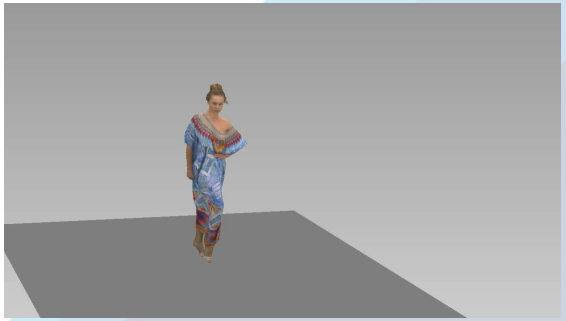
Mesh construction

Dynamic Mesh
(Frame by Frame Mesh construction)

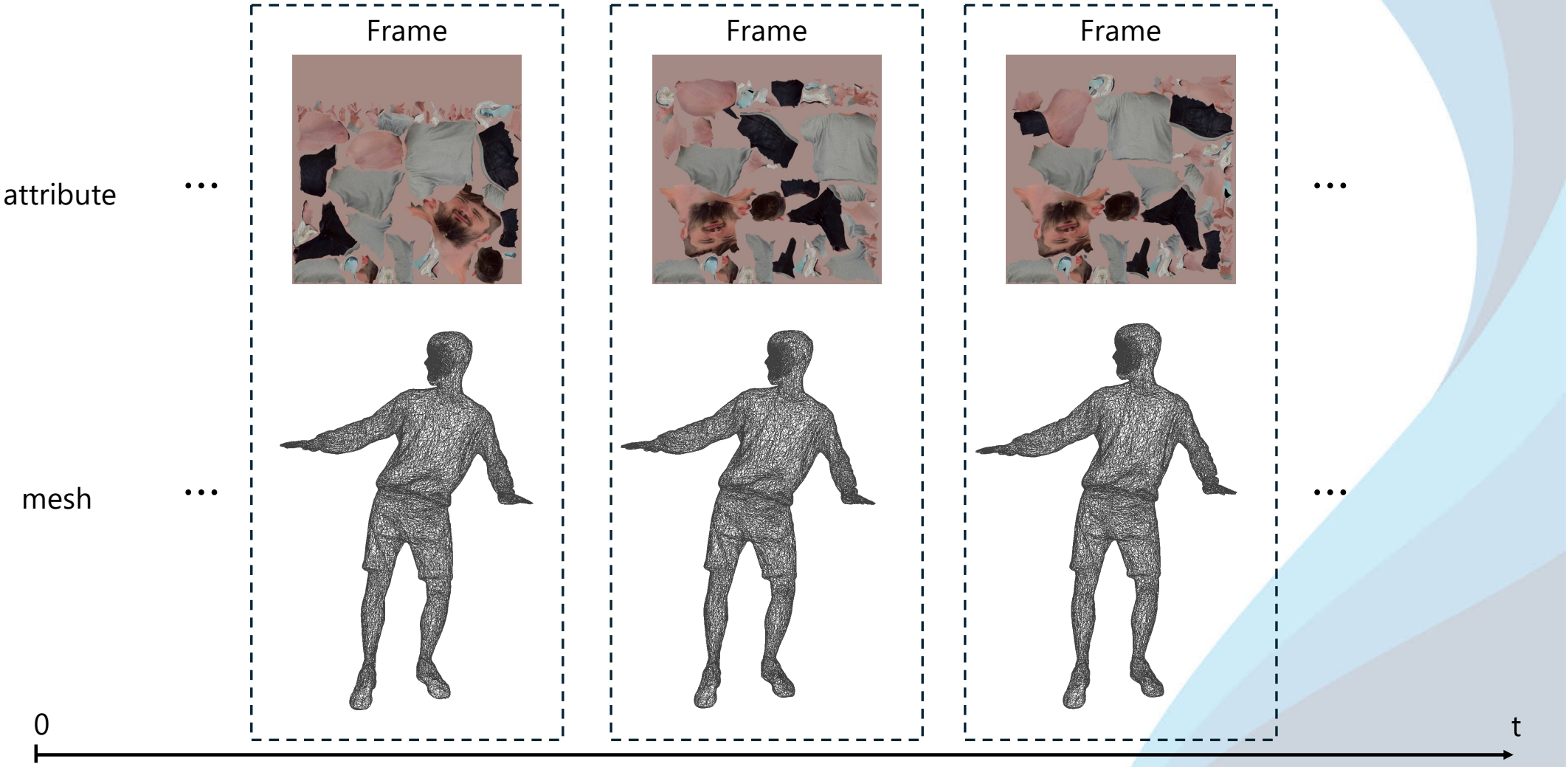
t



– Relatively easy to produce
✔ Easy to render



Mesh sequence



Considerations

- The easiest approach would be to encode mesh with static mesh codec and attribute with video codec, but
  - Static mesh coding may not be optimized for real-time decoding especially for high resolution mesh
  - Transferring data from CPU to GPU for high resolution mesh also may be not desirable

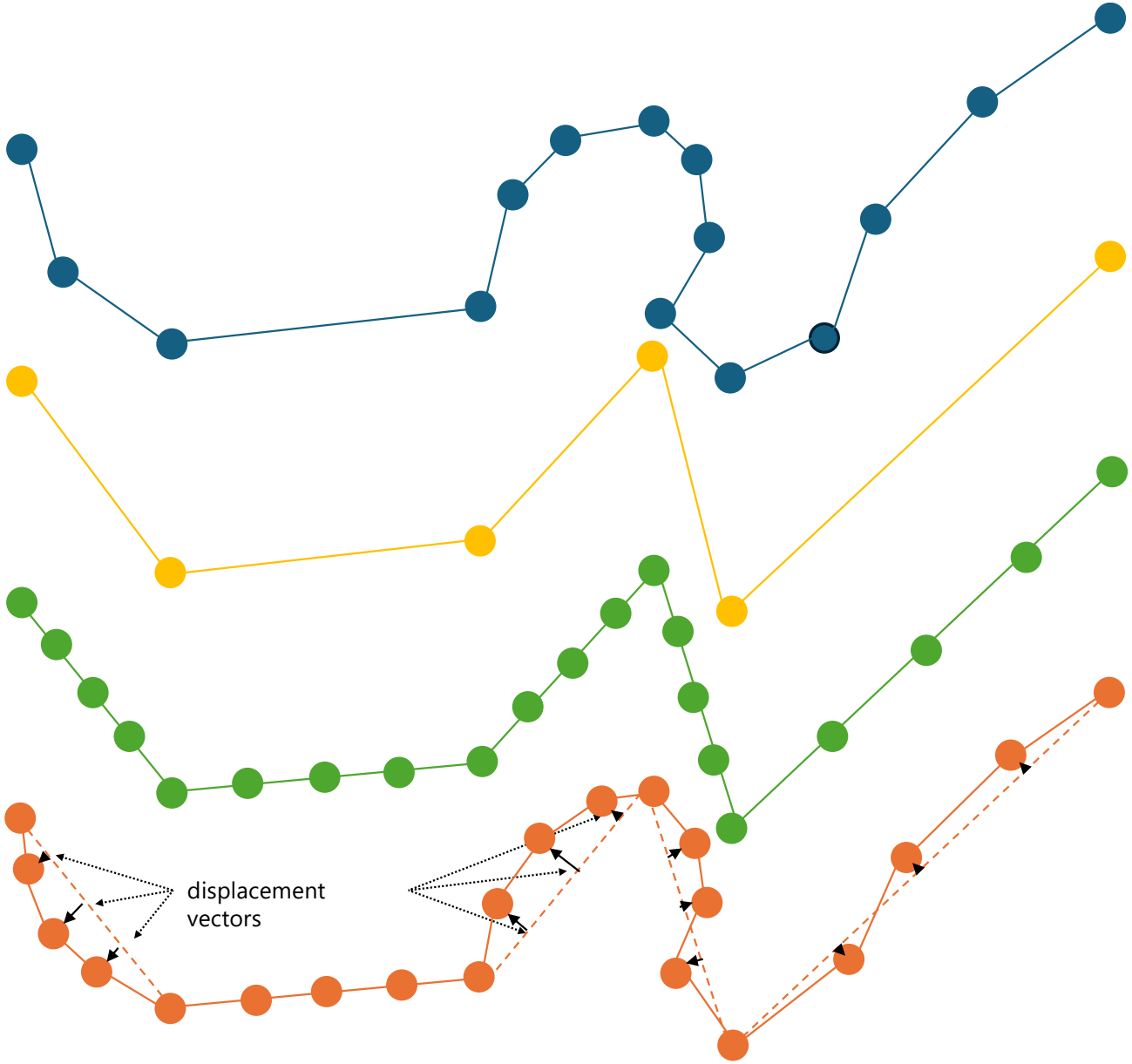
- A better idea: to decimate, subdivide and displace the mesh before encoding

original

decimated

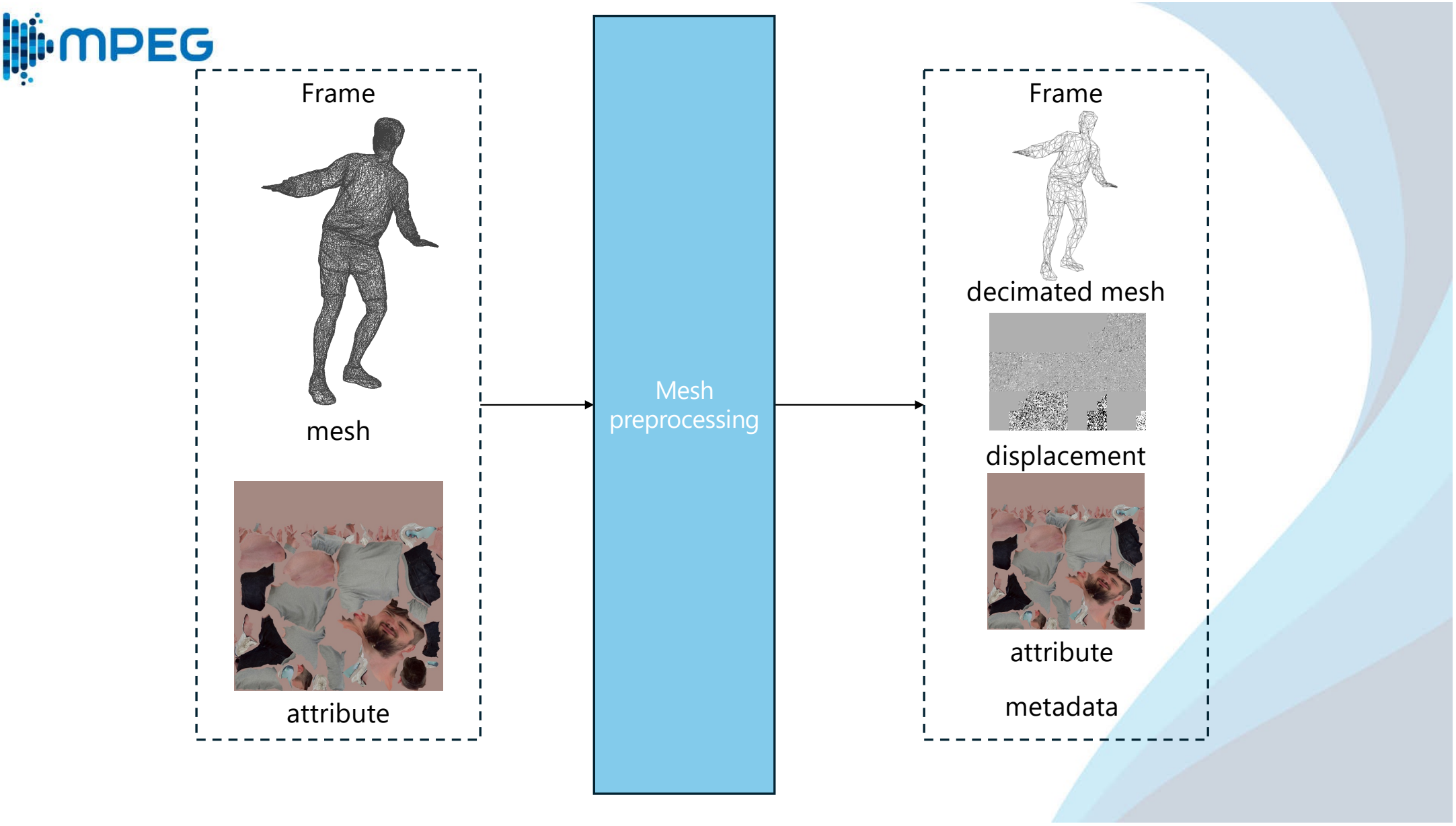
subdivided

displaced



Uniqueness of V-DMC

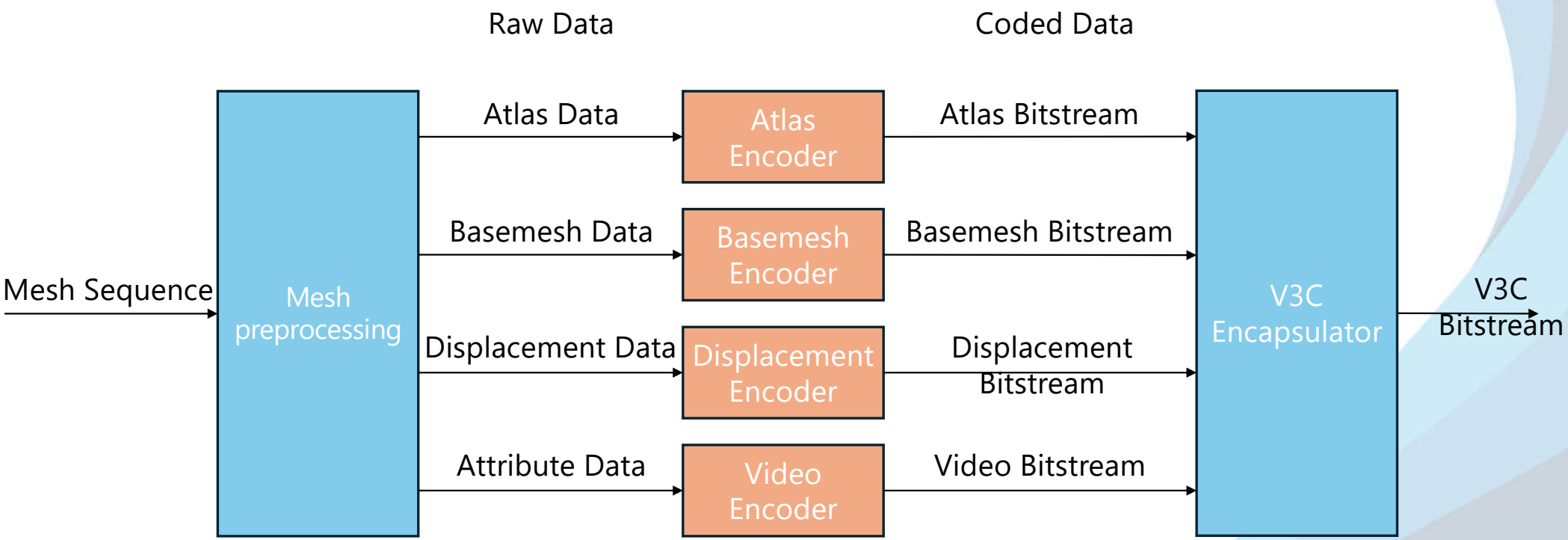
- Connectivity is lossy encoded using a basemesh sub-bitstream, and the geometry video and attribute videos are not aligned anymore
  - geometry is used for displacement of each vertex, being placed in the video frame in a raster scan order,
  - while attributes use the texture coordinates and encode surface properties


Frame
mesh
attribute
Mesh preprocessing
Frame
decimated mesh
displacement
attribute
metadata

Pre-processing tools

- mesh-simplification
- geometry parametrization
  - various subdivision scheme
  - displacement optimization
- texture parametrization
  - attribute transfer
  - temporal consistency
- mesh segmentation

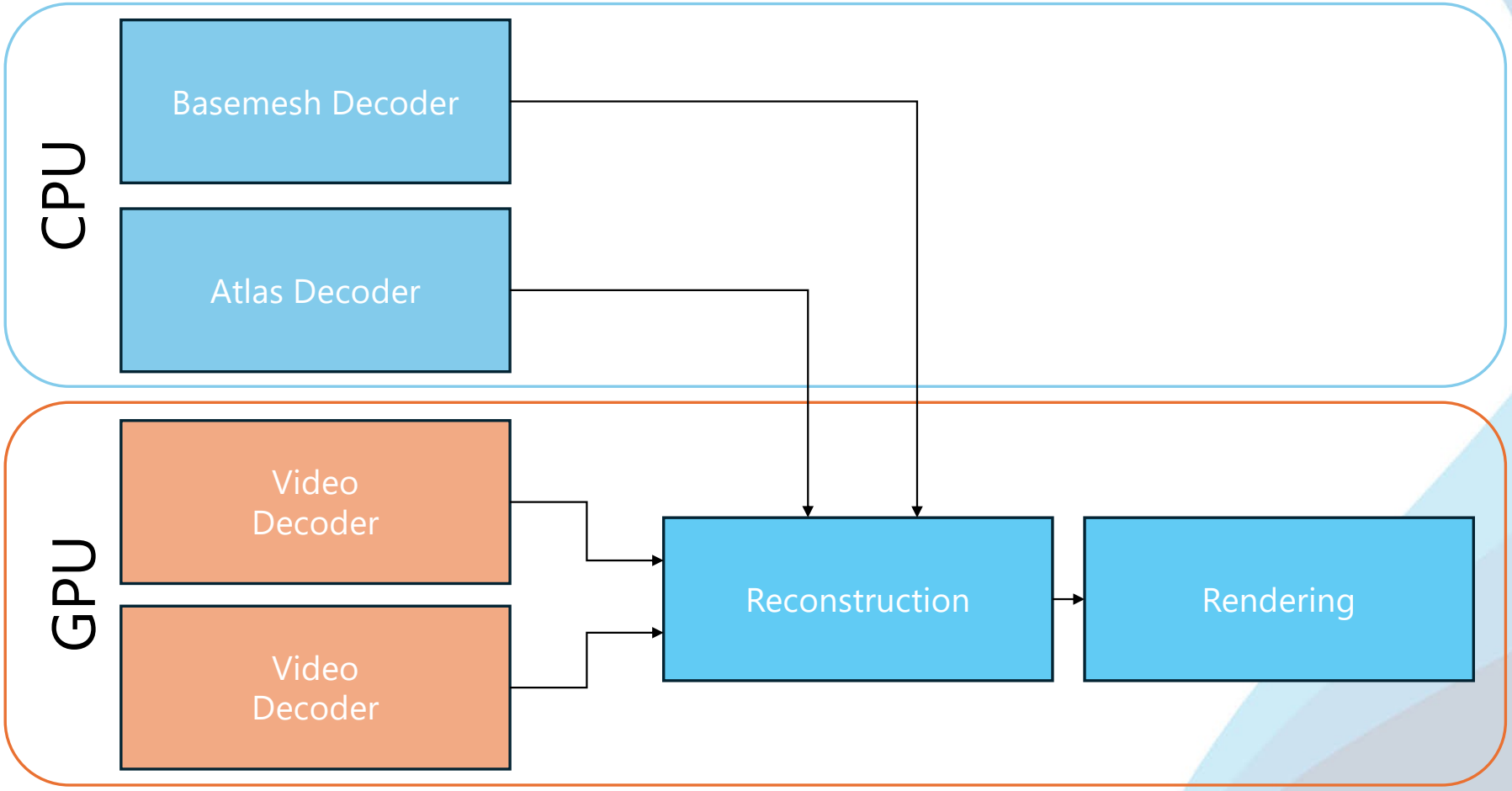
Encoder


Raw Data
Coded Data
Mesh Sequence
Mesh preprocessing
Atlas Data
Basemesh Data
Displacement Data
Attribute Data
Atlas Encoder
Basemesh Encoder
Displacement Encoder
Video Encoder
Atlas Bitstream
Basemesh Bitstream
Displacement Bitstream
Video Bitstream
V3C Encapsulator
V3C Bitstream

Encoding tools

- atlas encoder
  - temporal correspondences
- base mesh encoder
  - quantization
  - mesh coding algorithms
  - temporal correspondences
- displacement encoder
  - video encoder
  - ac displacement encoder
- attribute encoder
  - video encoder

Receiver Pipeline



Goals while deciding on architecture

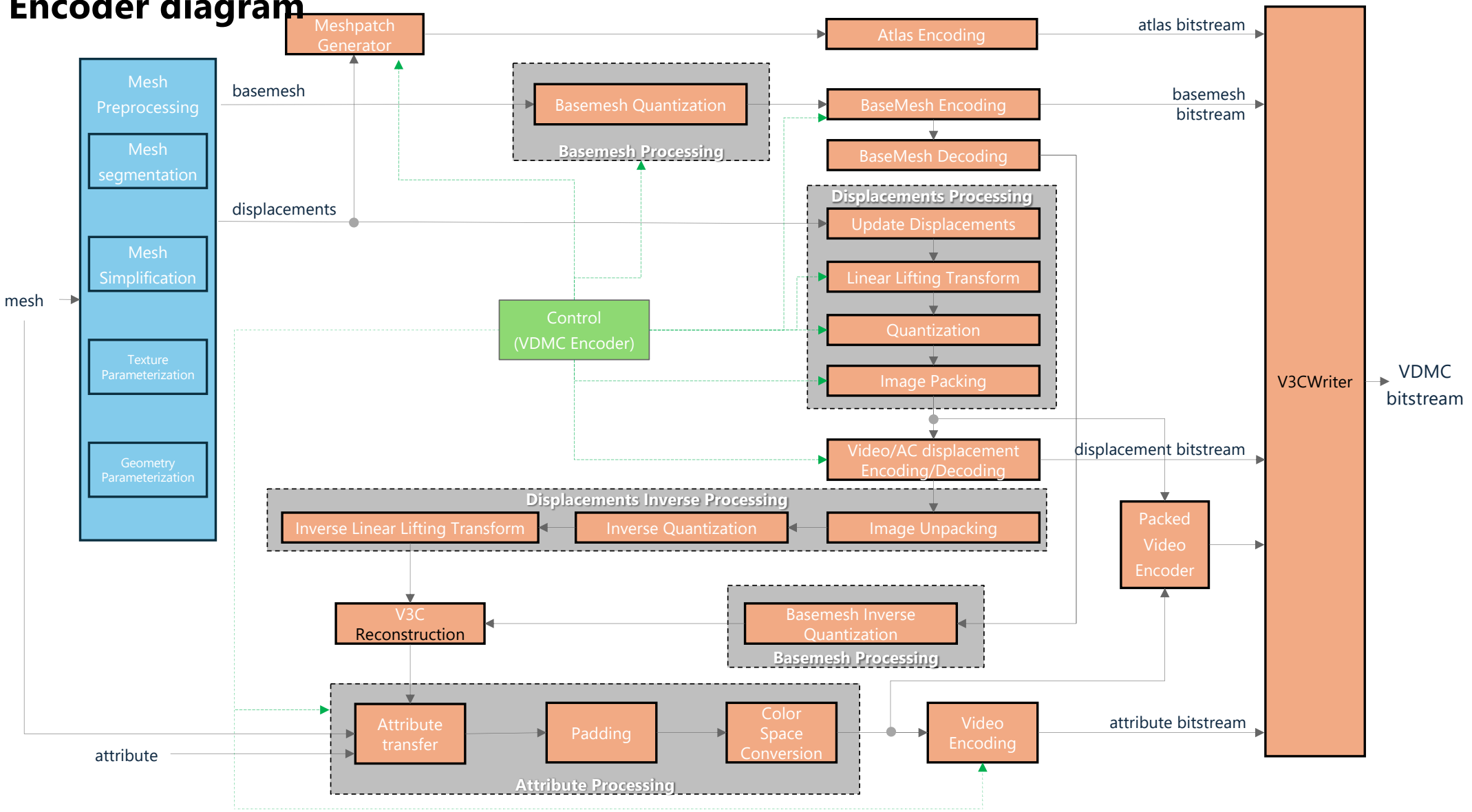
- Short time to market
- Real-time low-power decoding
- Real-time low-power rendering
- Efficient compression

- Extendable solution

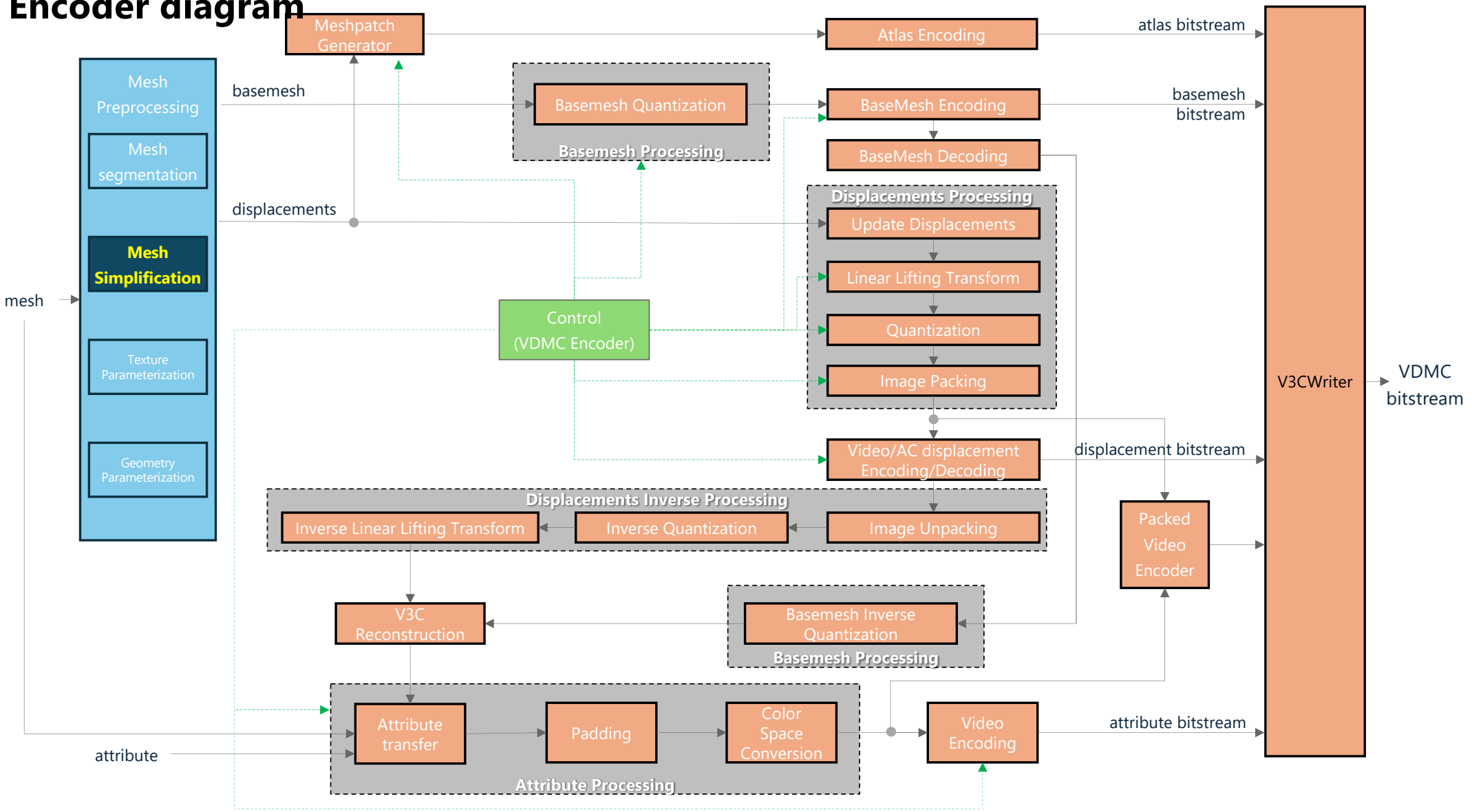
Agenda

- **Motivations for V-DMC and overall architecture** (Marius Preda – 15 minutes)
- **Core technologies and coding features** (Danillo Graziosi – 1 hour)
- **Syntax and semantics** (Lukasz Kondrad – 30 minutes)
- **Reconstruction techniques** (Lukasz Kondrad + Danillo Graziosi – 30 minutes)
- **Performance metrics and evaluation** (Wenjie Zou – 25 minutes)
- **Conclusions and future directions** (Marius Preda – 10 minutes)

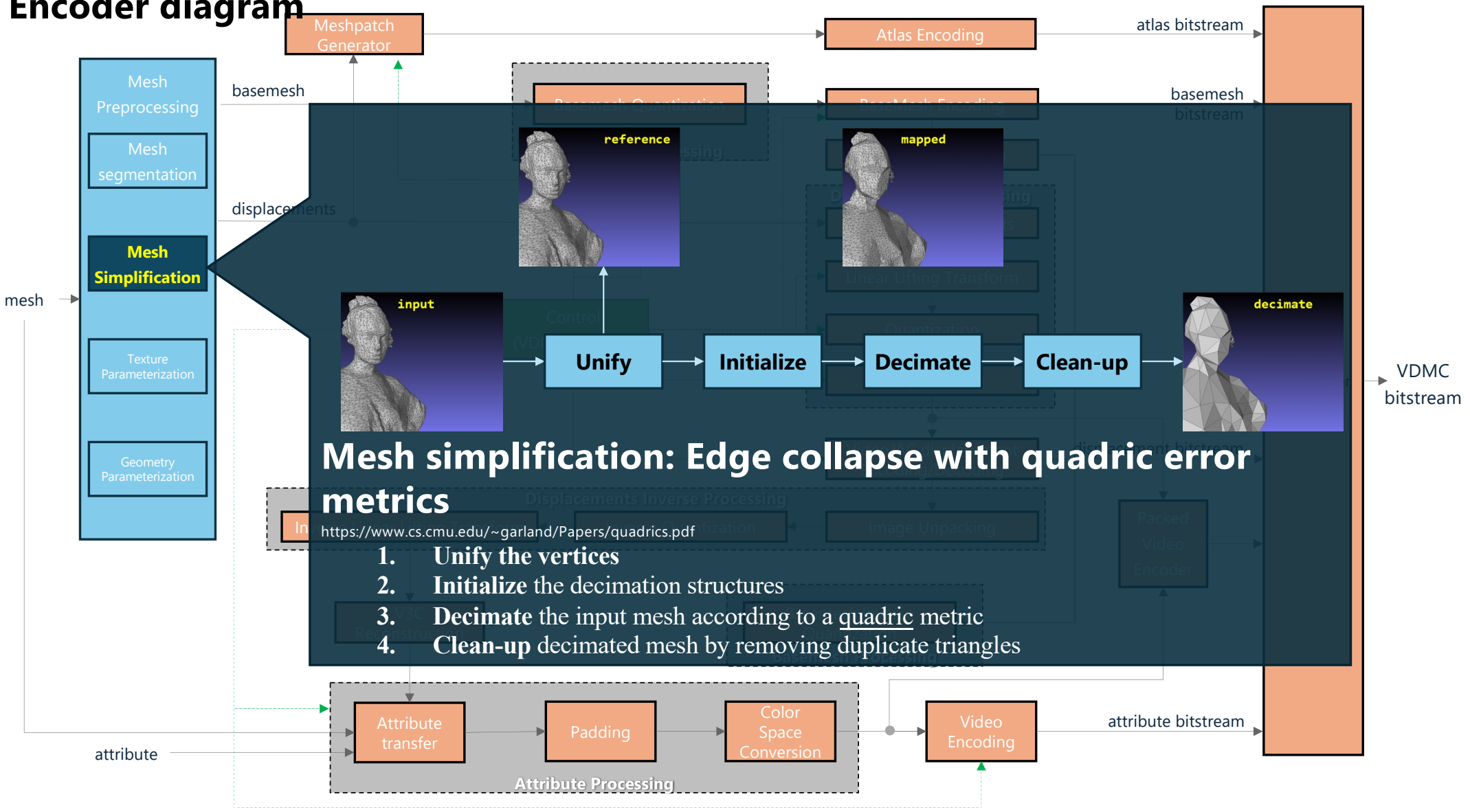
Encoder diagram



Encoder diagram

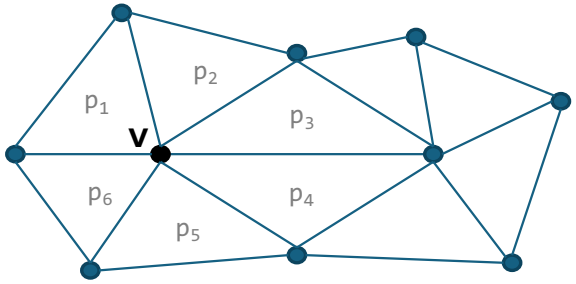


Encoder diagram



Pre-processing: mesh simplification

- Quadric cost: error of the vertex to a set of planes as the sum of squared distances

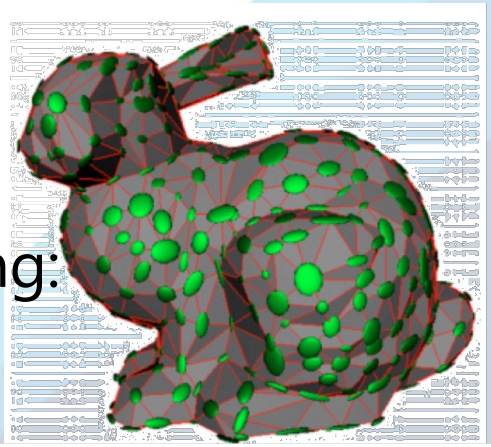


$$\Delta(\mathbf{v}) = \Delta([v_x\ v_y\ v_z\ 1]^T) = \sum_{\mathbf{p}\in\text{planes}(\mathbf{v})} (\mathbf{p}^T\mathbf{v})^2 = \mathbf{v}^T \left(\sum_{\mathbf{p}\in\text{planes}(\mathbf{v})} \mathbf{K_p} \right) \mathbf{v}$$

$\mathbf{p} = [a\ b\ c\ d]^T$ represents the plane defined by the equation $ax + by + cz + d = 0$ where $a^2 + b^2 + c^2 = 1$.

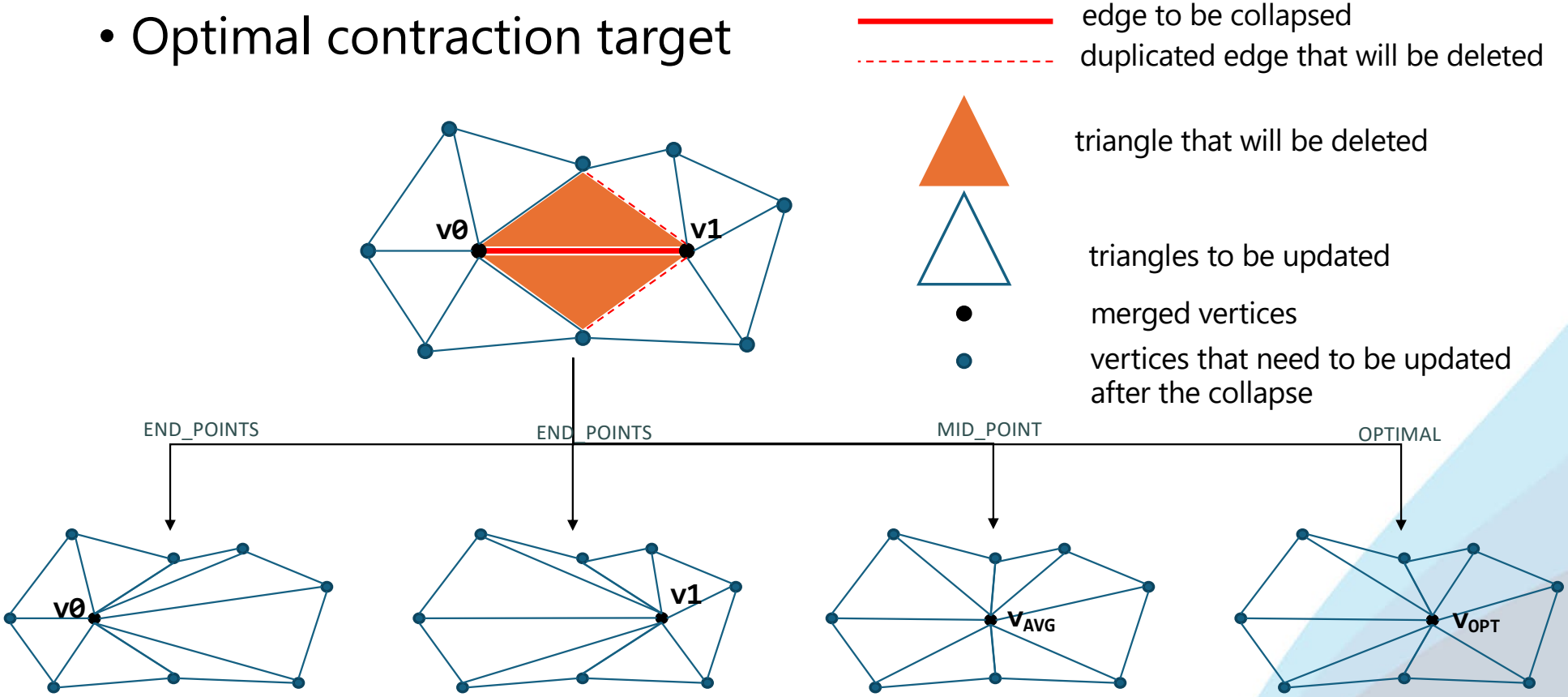
$$\mathbf{K_p} = \mathbf{p}\mathbf{p}^T = \begin{bmatrix} a^2 & ab & ac & ad \\ ab & b^2 & bc & bd \\ ac & bc & c^2 & cd \\ ad & bd & cd & d^2 \end{bmatrix}$$

- Additional modifications to the quadric cost:
  - Triangle area, Boundaries
- Error of contraction target also considers the following:
  - Normal flipping, Elongated triangles



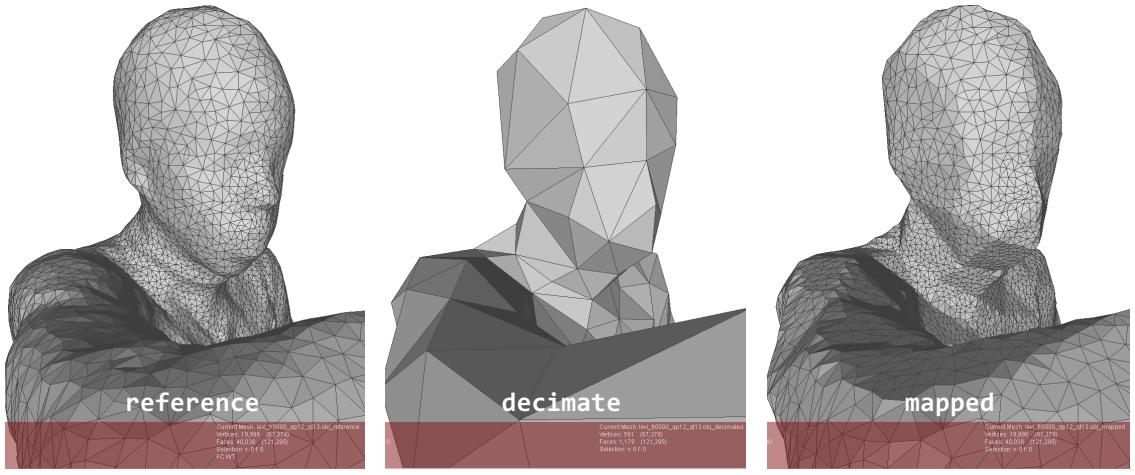
Pre-processing: mesh simplification

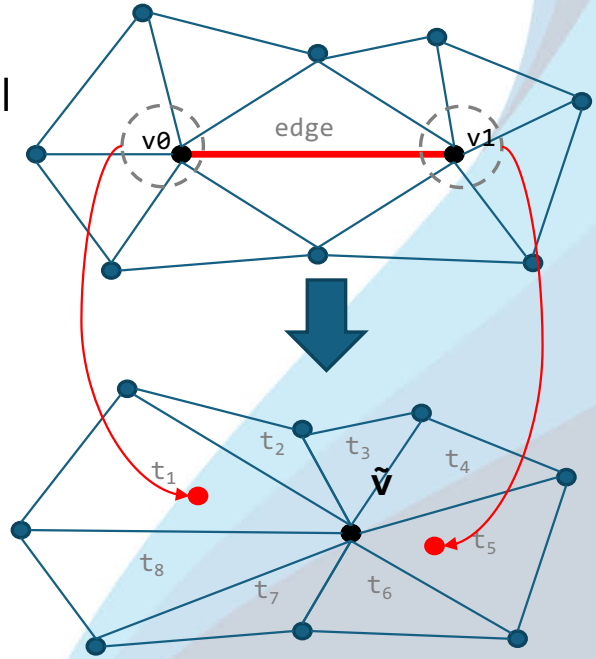
- Optimal contraction target



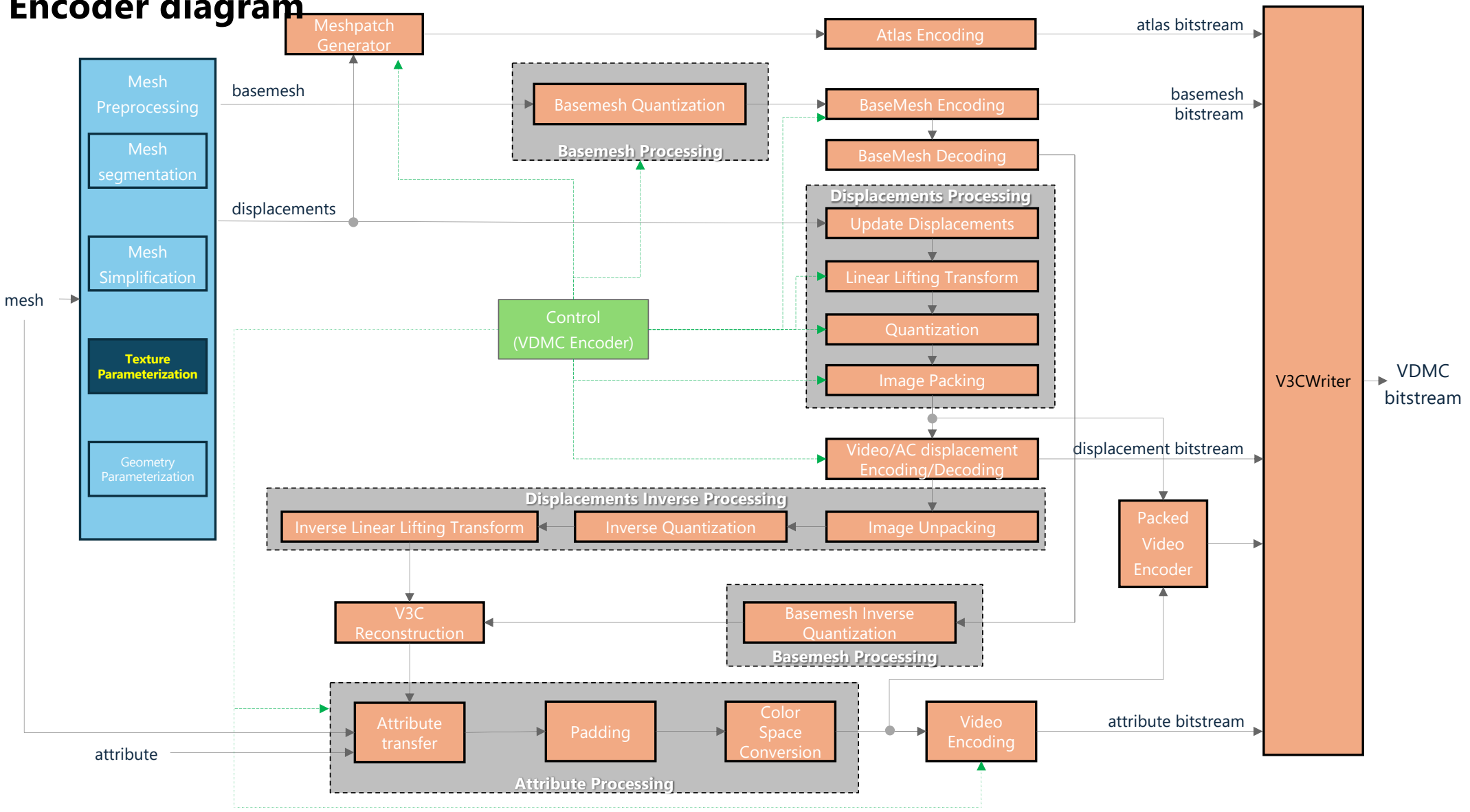
Pre-processing: mesh simplification

- Mapped mesh: mesh that maps the original vertices to the decimated mesh
  - Search for the closest point at the surface of the decimated mesh
  - Influence the error calculation by
    - Preserving the orientation of the original triangle's normal

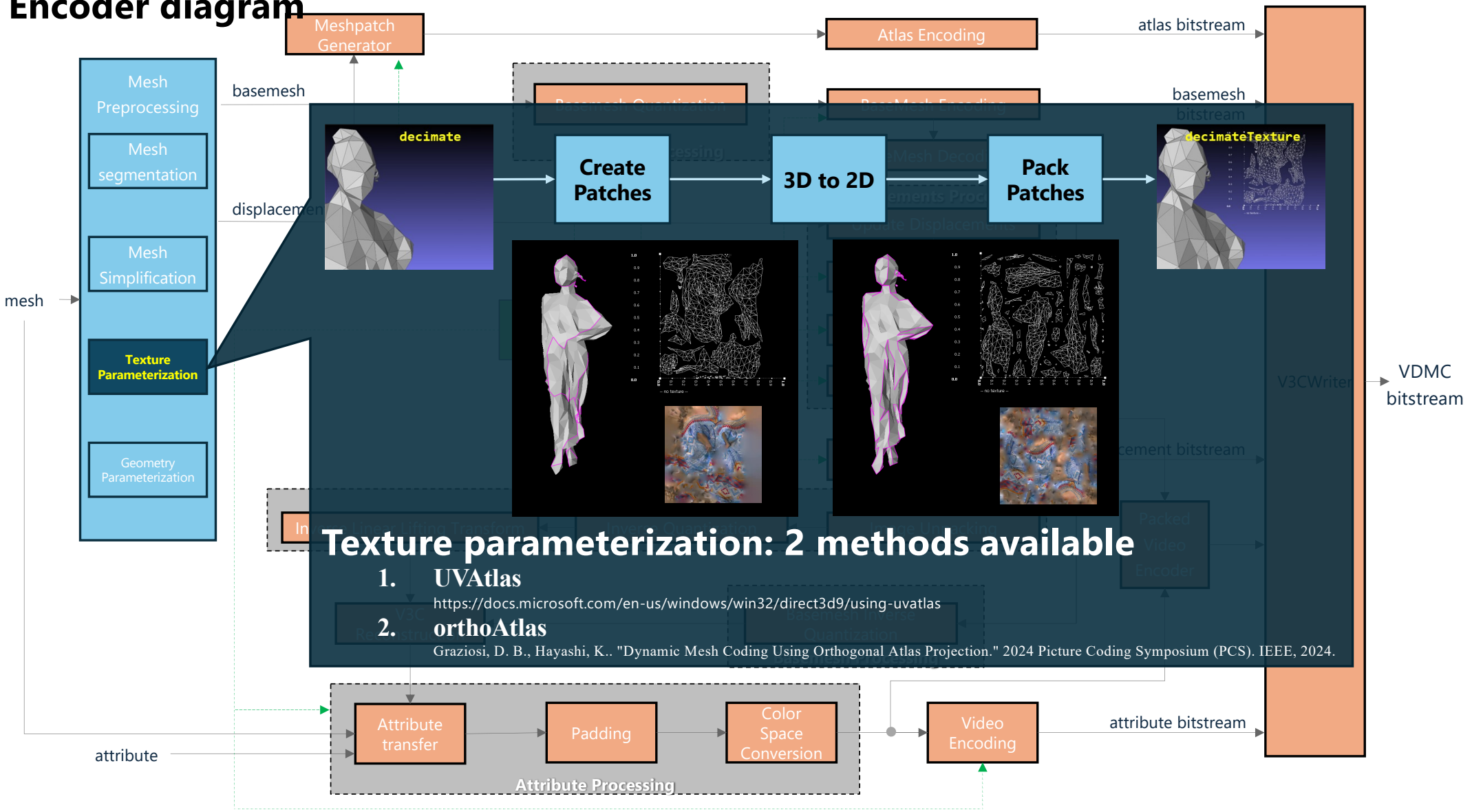




Encoder diagram

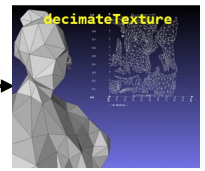


Encoder diagram

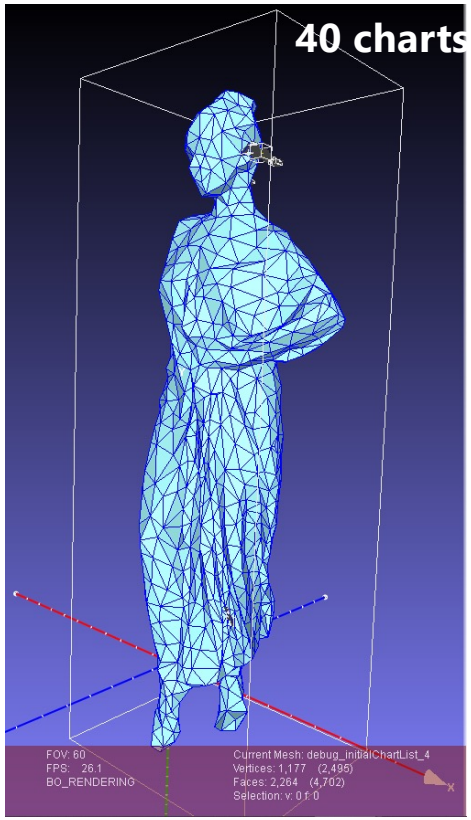


UVAtlas patches

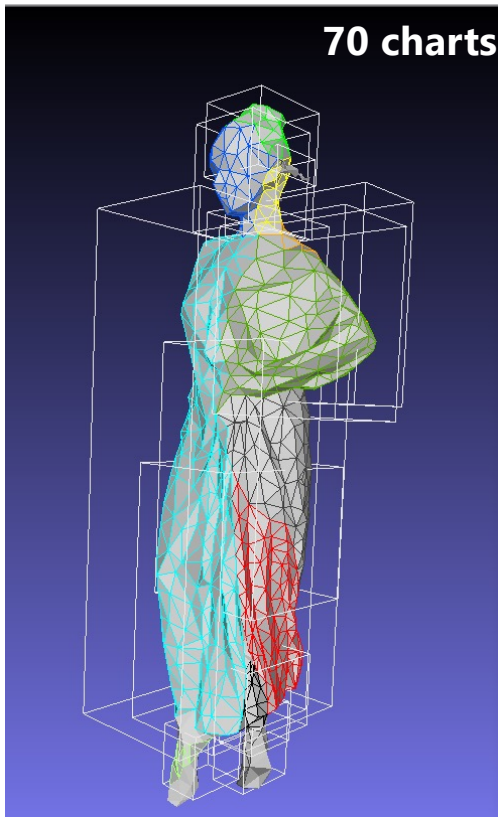
Create Patches → 3D to 2D → Pack Patches →



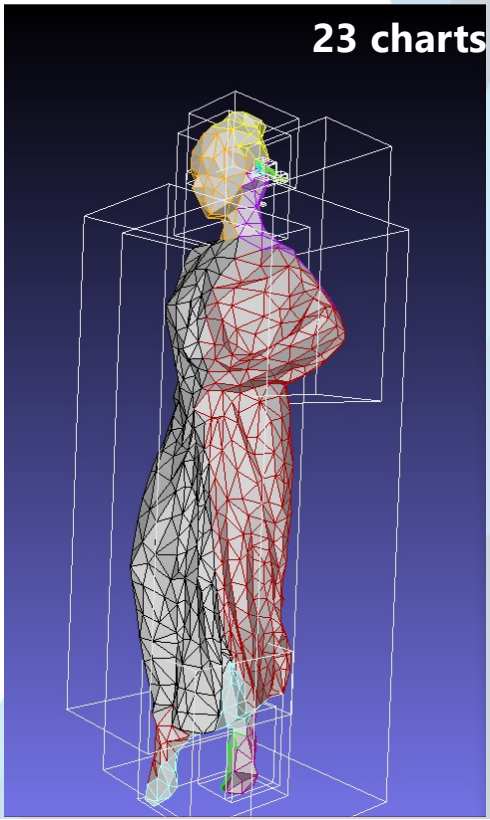
Initialize



Partition (ISO)

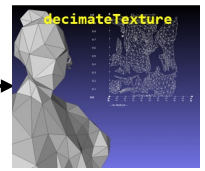


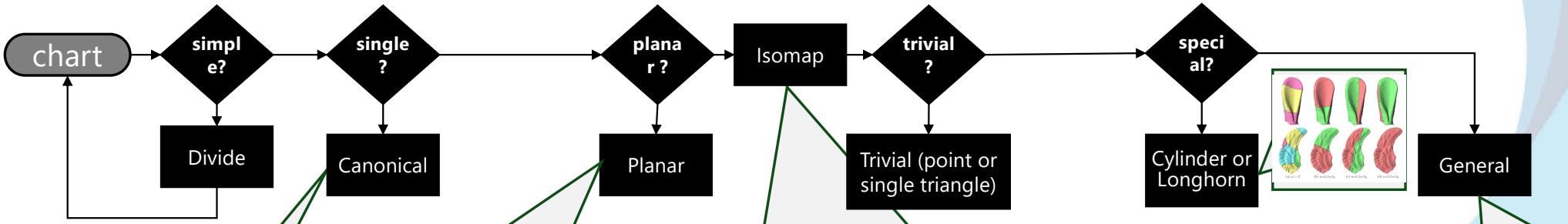
Partition (Merge)

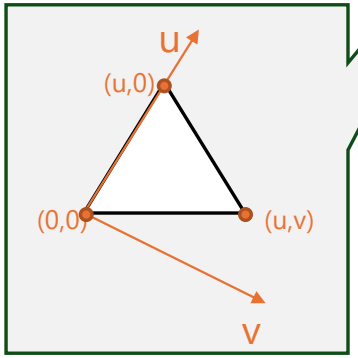


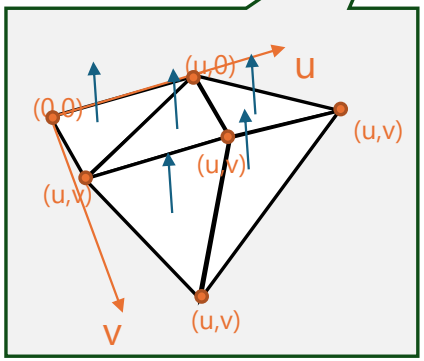
UVAtlas mapping

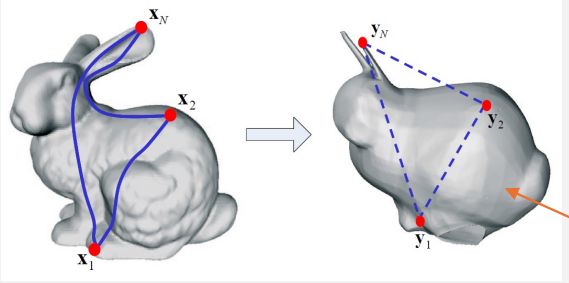
Create Patches → 3D to 2D → Pack Patches →



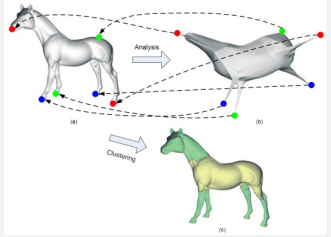






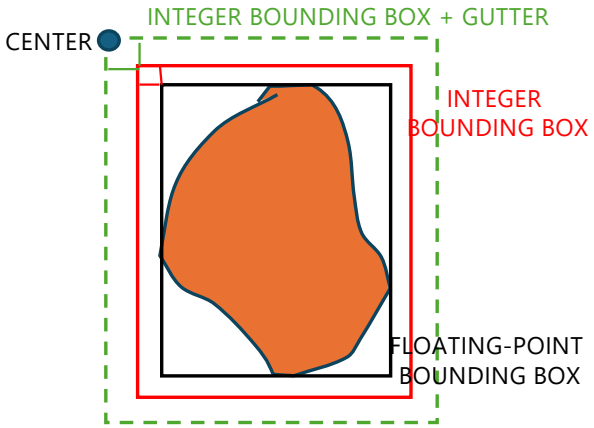


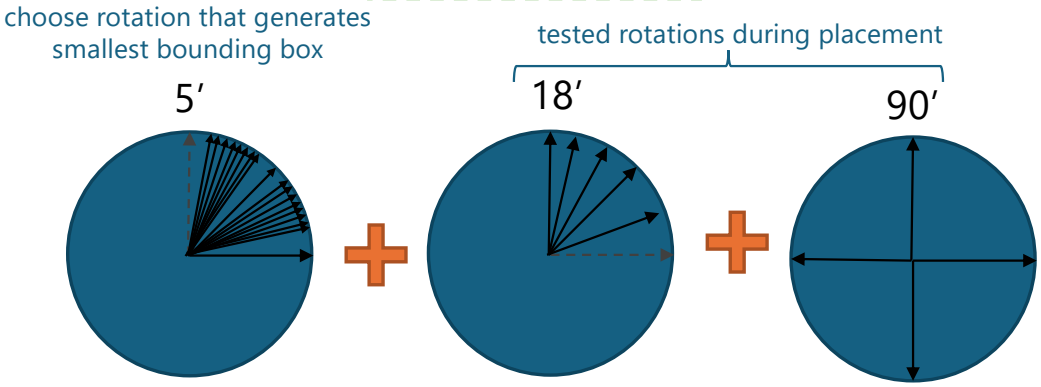
If eigenvalues' energy is small, it will be a trivial parameterization (point)
If the energy is concentrated in first 2 eigenvalues, it may be parameterized as a plane (if no overlaps)
Otherwise, the parameterization uses the reduced set of eigenvectors and eigenvalues (N=90% of the accumulated energy)

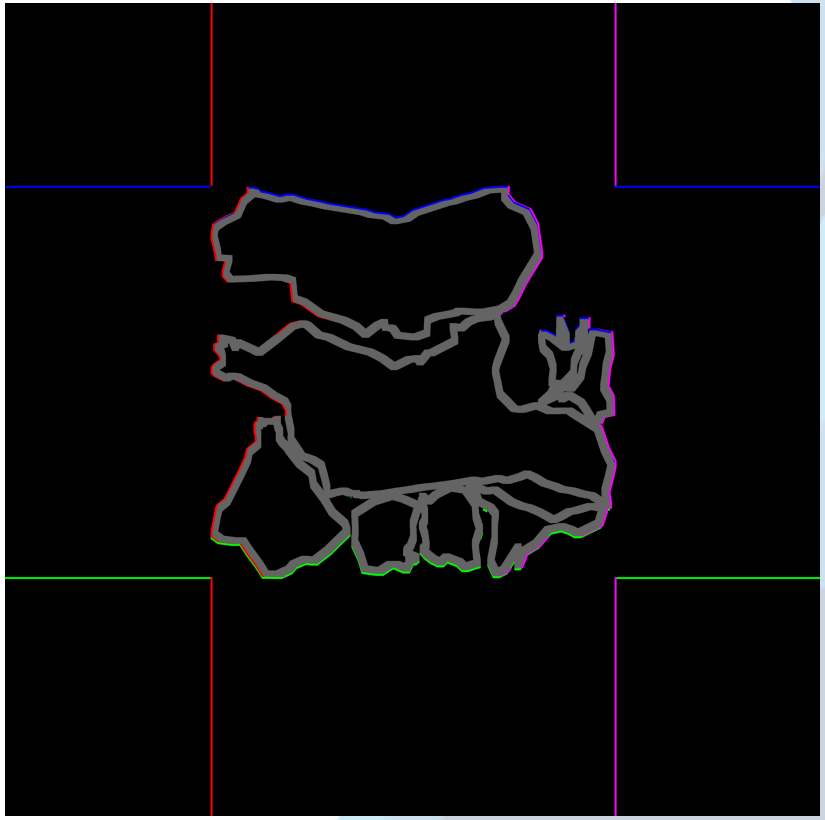


Segment the chart according to the representative vertices (landmarks further away from each other), then cluster triangles and divide the surface into sub-charts for processing

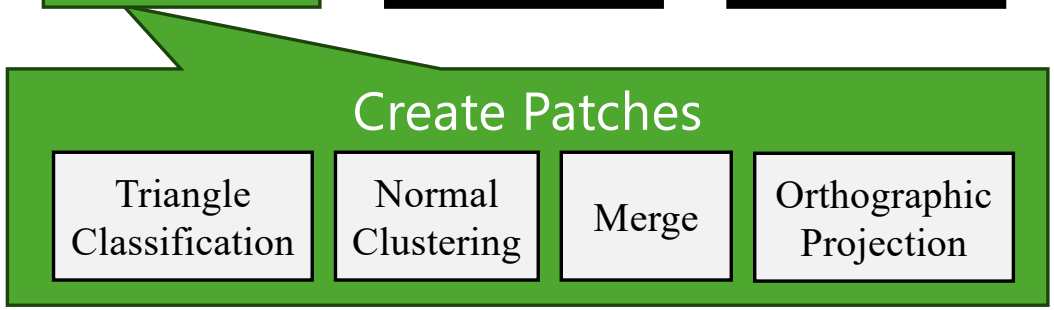
UVAtlas packing







orthoAtlas patches



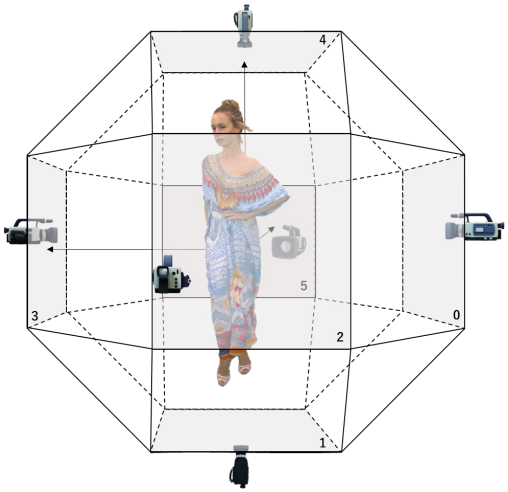
(**2351** triangles)
SUM OF PERIMETER: 186071,
SUM OF STRETCH L2: 2580.93,
AVG. STRETCH L2: 1.0978

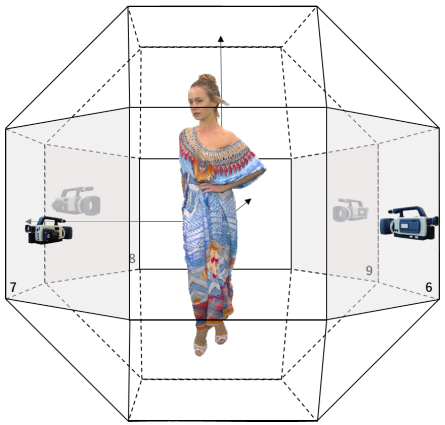
(**251 CC**)
SUM OF PERIMETER: 50721.7,
SUM OF STRETCH L2: 286.232,
AVG. STRETCH L2: 1.14037

(**55 CC**)
SUM OF PERIMETER: 28935.2,
SUM OF STRETCH L2: 90.7831,
AVG. STRETCH L2: 1.6506

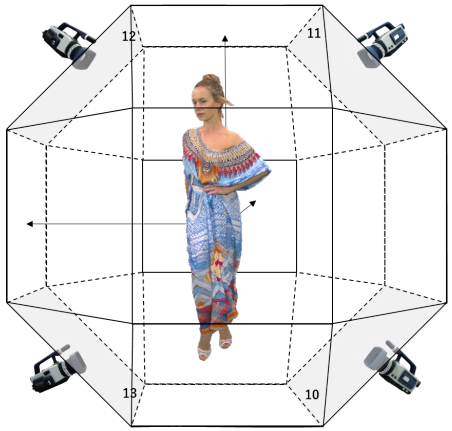
(**68 CC**)
SUM OF PERIMETER: 31455.1,
SUM OF STRETCH L2: 115.134,
AVG. STRETCH L2: 1.69315

orthoAtlas mapping

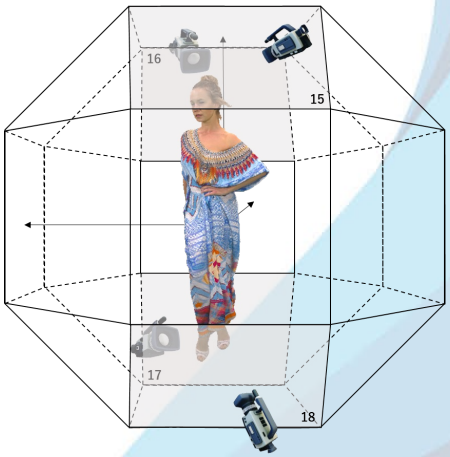




Rotation around Y-axis

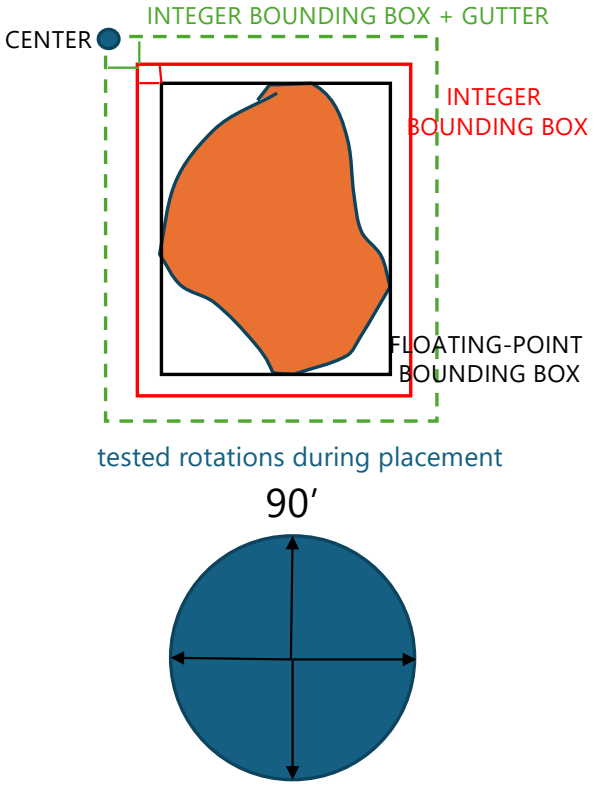


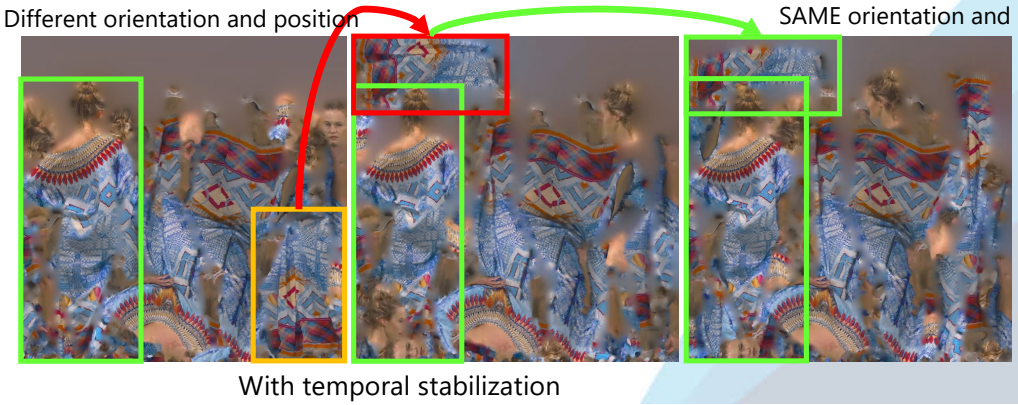
Rotation around Z-axis



Rotation around X-axis

orthoAtlas packing

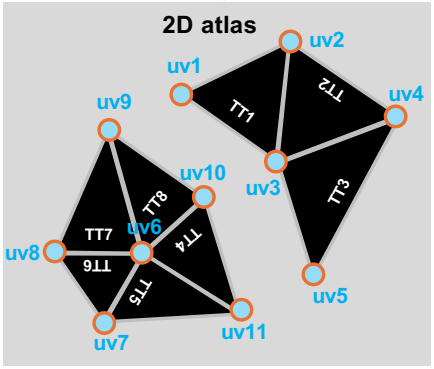




orthoAtlas signaling

- Atlas bitstream is used to transmit orthoAtlas projection parameters

Patch ID information



Basemesh

Vertex list
V1 (x1, y1, z1)
…
V8 (x8, y8, z8)

Triangle list
T1 (v1, v2, v3)
…
T8 (v1, v3, v6)

UV list
UV1 (**Patch1**, 0)
UV2 (**Patch1**, 0)
UV3 (**Patch1**, 0)
…
UV9 (**Patch2**, 0)
UV10 (**Patch2**, 0)
UV11 (**Patch2**, 0)

UV Triangle list
TT1 (uv1, uv2, uv3)
…
TT8 (uv9, uv10, uv6)

Atlas bitstream

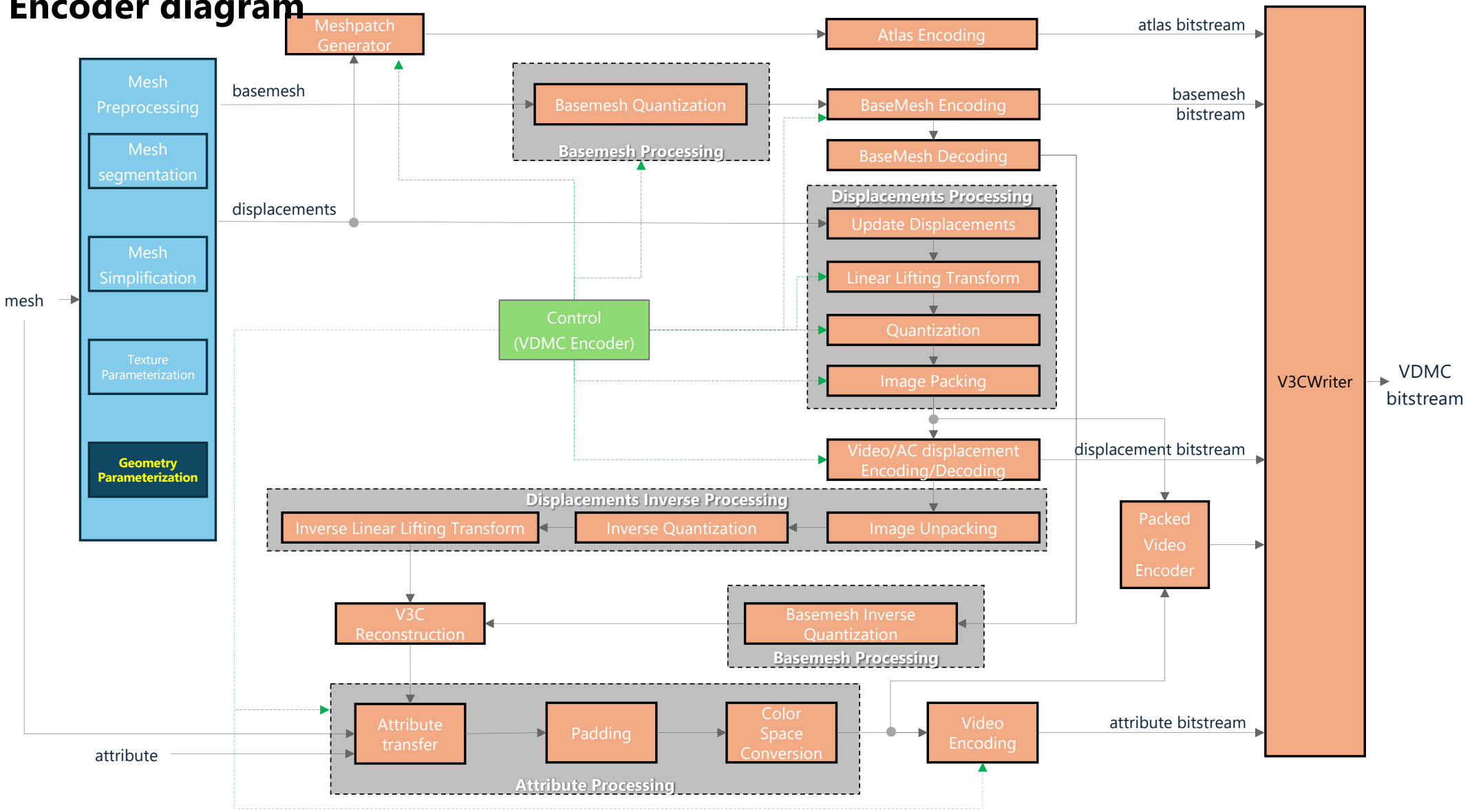
- Projection direction ID
- Orientation ID
- Patch position (x, y)
- Patch size (sizeX, sizeY)
- Patch scale

Can use UV coordinates, Face ID attribute, or implicitly derived using connected components

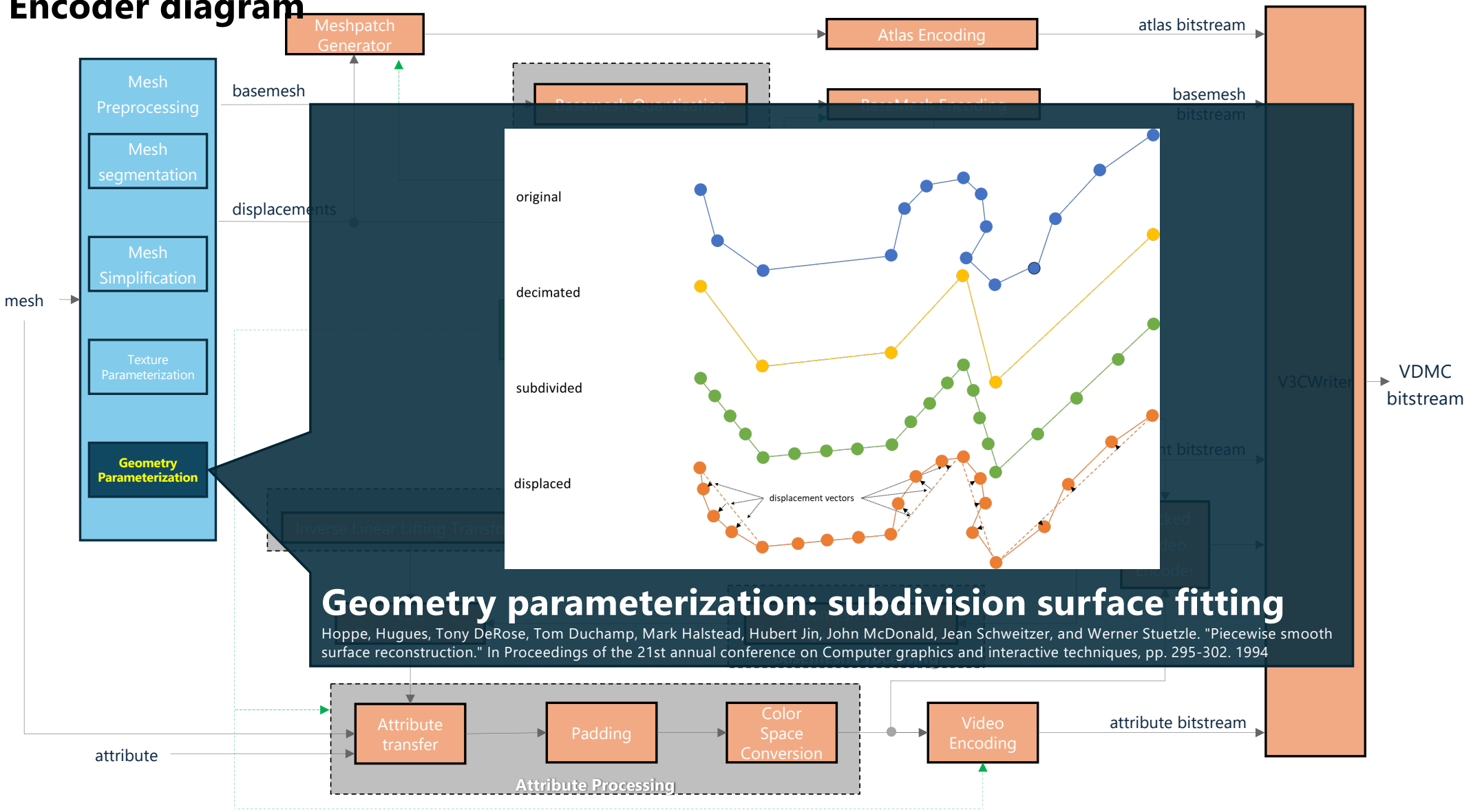
Projection

orthoAtlas vs. UVAtlas (anchor)

Encoder diagram



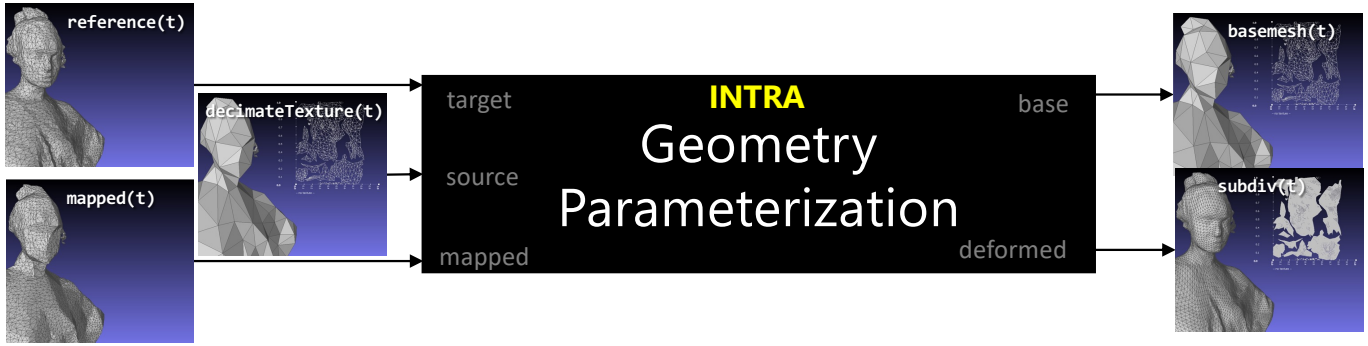
Encoder diagram



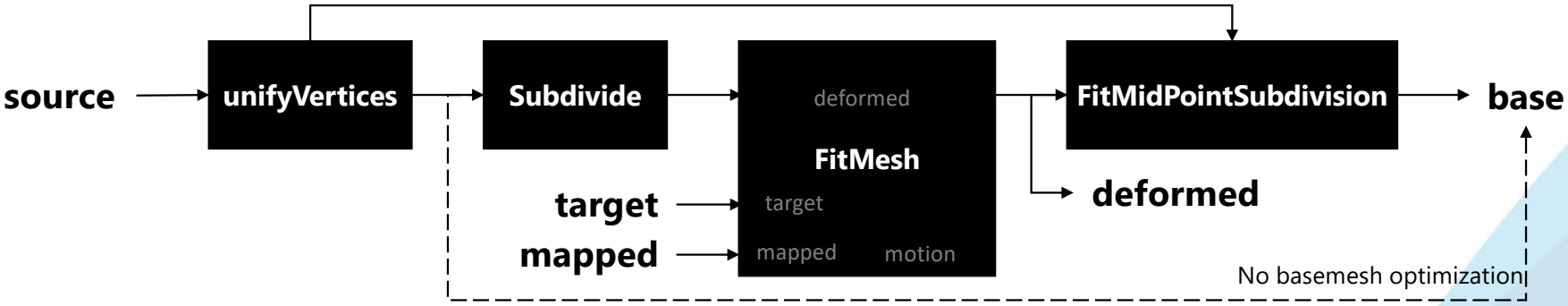
Geometry parameterization: subdivision surface fitting

Hoppe, Hugues, Tony DeRose, Tom Duchamp, Mark Halstead, Hubert Jin, John McDonald, Jean Schweitzer, and Werner Stuetzle. "Piecewise smooth surface reconstruction." In Proceedings of the 21st annual conference on Computer graphics and interactive techniques, pp. 295-302. 1994

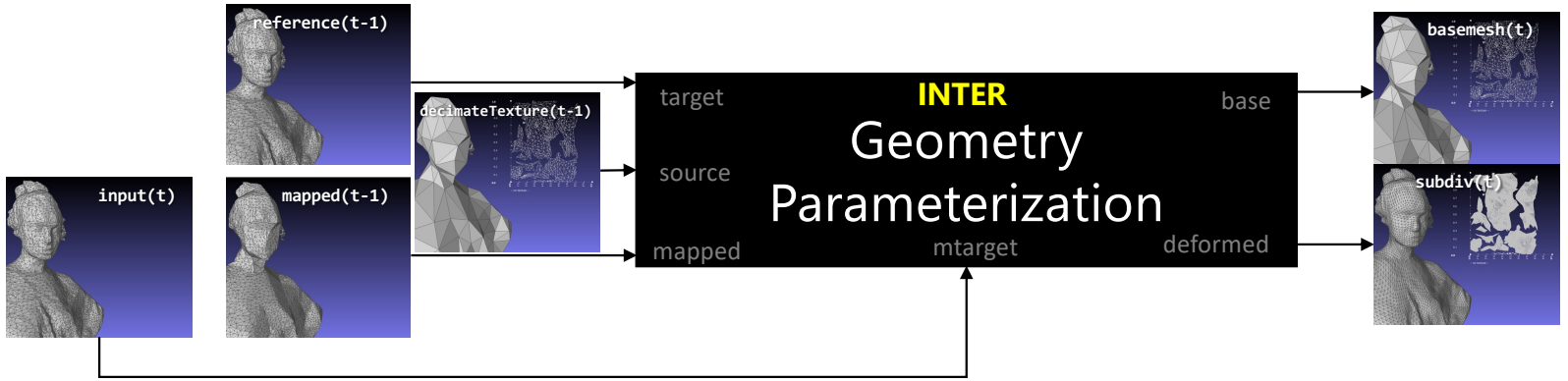
INTRA geometry parameterization



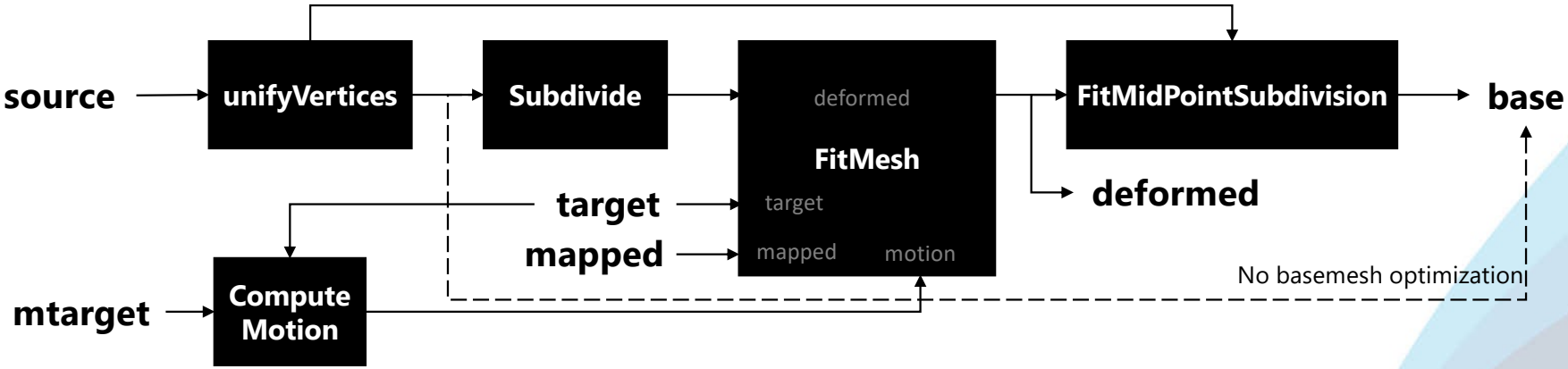
Fit the **source** (decimated mesh) to the **target** (reference input mesh)



INTER geometry parameterization **with** mapping

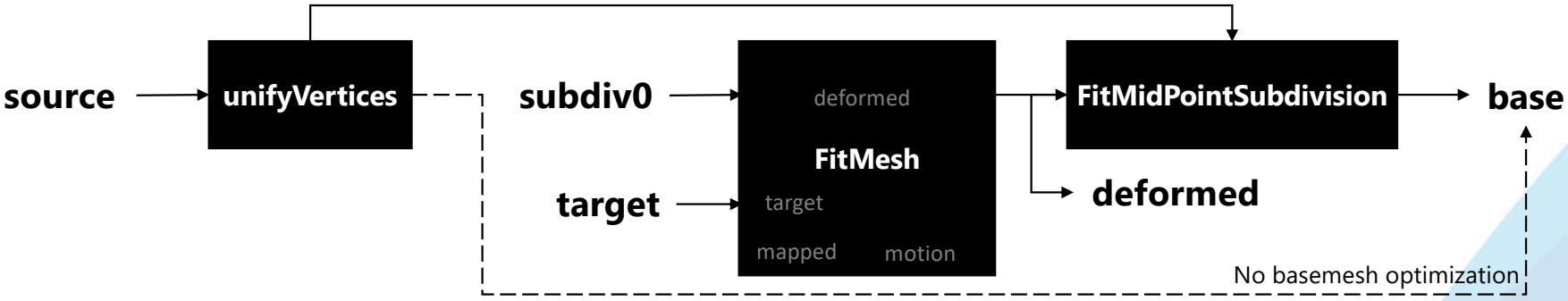


Fit the **source** (previous decimated mesh) to the **target** (previous input) using motion vectors

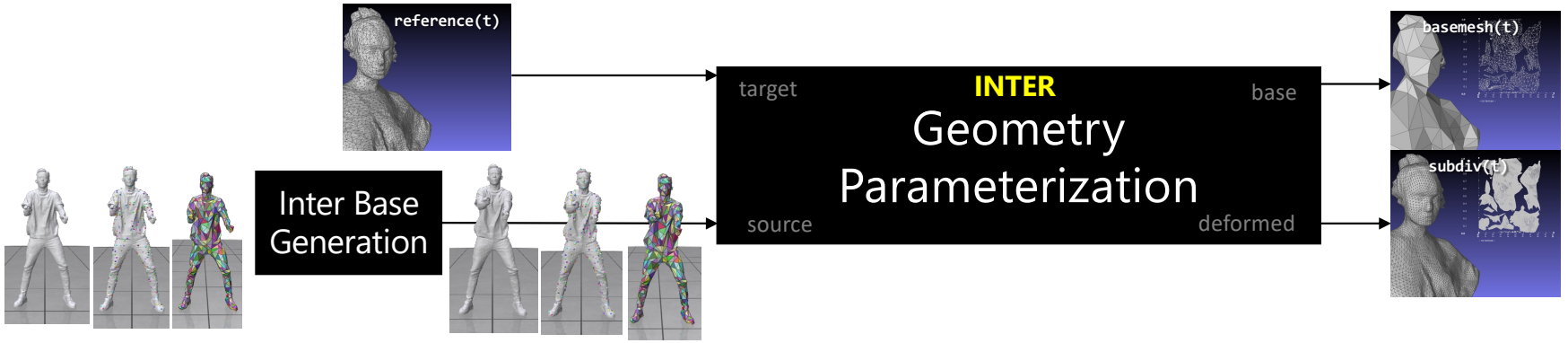


INTER geometry parameterization **without** mapping

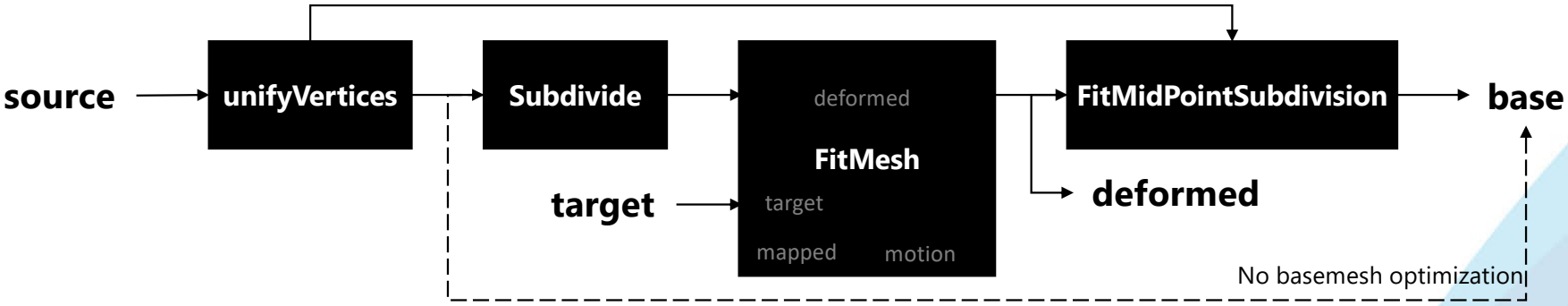
Fit the **source** (previous subdivided decimated mesh) to the **target** (current input)



INTER geometry parameterization **without** mapping

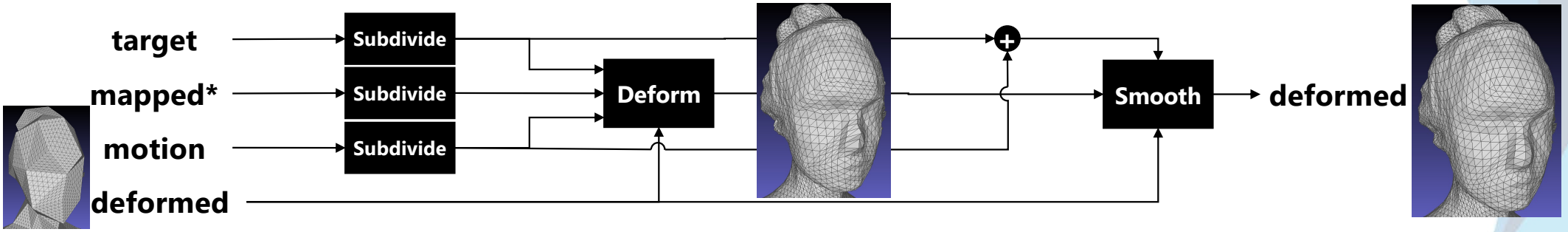


Fit the **source** (tracked basemesh) to the **target** (current input)

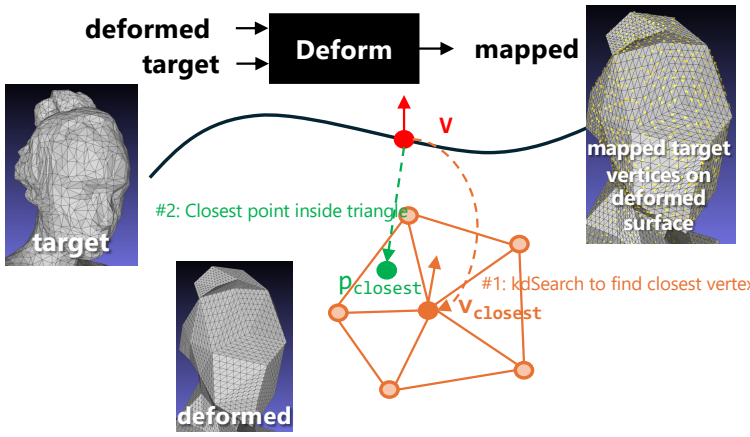


Jin, X., Xu, J. and Kawamura, K., 2024, April. Embedded Graph Representation for Inter-Frame Coding of Dynamic Meshes. In ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 4070-4074). IEEE.

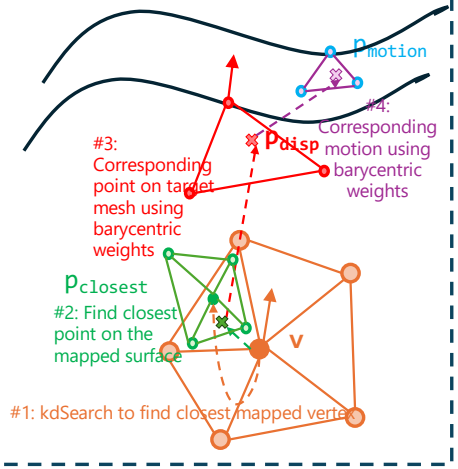
Fitmesh: surface fitting between two meshes



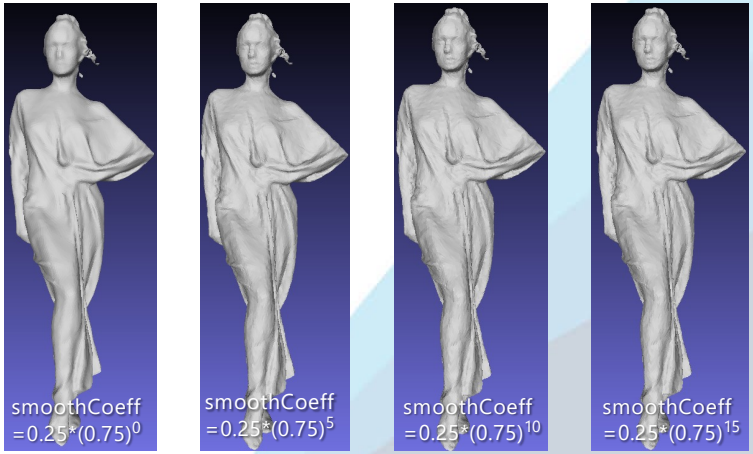
***When mapped is missing, create it by deforming the target to the subdivided basemesh**



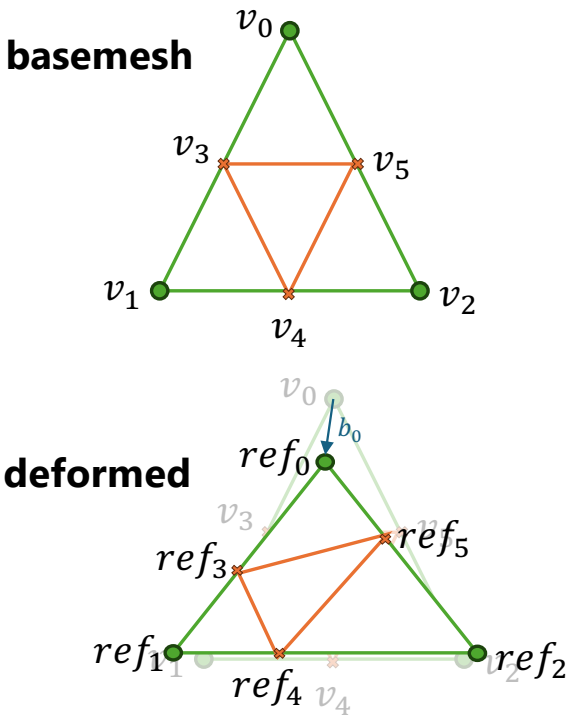
Find the closest point in the surface of the mapped mesh and the motion compensated target mesh



When smoothing is selected, apply iterative graph smoothing with decay ration



FitMidPointSubdivision: basemesh post-processing



$$\begin{bmatrix} v_0 \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1/2 & 1/2 & 0 \\ 0 & 1/2 & 1/2 \\ 1/2 & 0 & 1/2 \end{bmatrix}}_{M} * \begin{bmatrix} v_0 \\ v_1 \\ v_2 \end{bmatrix}$$

Midpoint subdivision can be model by matrix operation

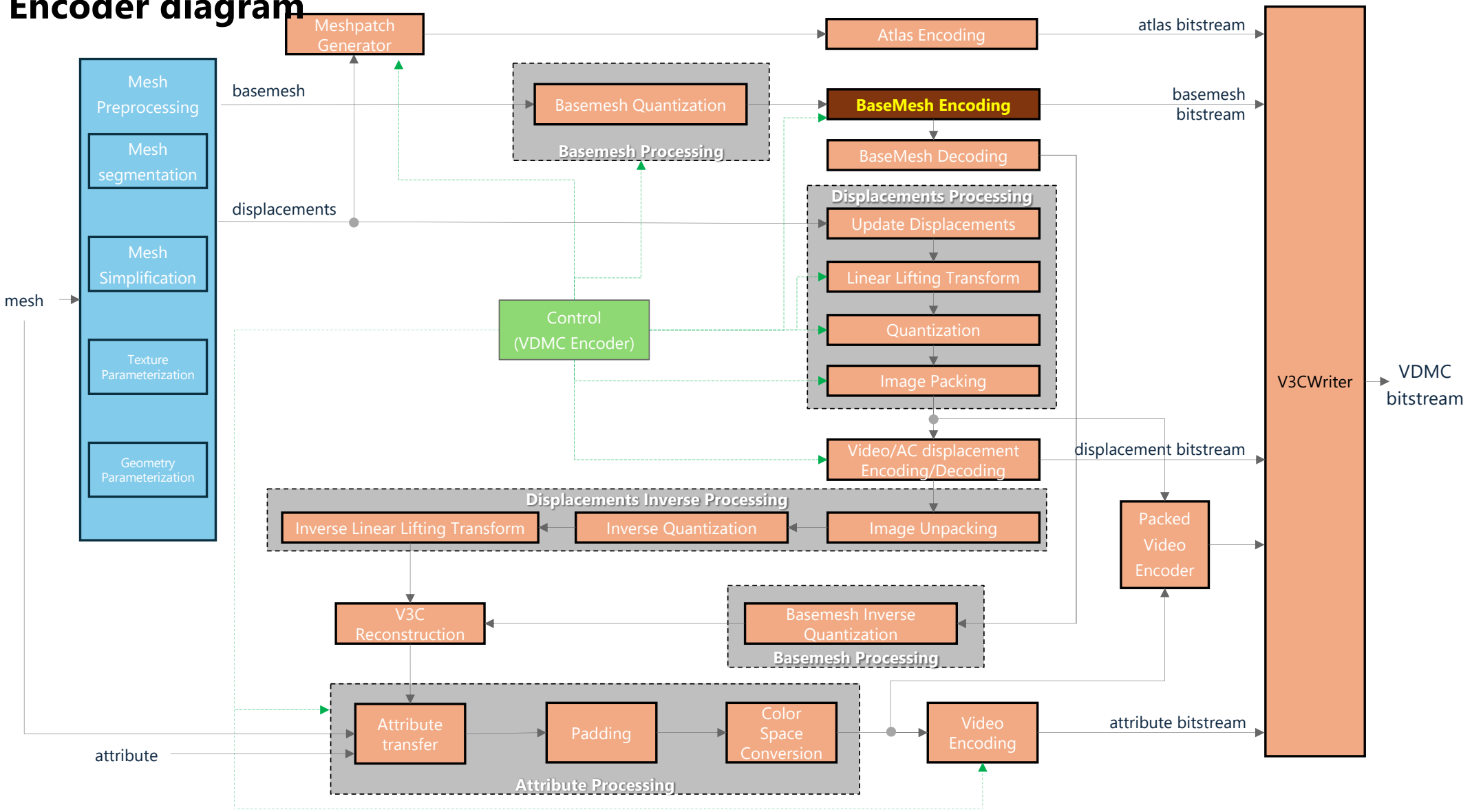
$$\begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \end{bmatrix} = \begin{bmatrix} ref_0 \\ ref_1 \\ ref_2 \\ ref_3 \\ ref_4 \\ ref_5 \end{bmatrix} - \begin{bmatrix} v_0 \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{bmatrix}$$

Difference between the deformed mesh and the subdivided basemesh are the displacement vectors.

$$\begin{bmatrix} r_0 \\ r_1 \\ r_2 \end{bmatrix} = M^T * \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \end{bmatrix} \Rightarrow error = \frac{{r_0}^2 + {r_1}^2 + {r_2}^2}{3}$$

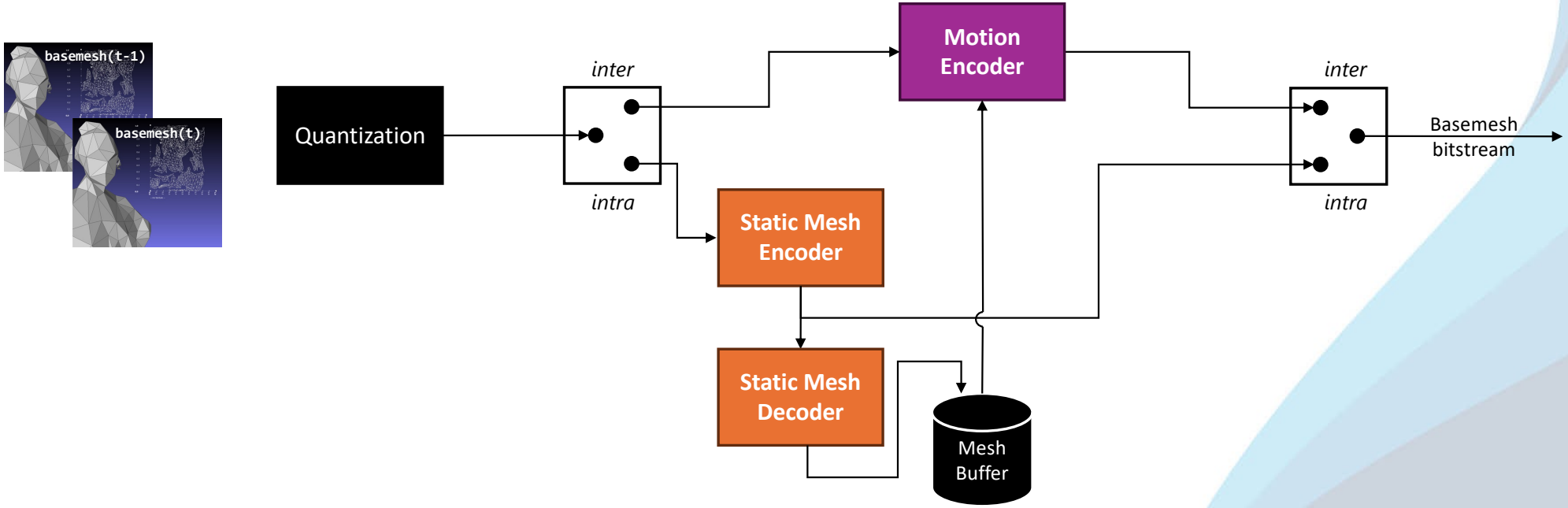
Apply conjugate gradient method to solve a least square problem and modify the vertex position of the basemesh to reduce the displacement

Encoder diagram



Basemesh codec structure

- Specification defines a basemesh codec that can select any static mesh codec and motion codec, indicated by a codec ID. FourCC codes can be used for codec selection as well.



Basemesh coding

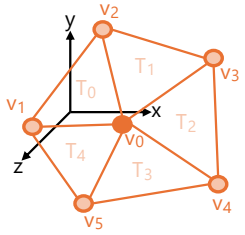
- Static mesh codec (Annex I): MEB (MPEG Edgebreaker)

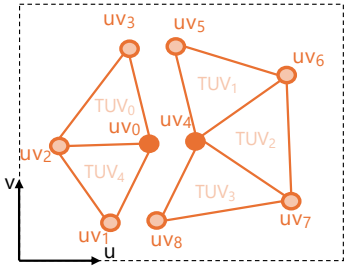
vertex positions			
V_{idx}	X	Y	Z
0	x_0	y_0	z_0
1	x_1	y_1	z_1
2	x_2	y_2	z_2
...			

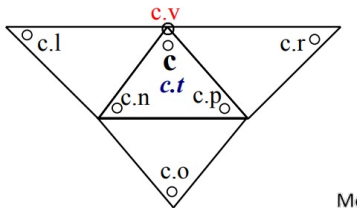
vertex uv coord.		
UV_{idx}	U	V
0	u_0	v_0
1	u_1	v_1
2	u_2	v_2
3	u_3	v_3
...		

triangle			
T_{idx}	V_0	V_1	V_2
0	0	1	2
1	0	2	3
...			

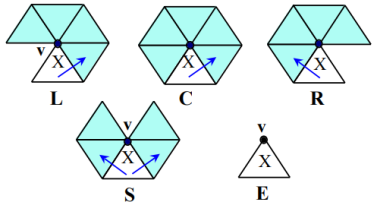
triangle uv			
TUV_{idx}	UV_0	UV_1	UV_2
0	0	2	3
1	4	5	6
...			

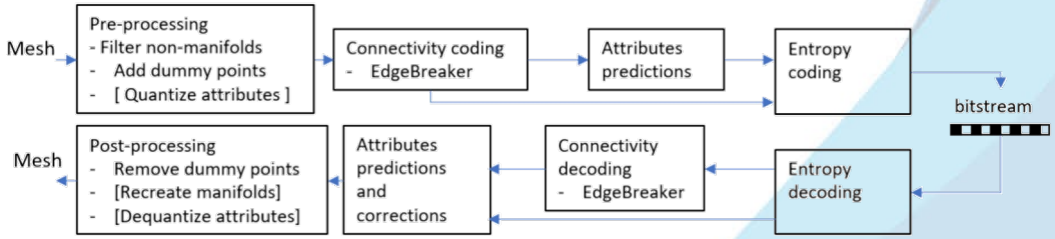






Convert to CornerTable Mesh





Include several other tools that were optimized during the standardization process

Basemesh coding

- Motion codec (Annex L) :
  - Tracked mesh motion coding
    - Duplicate vertex motion coding

Vertex to be encoded/decoded

Already encoded/decoded neighbor

Non-encoded/decoded neighbor

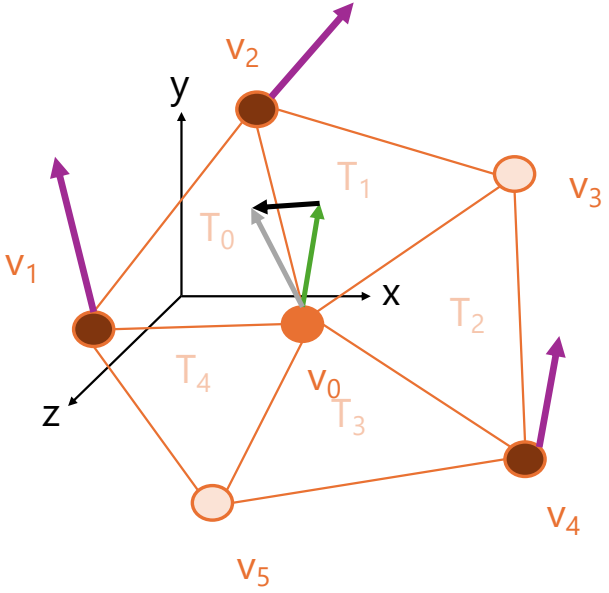
Duplicate vertex to be encoded/decoded

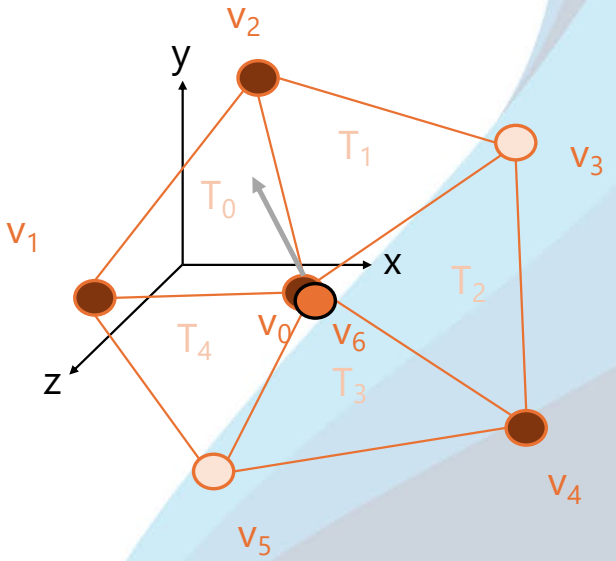
Available motion vector

Predicted motion vector

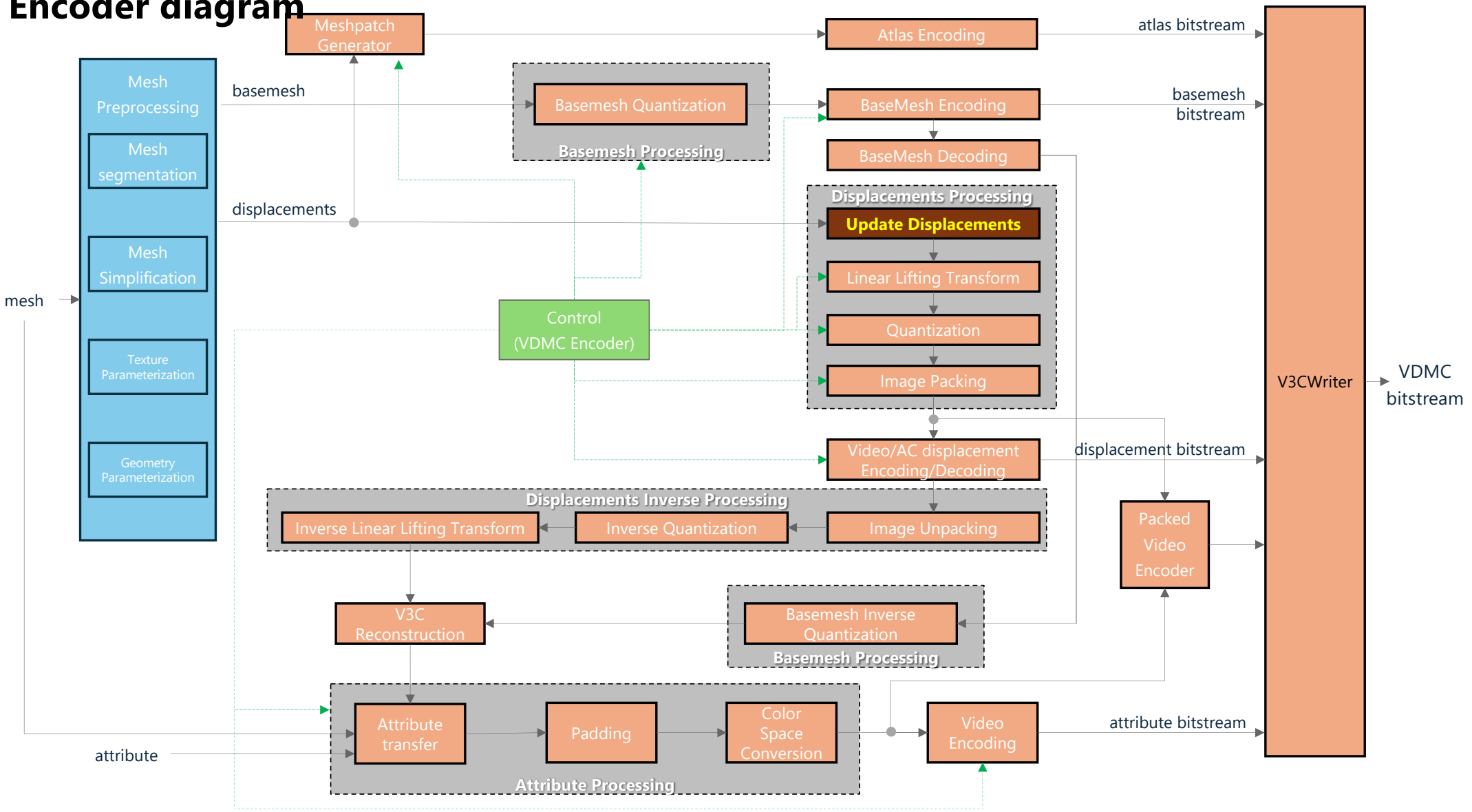
Real motion vector

Prediction residual



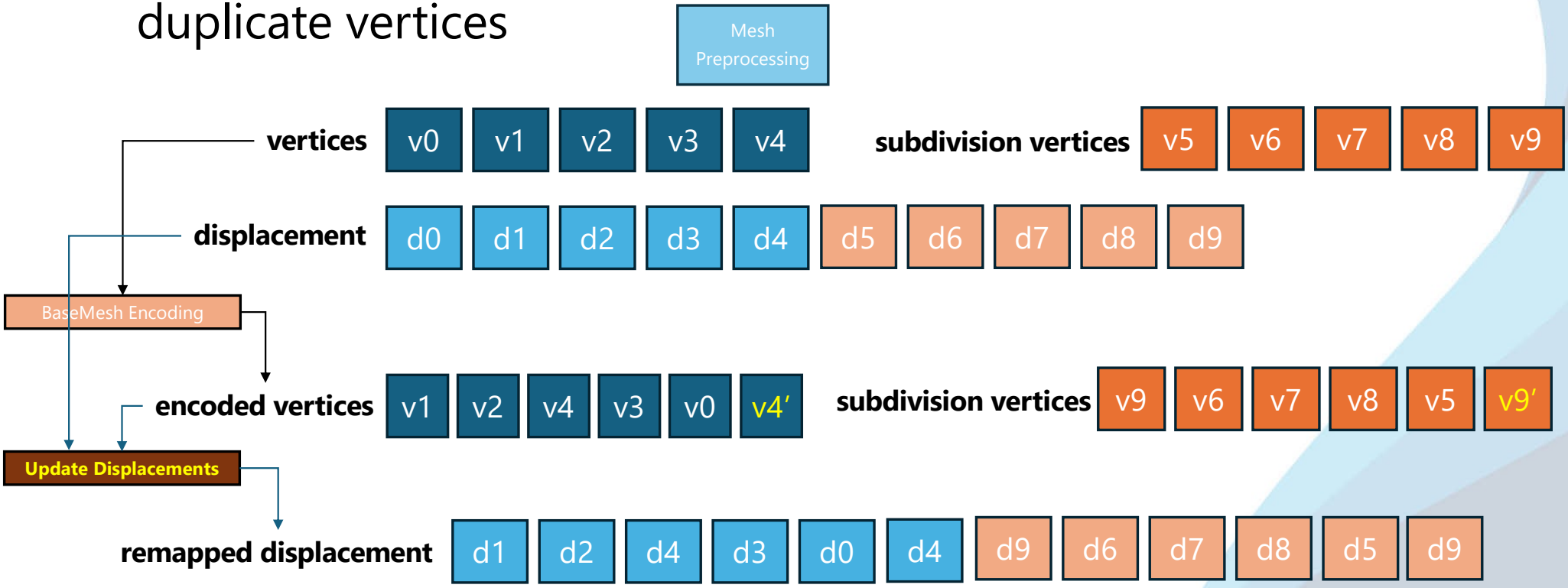


Encoder diagram



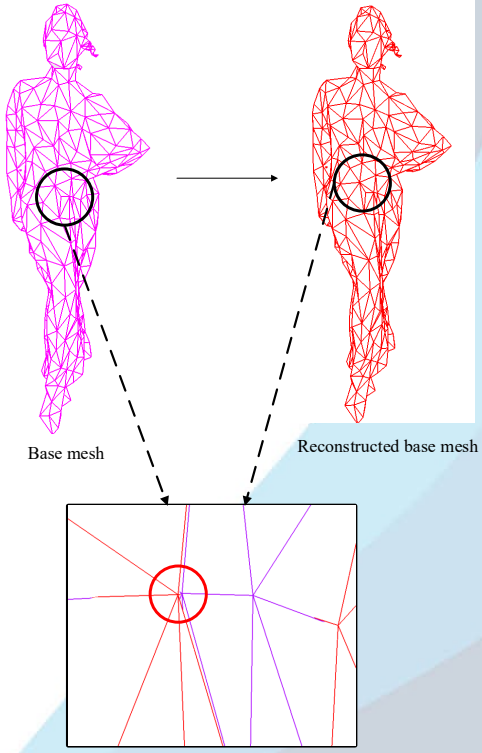
Remapping encoded basemesh to disparity

- Basemesh encoding may change vertex order and introduce new duplicate vertices



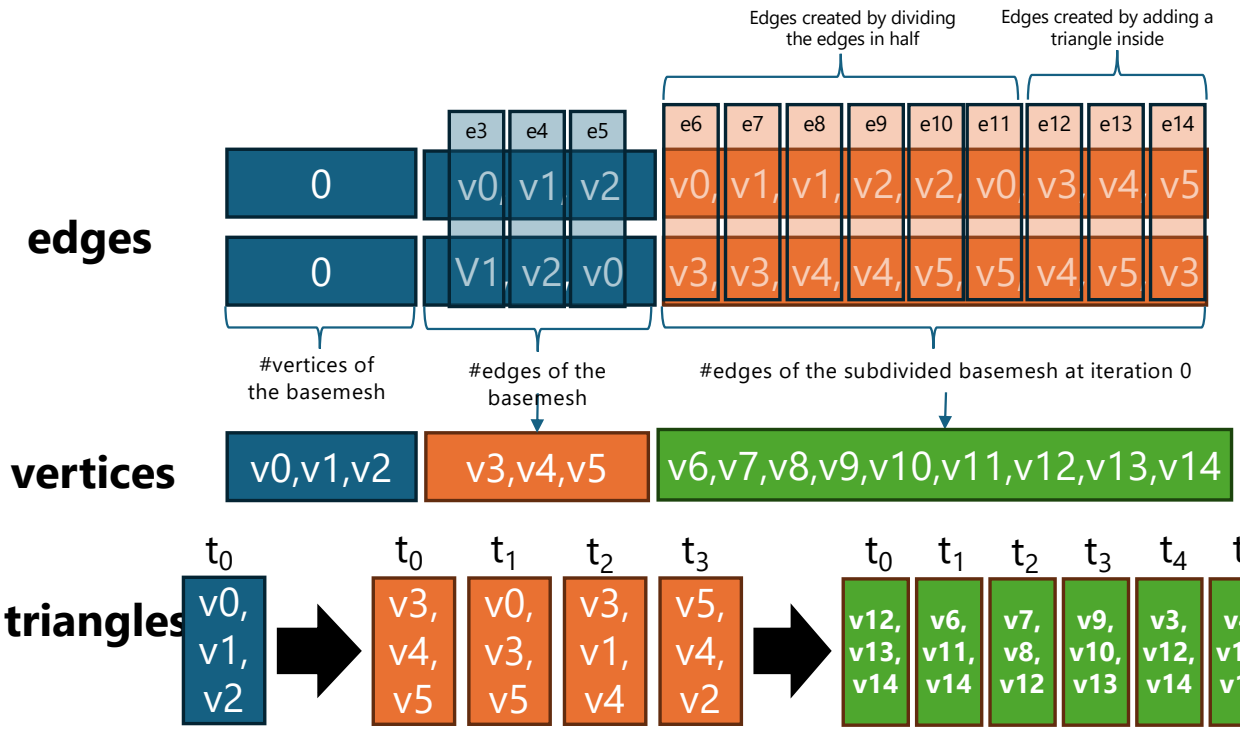
Displacement optimization

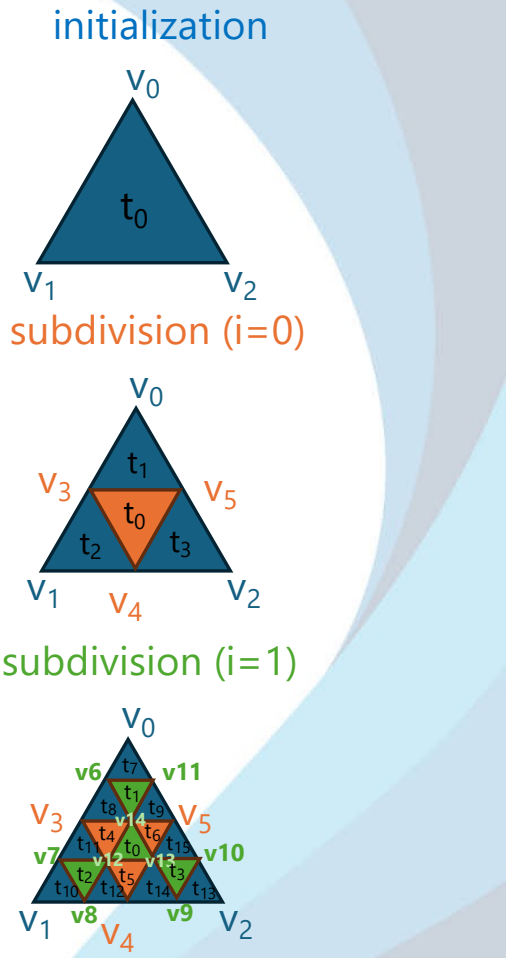
- Optimization by using the decoded basemesh instead of the uncompressed basemesh.
  - geometry parameterization algorithm is called again to fine tune the displacement values



Basemesh subdivision

- Midpoint $v[n] = \frac{(e[n][0] + e[n][1])}{2}$

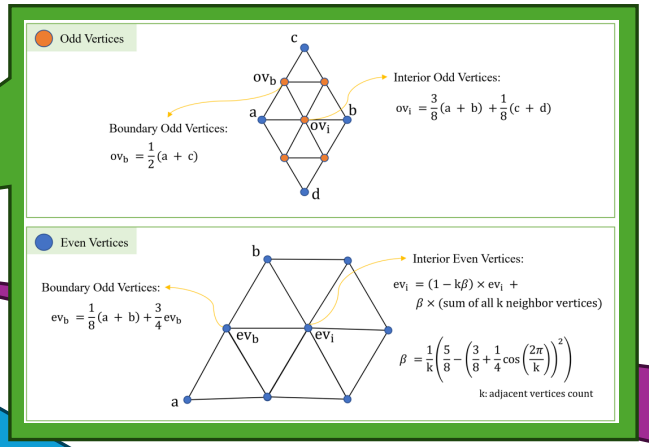




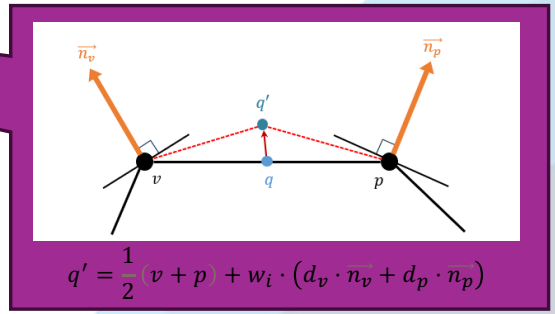
Subdivision methods supported by V-DMC

- Additional methods introduce filtering in the vertex generation phase

asve_subdivision_method afve_subdivision_method mdu_subdivision_method	Name of subdivision method
0	MIDPOINT
1	LOOP
2	NORMAL
3	PYTHAG
4..7	RESERVED



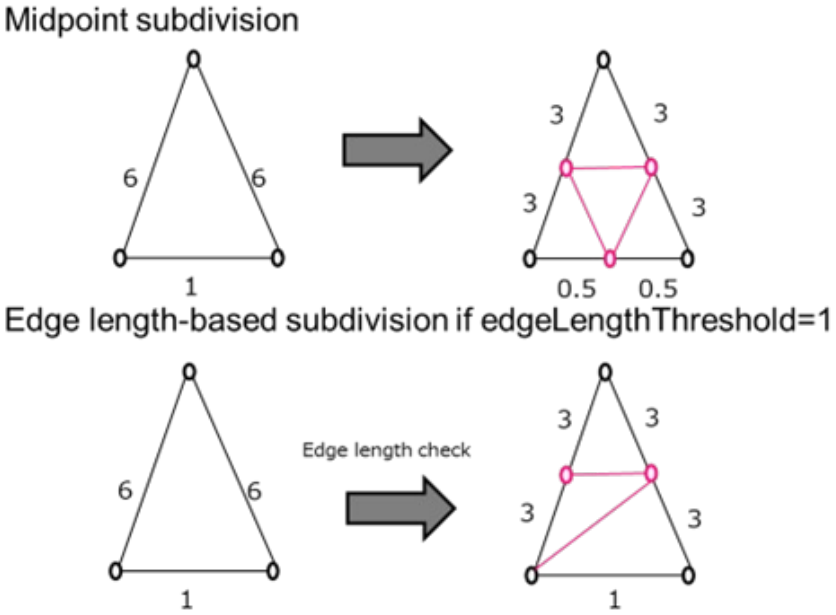
Arithmetic mean : $AM = (x_1 + x_2 + \cdots + x_n)/n$
Geometric mean : $GM = \sqrt[n]{x_1 \times x_2 \times \ldots \times x_n}$
Harmonic mean : $HM = n/(1/x_1 + 1/x_2 + \cdots + 1/x_n)$



m66335	145 - OnLine	[V-DMC][EE4.15] Experimental report of Loop and LS3 subdivision methods for EE4.15 Test 2	Seong Yong Lim, Da-Yun Nam
m67581	146 - Rennes	[V-DMC][EE4.15-related][New] Normal-based subdivision method	Dae-Hyun Kim, Jong-Yeul Suh, Han-Je Park, Seo-kyoung Koh, Na-seong Kwon (LG Electronics), Joo-Hyung Byeon, Min-Tae Kim, Dong-Gyu Sim (Kwangwoon Univ.)
m68785	147 - Sapporo	[V-DMC][EE4.15-related][New] Alternative Means for Mesh Subdivision Prediction	Maja Krivokuća (InterDigital), Bertrand Chupeau (InterDigital)

Edge-adaptive subdivision

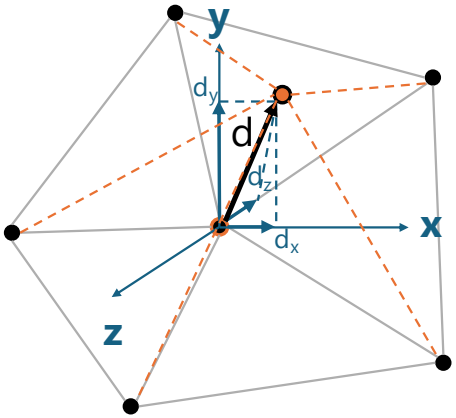
- Control the number of triangles by imposing an edge length threshold



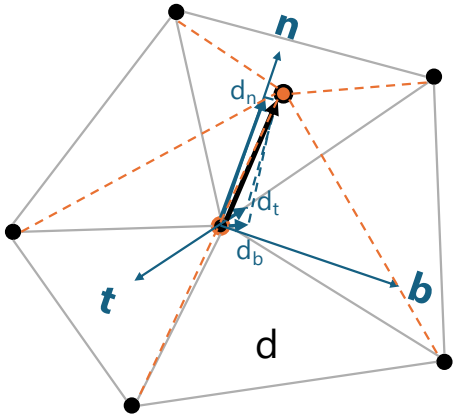
Tomoya Naganuma, Danillo Graziosi, Ali Tabatabai, Kao Hayashi, Satoru Kuma, Ohji Nakagami. "DYNAMIC MESH CODING USING EDGE LENGTH-BASED ADAPTIVE SUBDIVISION." *2025 International Conference on Image Processing (ICIP)*. IEEE, 2025.

Displacement representation

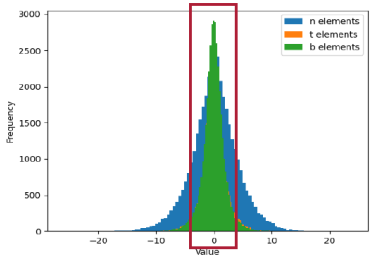
- 3 components expressed in local vs. canonical coordinate system
  - 1D coding of normal component only for local c



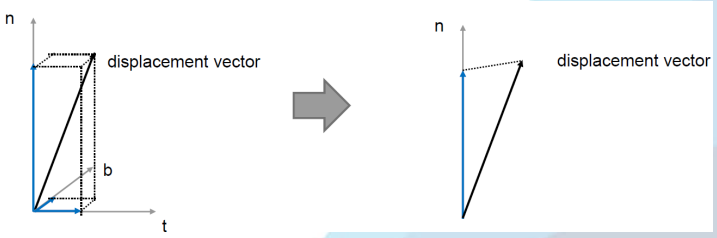
canonical coordinate system



local coordinate system



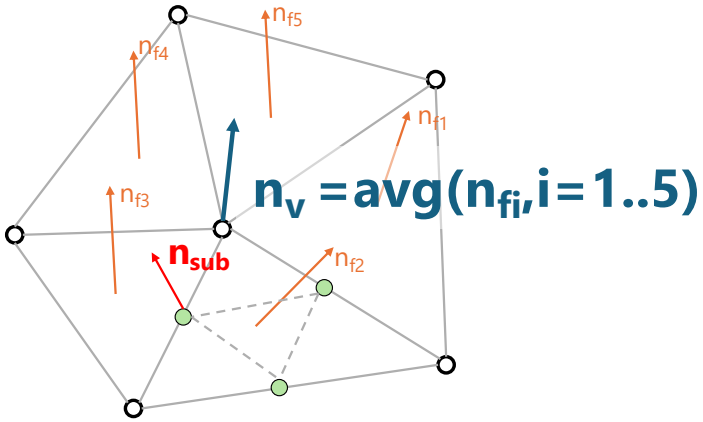
Distribution of each element of the displacements.

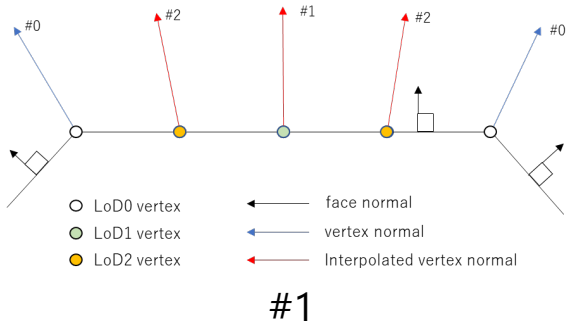


1D displacement coding

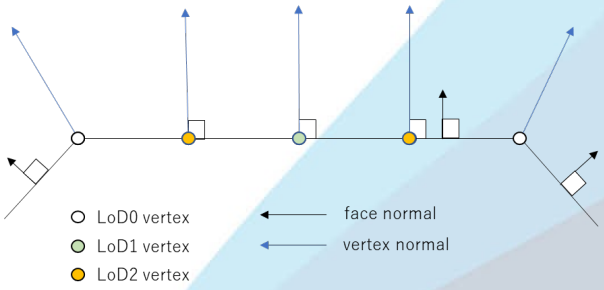
Normal calculation

- Basemesh vertices → average of connected face normals
- Normal for subdivided meshes
  1. Interpolate normal vertices from supporting edge (from higher LoD)
     - Enable LoD processing
  2. Re-calculate normal averaging face normals from connected triangles





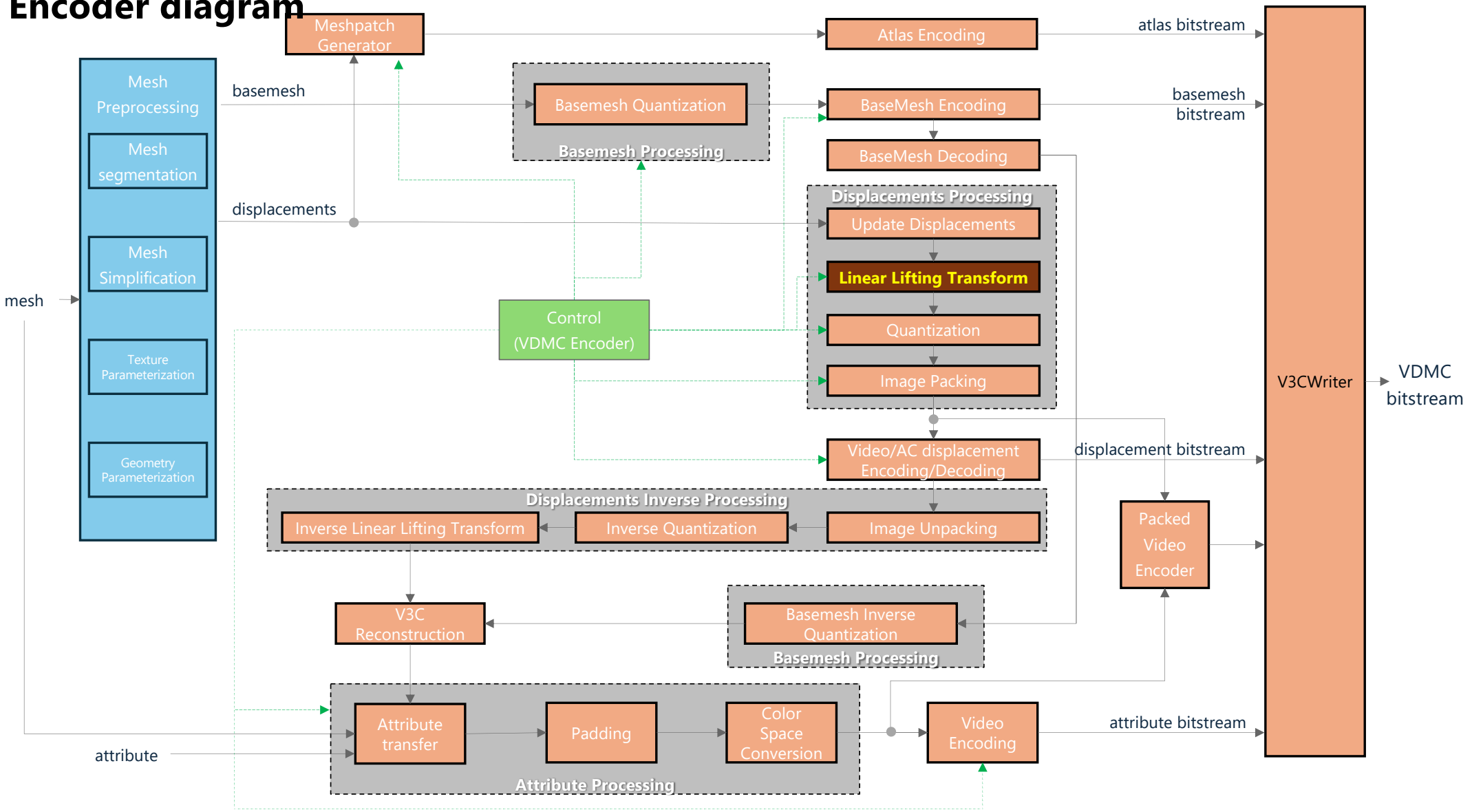
#1



#2

m61020	140 - Mainz	[V-DMC][New Proposal] Improvement on local coordinate system for displacement coding	Satoru Kuma, Kao Hayashi, Danillo Graziosi, Ohji Nakagami
m68374	147 - Sapporo	[V-DMC] On Subdivided Normals and LoD reconstruction	Patrice Rondao Alface, Lukasz Kondrad

Encoder diagram



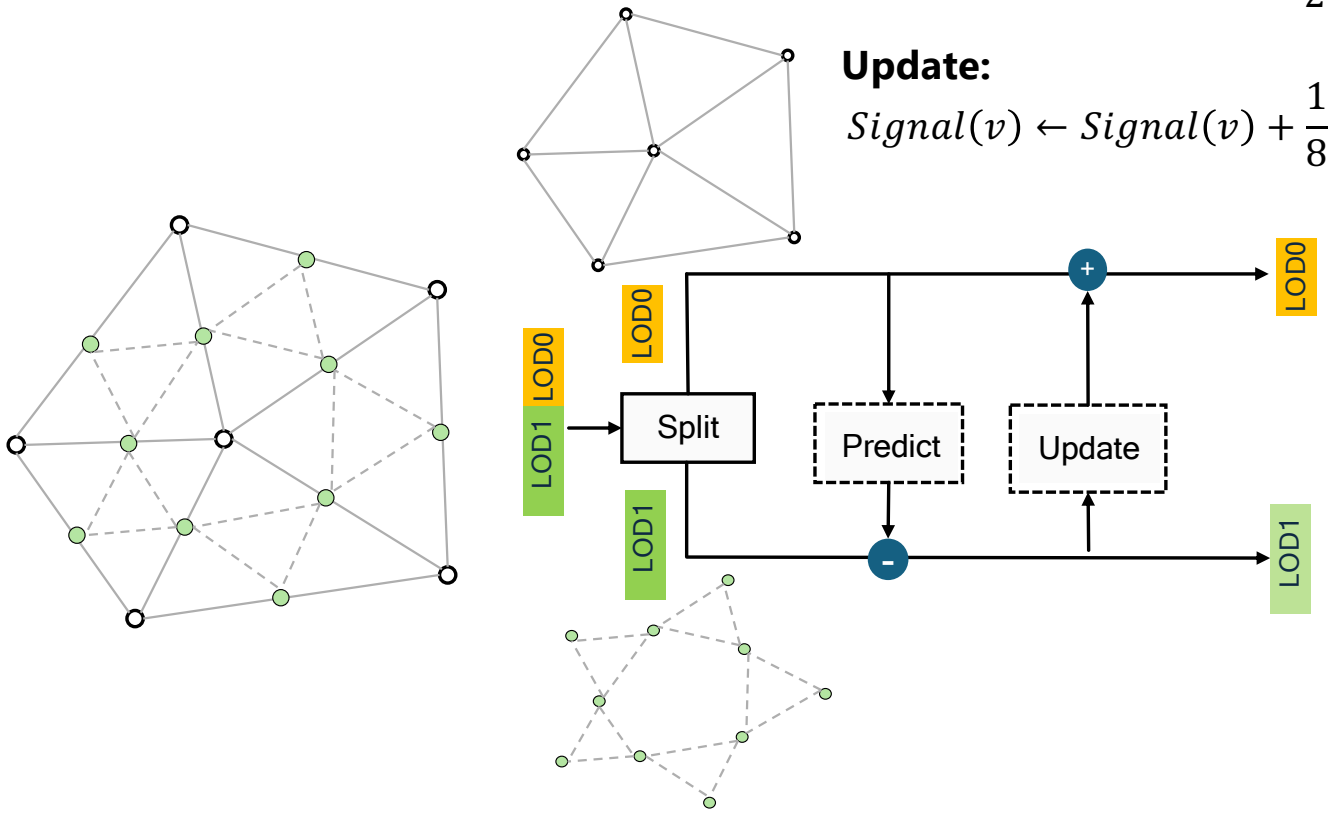
Linear Lifting transform

Prediction:

$$Signal(v) \leftarrow Signal(v) - \frac{1}{2}\left(Signal(v_1) + Signal(v_2)\right)$$

Update:

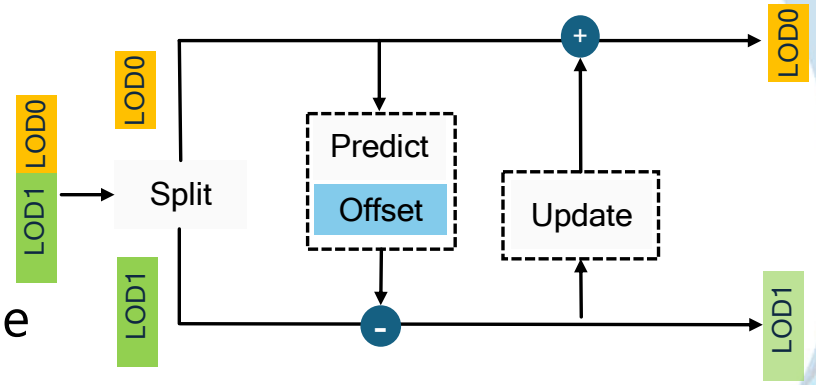
$$Signal(v) \leftarrow Signal(v) + \frac{1}{8}\sum_{w \in v^*} Signal(w)$$

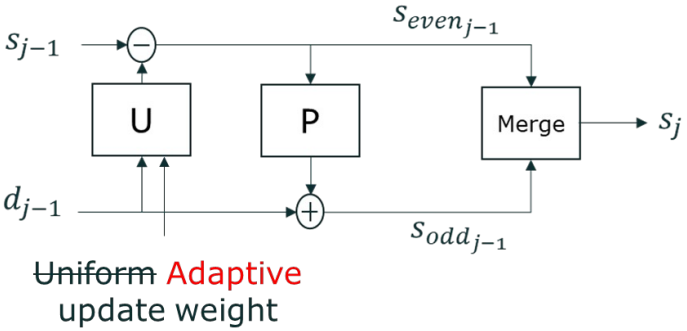


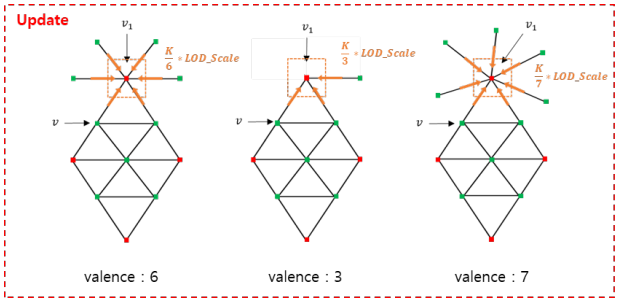
http://www.mat.unimi.it/users/naldi/lifting.pdf

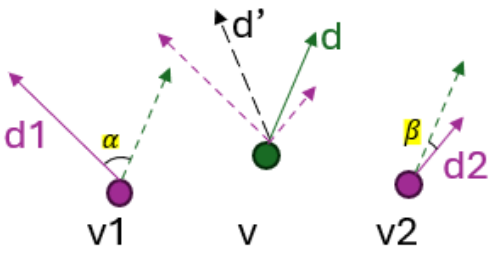
Linear lifting improvements

- Adding offset per LoD
- Using adaptive weights
  - Weights based on valence
  - Directional weights based on coherence



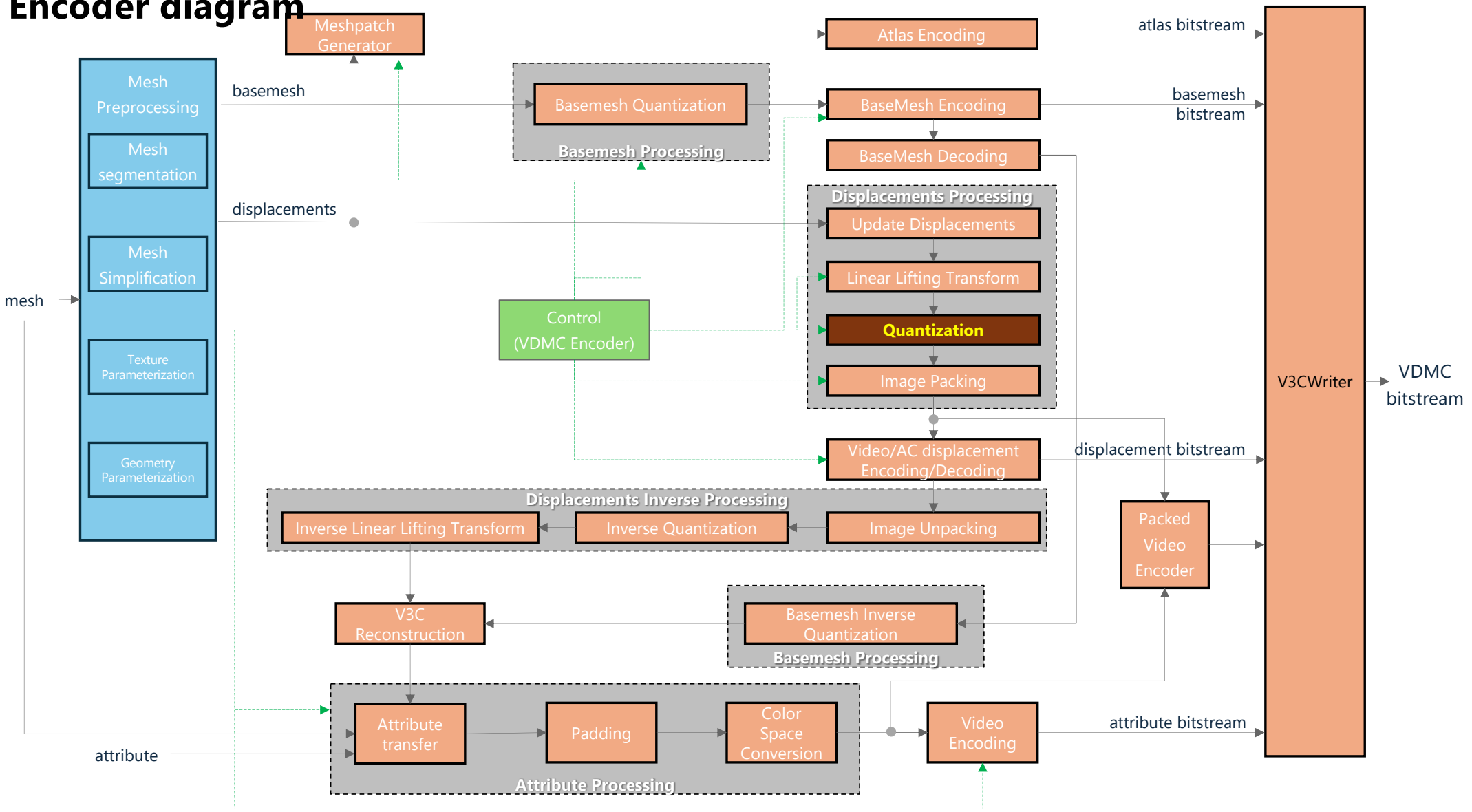






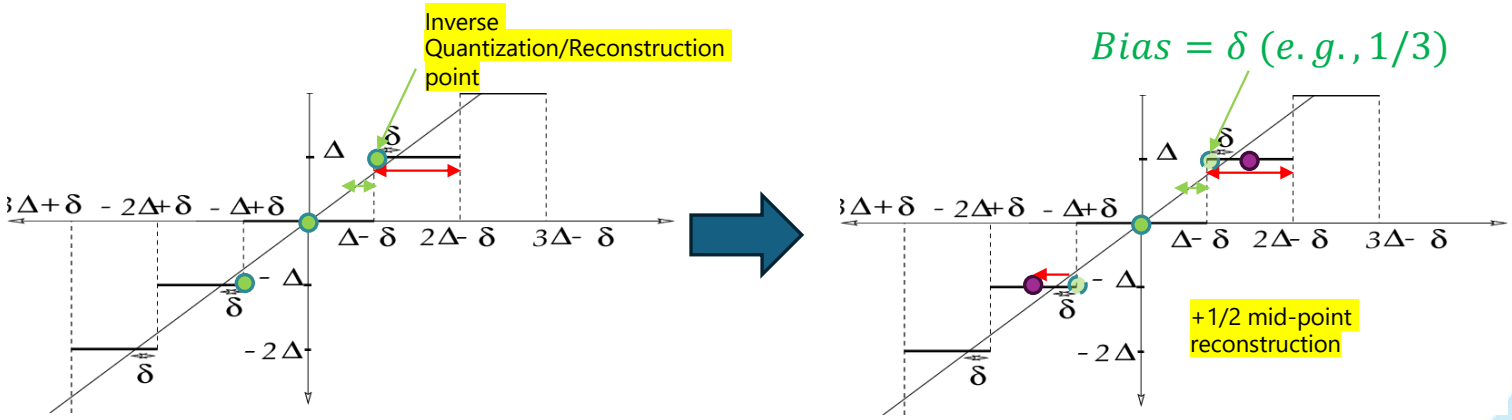
m66596	145 - OnLine	[V-DMC][EE 4.7] Report on Test 7.5 On lifting transform for displacements	Reetu Hooda, Geert Van der Auwera, Adarsh Krishnan Ramasubramonian, Anique Akhtar, Marta Karczewicz (Qualcomm)
m65389	144 - Hannover	[V-DMC] [New][EE4.7-related] Valence-based adaptive lifting update	Han-je Park, Dae-hyun Kim, Jong-Yeul Suh (LGE), Joo-Hyung Byeon, Min-tae Kim, Dong-gyu Sim (KWU)
m67644	146 - Rennes	[V-DMC][EE4.7-related][New] Improvements of lifting wavelet transform for displacements	Marta Milovanovic, Chao Cao (Ofinno)

Encoder diagram

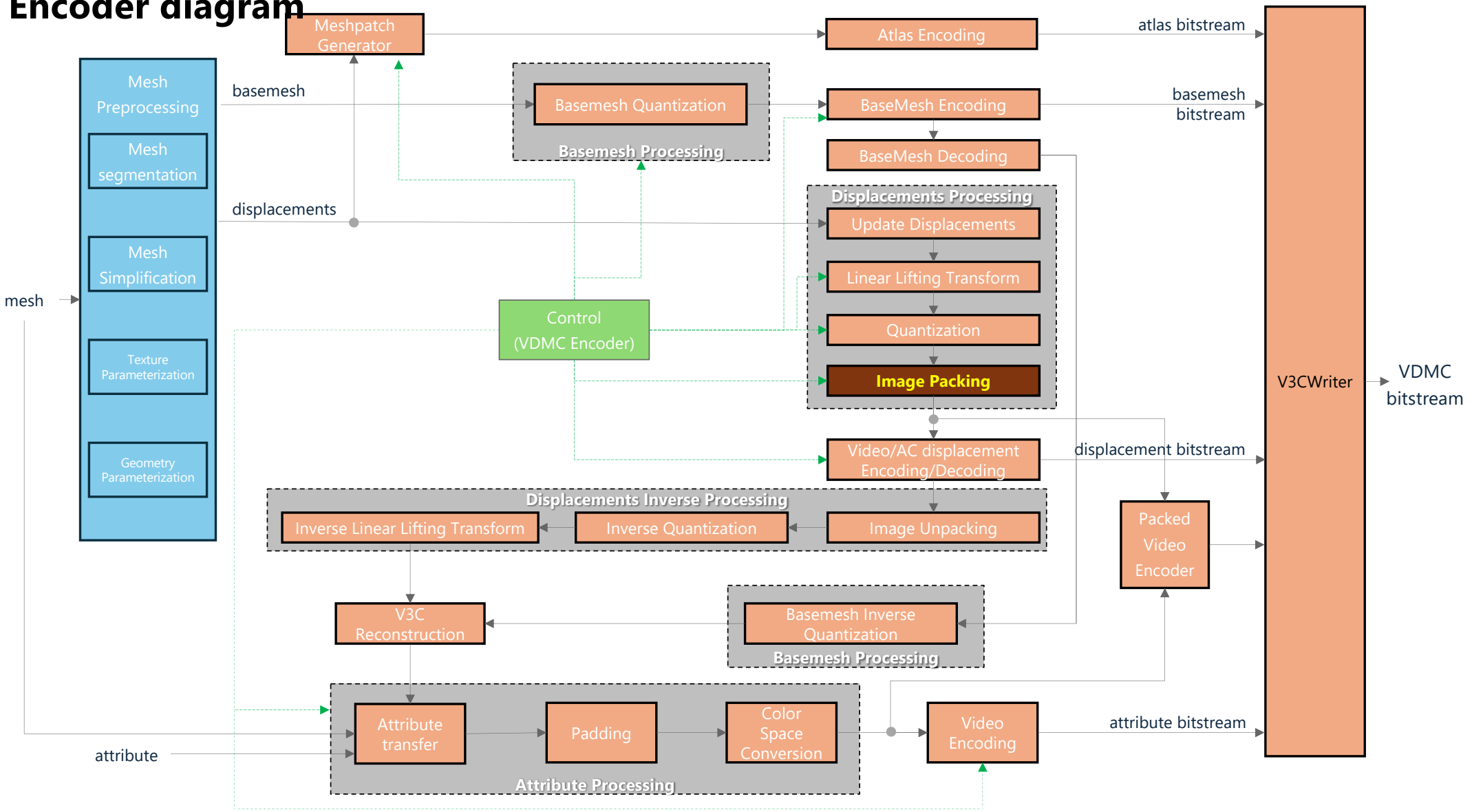


Transform coefficient quantization

- Different methods to signal the quantization steps
  - Direct mode: QP
  - Quantization Parameter: $2^{-\left(16+bitdepth_{offset}-bitdepth+\left(\frac{4-QP}{6}\right)\right)}$
  - Application of offsets to the quantization steps per LoD



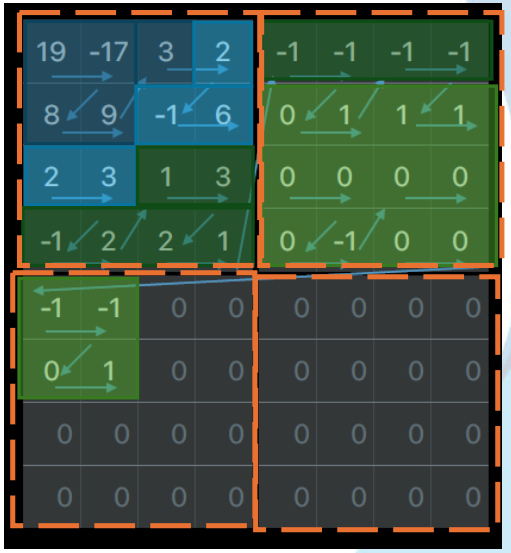
Encoder diagram

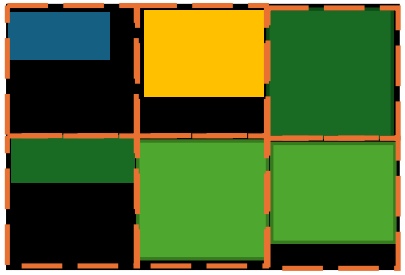


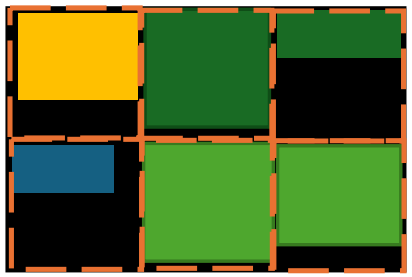
Displacement packing

- Video frame packing
  - Traverse coefficients from low to high

- Improvements: slice and rectangular packing
  - Separate LoDs in video slices
  - Rectangular LoD packing

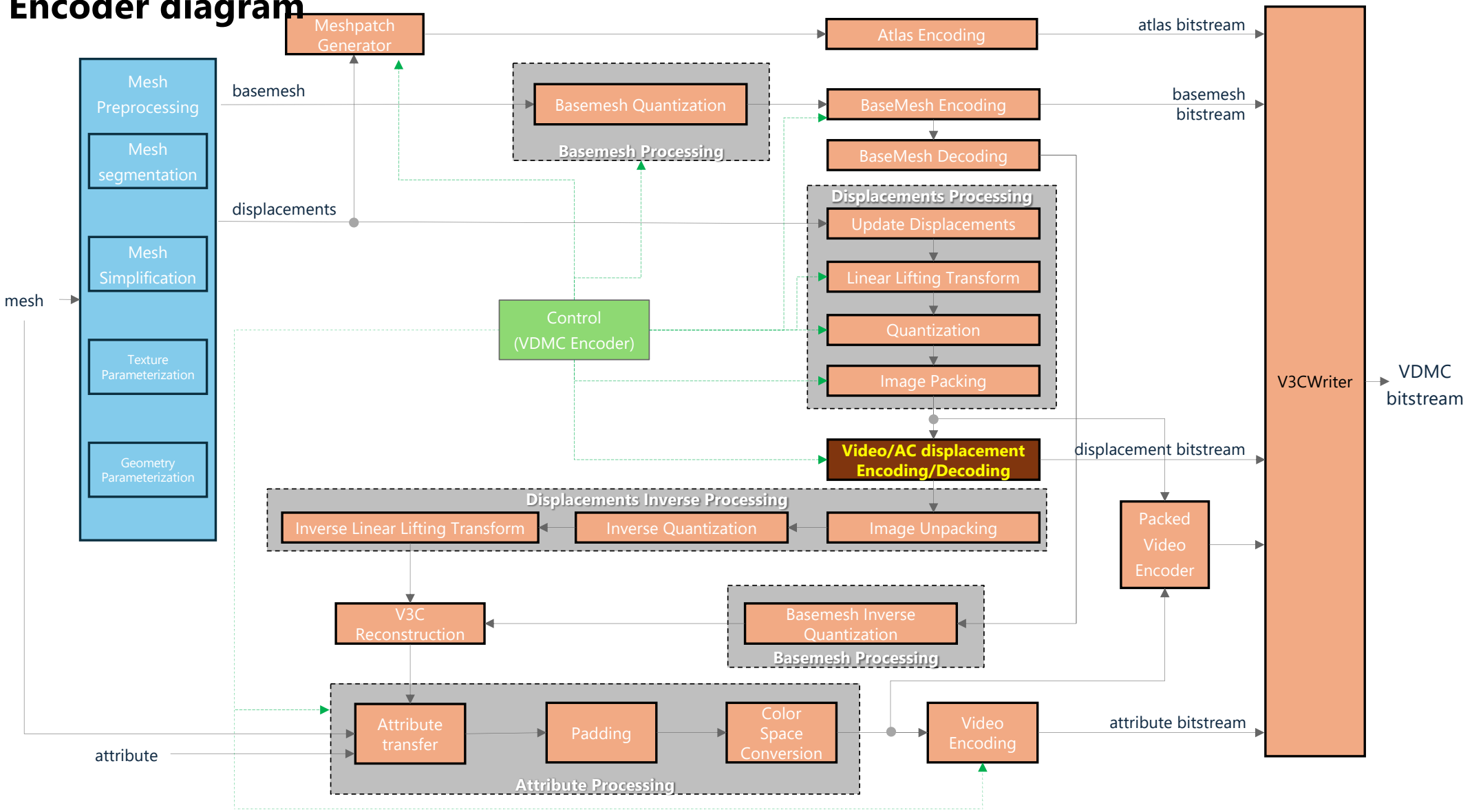






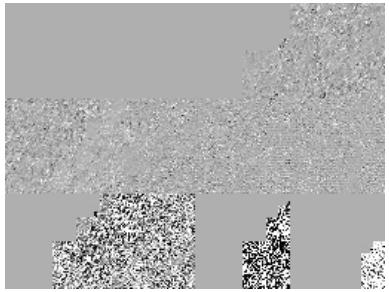
LoD-0 = $(x, y, SizeX, sizeY)_0$
LoD-1 = $(x, y, SizeX, sizeY)_1$
LoD-2 = $(x, y, SizeX, sizeY)_2$
LoD-3 = $(x, y, SizeX, sizeY)_3$

Encoder diagram



Displacement coding

- Geometry video coding



Video Encoding

Video codecs defined via profiles or SEI messages (using MP4RA)

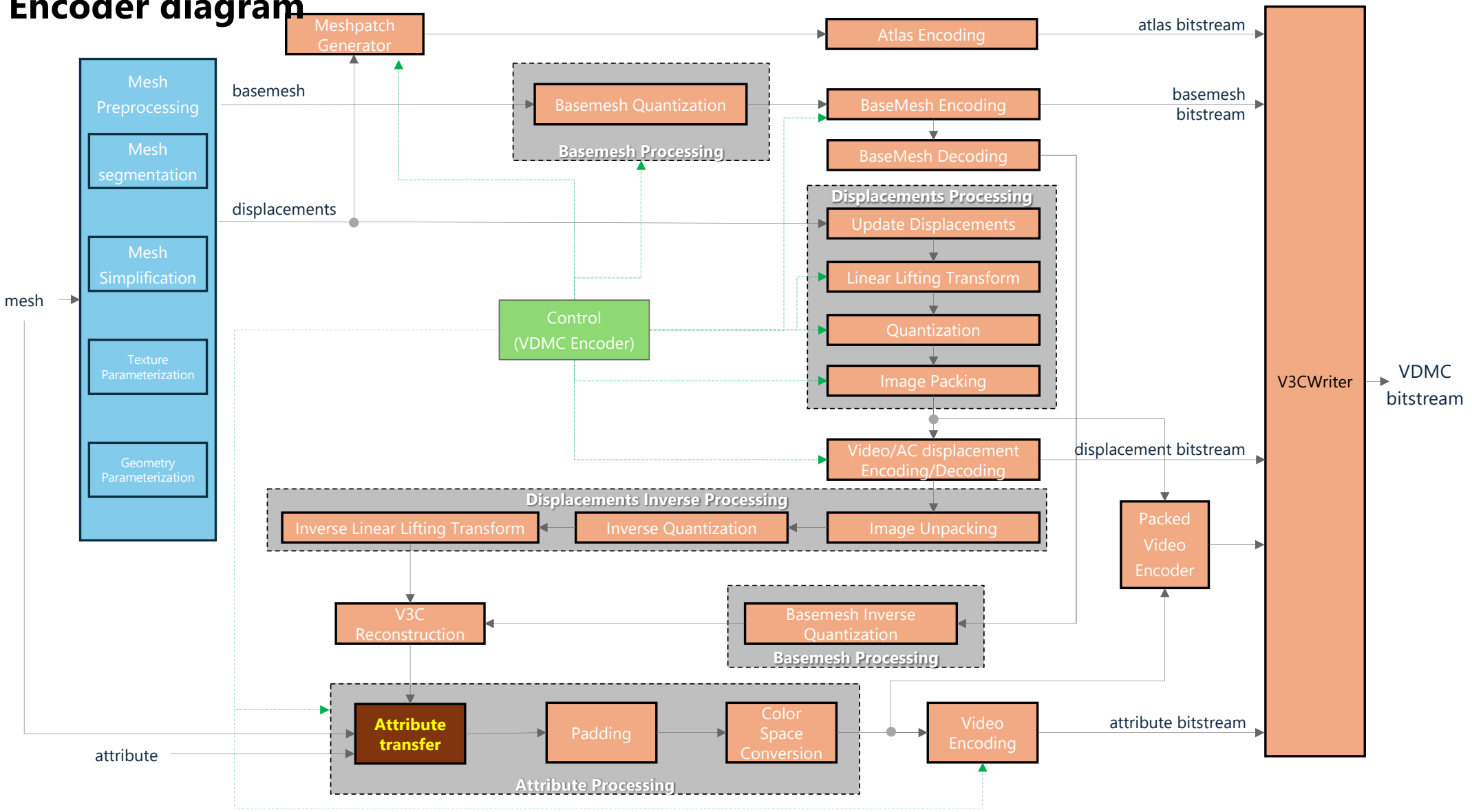
- AC displacement coding (Annex J)
  - Use NAL units like video solutions
  - Incorporate several techniques for efficient arithmetic coding of displacements
    - Block-based CABAC, fixed-point quantization, etc.

01011101010111 101010110100011..

AC Encoding

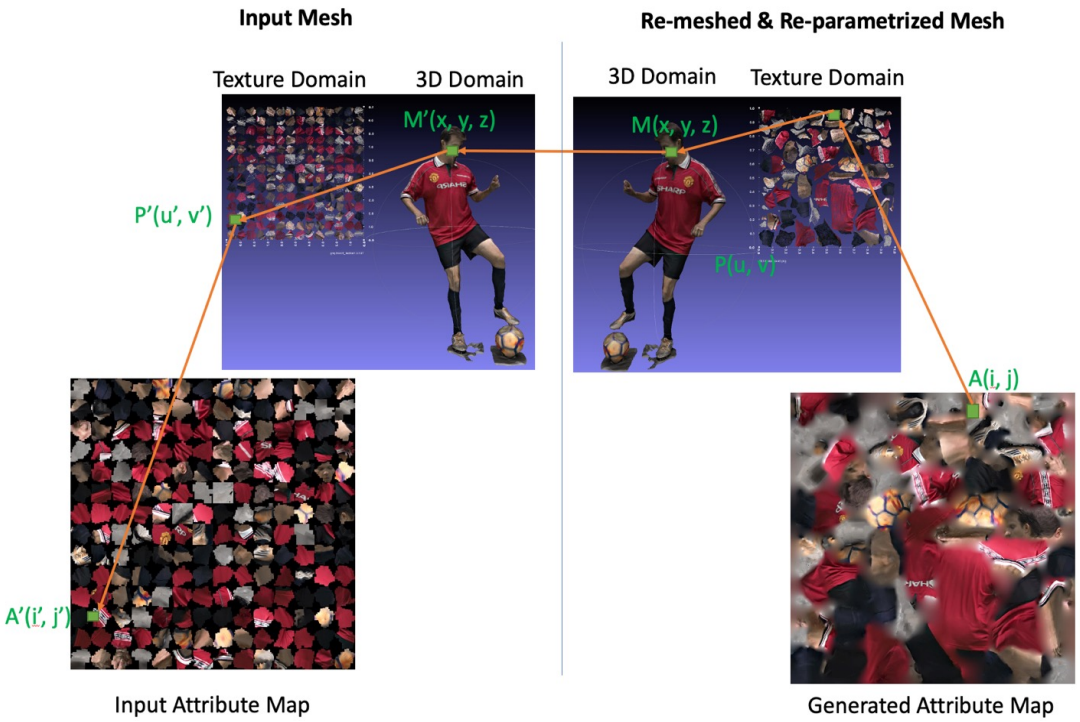
m60300	139 - OnLine	Arithmetic Coding of Displacements for Subdivision-based Mesh Compression	Chao Huang, Xiang Zhang, Xiaozhong Xu, Jun Tian, Shan Liu
m61065	140 - Mainz	[V-DMC][new] Block-Based Context-Adaptive Arithmetic Coding of Displacements	H. Nishimura, K. Kawamura, K. Kishimoto (KDDI)
m66302	145 - OnLine	[V-DMC][EE4.7 Test5-related] V-DMC Displacement Wavelet Coefficient Inter Prediction with Fixed-Point Quantization	Geert Van der Auwera, Adarsh Krishnan Ramasubramonian, Reetu Hooda, Anique Akhtar, Marta Karczewicz (Qualcomm),

Encoder diagram

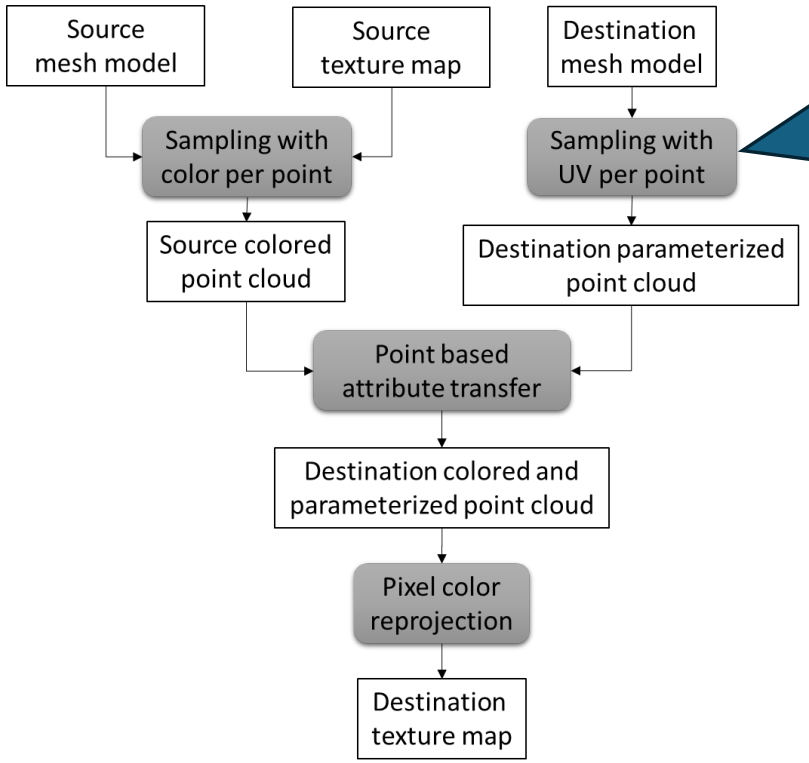


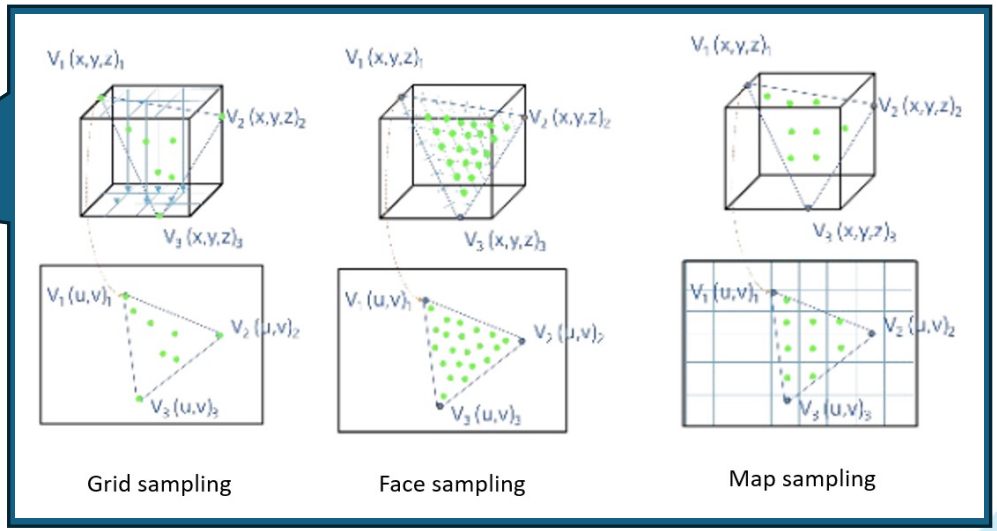
Texture transfer

- Optimizes mesh texture quality when geometry distortions are present

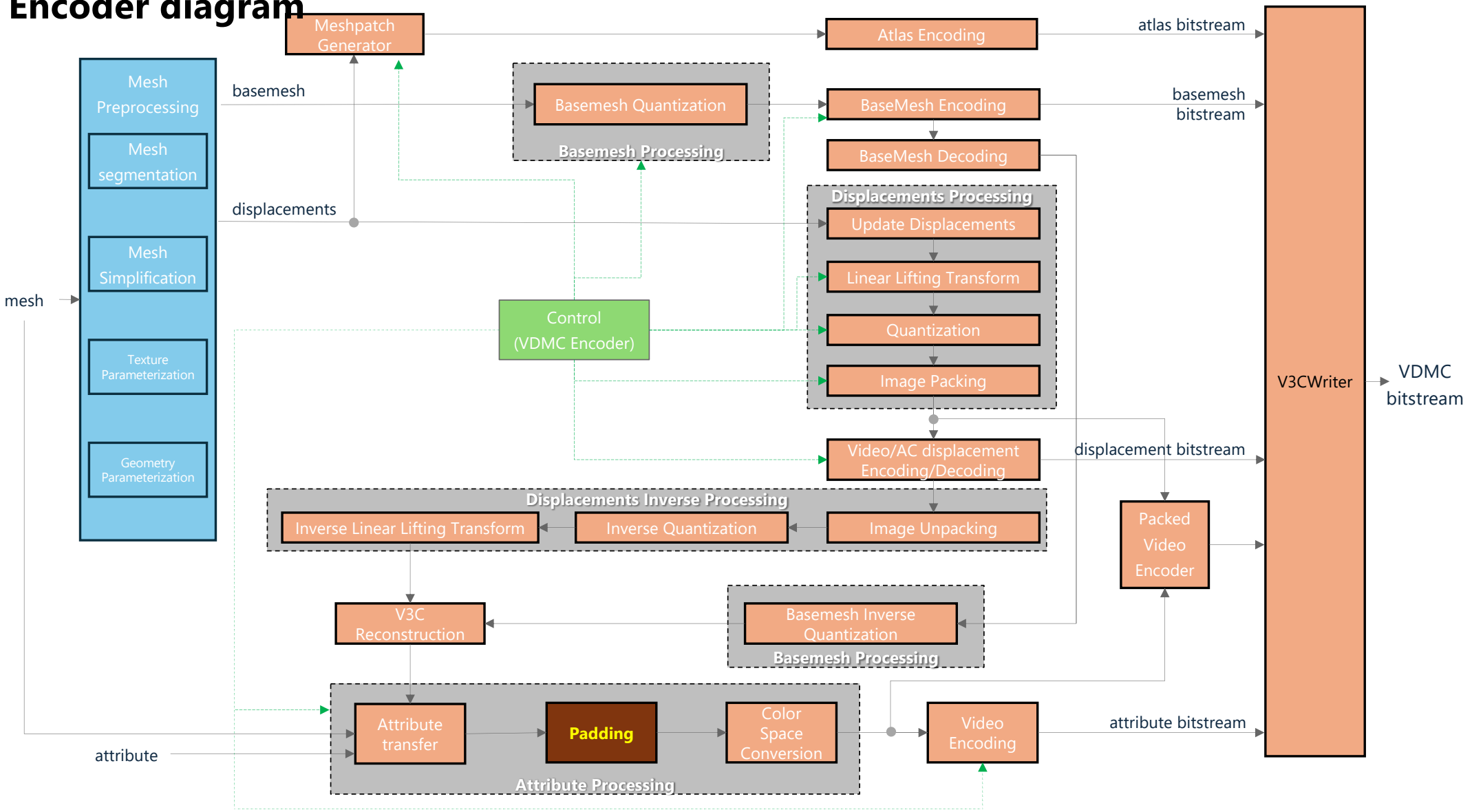


Texture transfer improvement



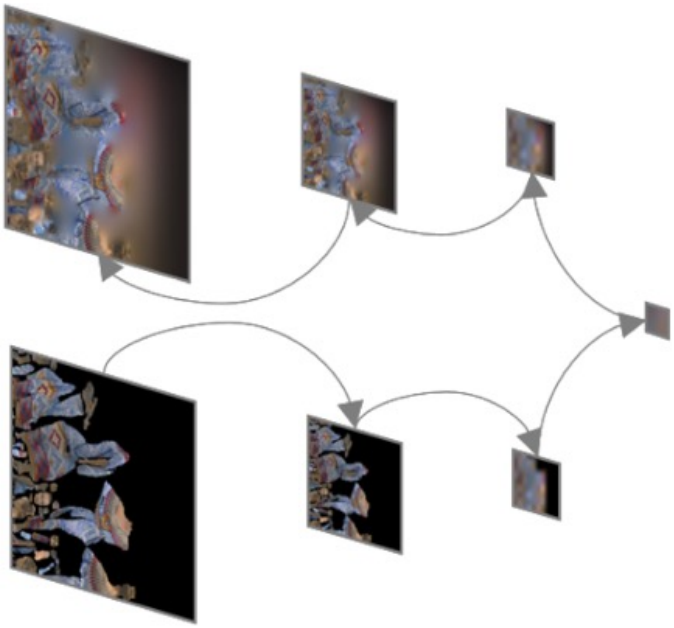


Encoder diagram

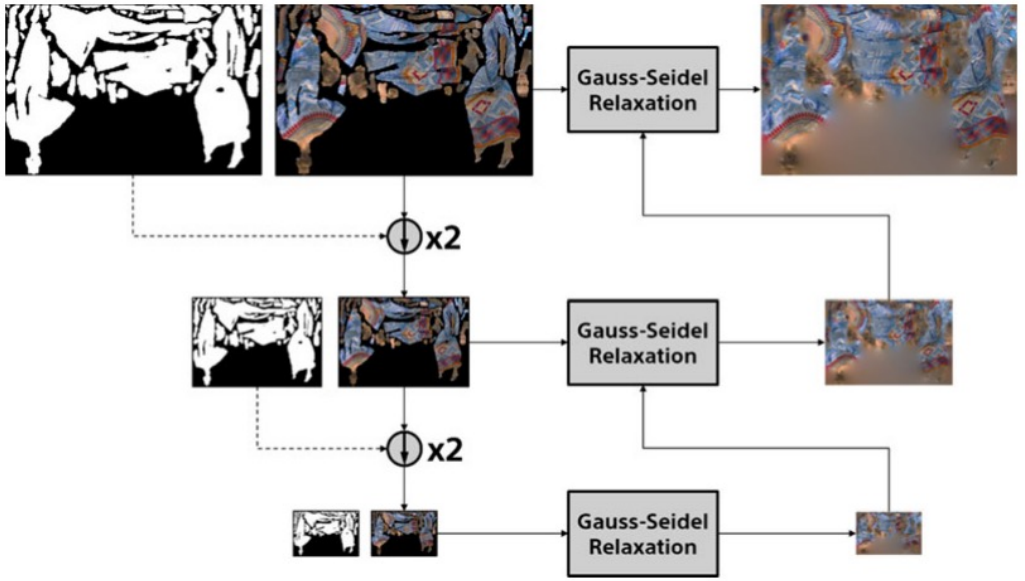


Texture Padding

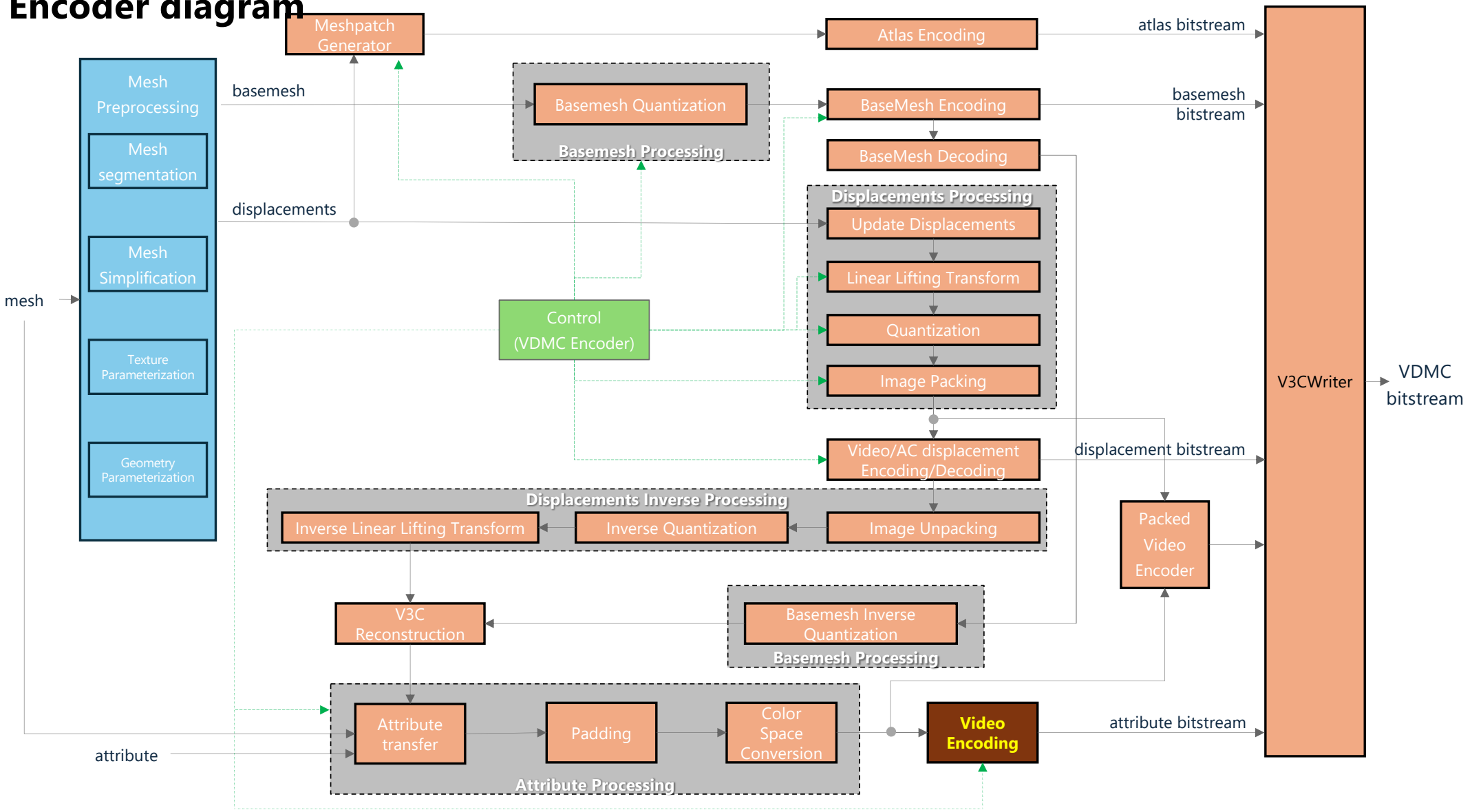
Smoothed Push-Pull



Harmonic Background Filling

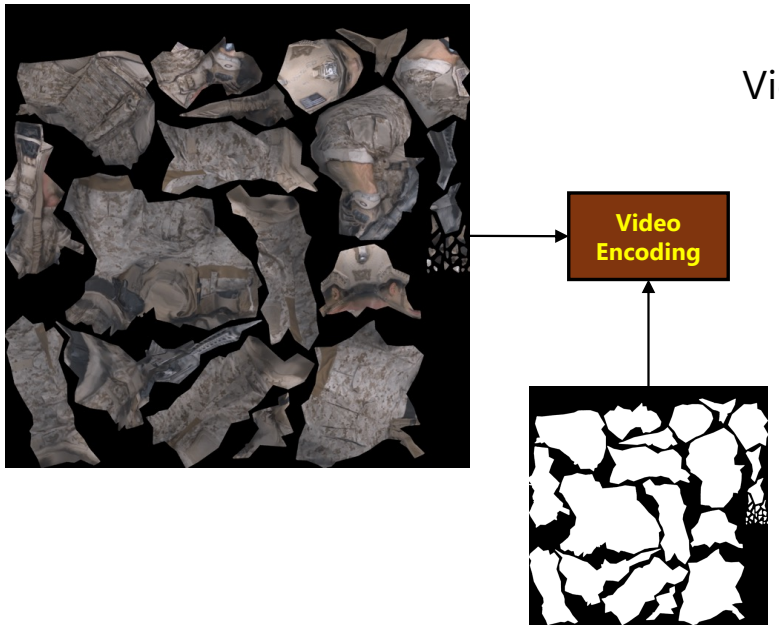


Encoder diagram



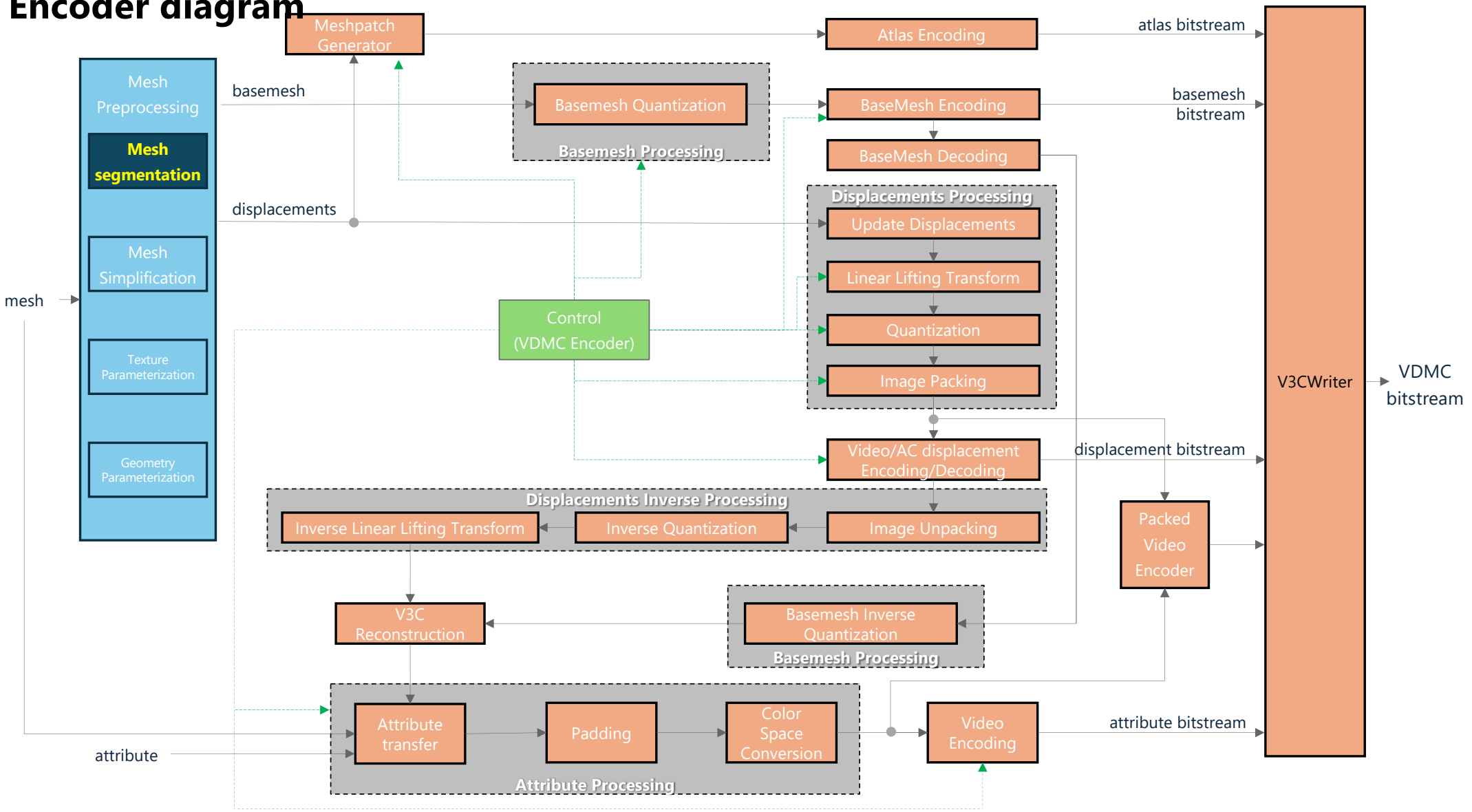
Attribute video coding

- Improve compression by masking the areas of the texture islands



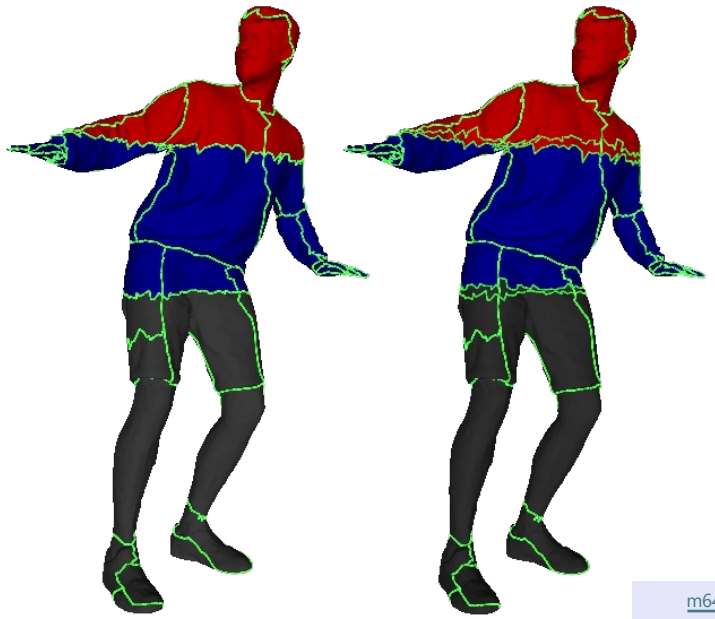
Video codecs defined via profiles or SEI messages (using MP4RA)

Encoder diagram



Mesh segmentation

- 3D space segmentation
  - Border overlap

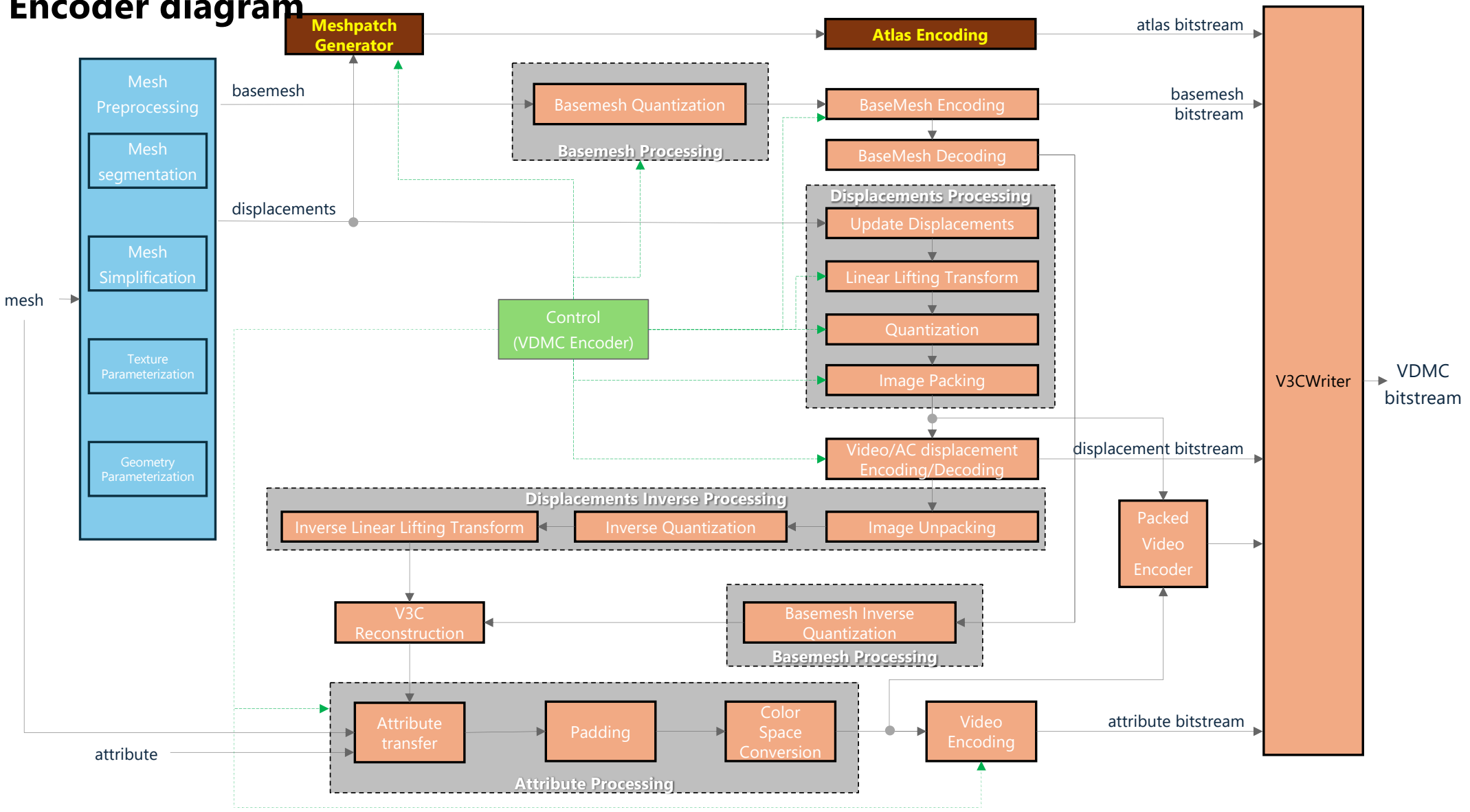


- Mesh density segmentation

- Basemesh segmentation

m64504	143 - Geneva	[V-DMC][EE4.11] NEW MESH SEGMENTATION BASED ON TRIANGLE AREA	Danillo Bracco Graziosi, Alexandre Zaghetto, Ali Tabatabai
m63343	142 - Antalya	[V-DMC][EE4.11 related][New] Correction Processing of Submesh Border Vertices	Koki Kishimoto, Kei Kawamura, Hitoshi Nishimura, Jianfeng Xu

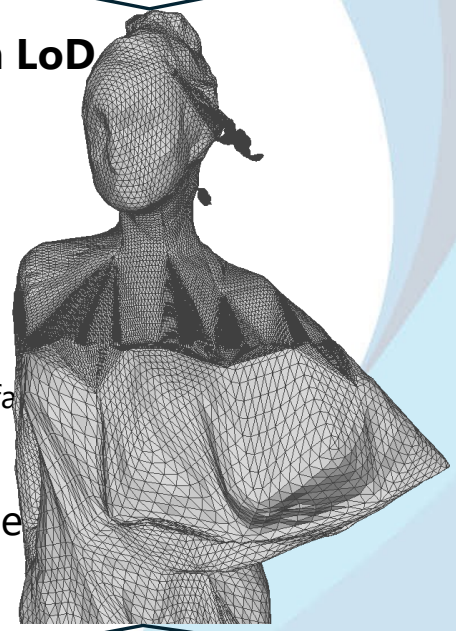
Encoder diagram



Meshpatch

```
[     5-1]meshpatchDataUnit(){
[     5-2]PduSubmeshId ..................................... = 0       ue(v)
[        ]...
[     6-7]PduParametersOverrideFlag ........................ = 1       u(1)
[     7-0]PduSubdivisionIterationCountOverrideFlag .......... = 1       u(1)
[     7-3]PduSubdivisionIterationCount ..................... = 4       u(3)
[        ]...
```

- Coding unit that carries information related to a **submesh** or a **submesh LoD**
  - Submesh ID
  - (LoD index)
  - Placement of displacement values in geometry video or subdisplacement ID
  - Number of displacements per LoD
  - Placement of texture in texture video
- The following parameters can be changed at submesh level
  - Subdivision
    - Example: increased subdivision for top submesh (to increase subdivision value for the fa
  - Quantization parameters
  - Transform parameters
- The additional parameters can be included, according to sequence-define
  - Quantization offsets
  - Transforms offsets
  - orthoAtlas parameters



```
[     94-3]meshpatchDataUnit(){
[     94-6]PduSubmeshId ..................................... = 2       ue(v)
[        ]...
[     96-5]PduParametersOverrideFlag ........................ = 0       u(1)
[        ]...
```

Atlas coding

- INTER, MERGE, and SKIP Meshpatch
  - Meshpatches that explore the temporal correlation between parameters to reduce the bit overhead

```
[     5-0]vdmcAtlasTileDataUnit(I_TILE){
[     5-1]PatchMode ..................................... = 0      ue(v)
[     5-1]meshpatchDataUnit(){
[     5-2]PduSubmeshId .................................. = 0      ue(v)
[        ]...
[    42-3]PatchMode ..................................... = 0      ue(v)
[    42-3]meshpatchDataUnit(){
[    42-6]PduSubmeshId .................................. = 1      ue(v)
[        ]...
[    78-5]PatchMode ..................................... = 0      ue(v)
[    78-5]meshpatchDataUnit(){
[    79-0]PduSubmeshId .................................. = 2      ue(v)
[        ]...
[   114-5]PatchMode ..................................... = 14     ue(v)
[   114-5]} //~vdmcAtlasTileDataUnit(I_TILE)
```

Frame #0

```
[     5-0]vdmcAtlasTileDataUnit(P_TILE){
[     5-3]PatchMode ..................................... = 2      ue(v)
[     5-3]interMeshpatchDataUnit(){
[     5-4]ImduRefPatchIndex ............................. = 0      se(v)
[        ]...
[    34-6]PatchMode ..................................... = 2      ue(v)
[    34-6]interMeshpatchDataUnit(){
[    34-7]ImduRefPatchIndex ............................. = 0      se(v)
[        ]...
[    62-4]} //~interMeshpatchDataUnit
[    62-7]PatchMode ..................................... = 2      ue(v)
[    62-7]interMeshpatchDataUnit(){
[    63-0]ImduRefPatchIndex ............................. = 0      se(v)
[        ]...
[    90-6]PatchMode ..................................... = 14     ue(v)
[    90-6]} //~vdmcAtlasTileDataUnit(P_TILE)
```

Frame #1

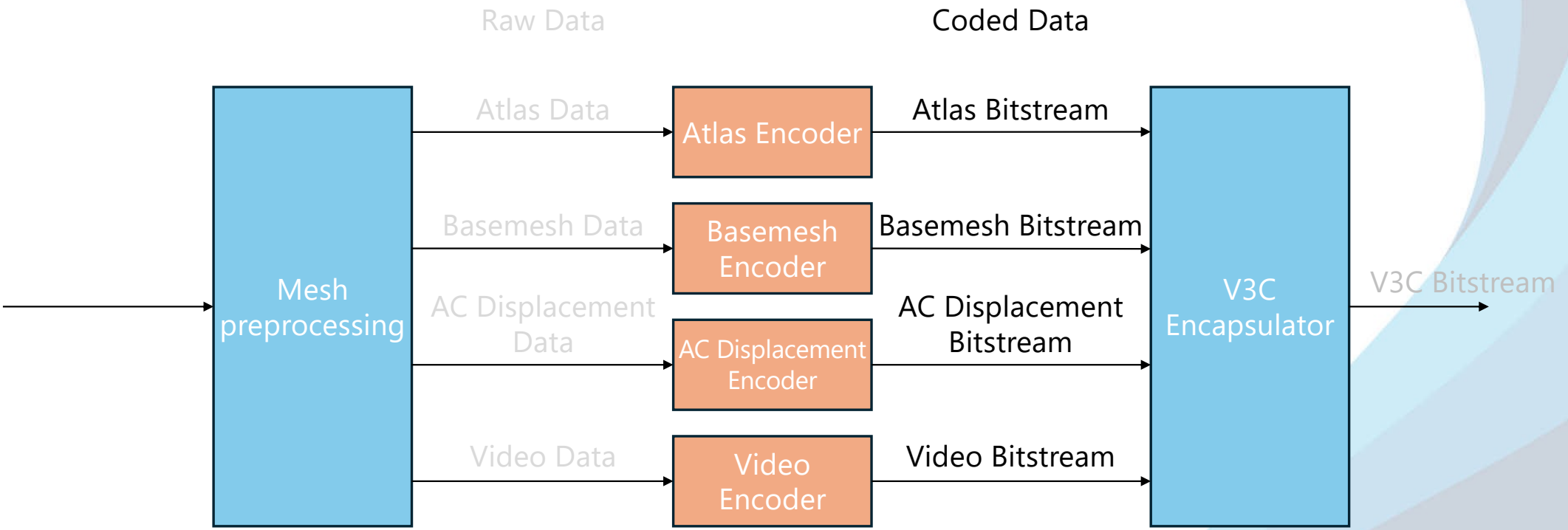
Agenda

- **Motivations for V-DMC and overall architecture** (Marius Preda – 15 minutes)
- **Core technologies and coding features** (Danillo Graziosi – 1 hour)
- **Syntax and semantics** (Lukasz Kondrad – 30 minutes)
- **Reconstruction techniques** (Lukasz Kondrad + Danillo Graziosi – 30 minutes)
- **Performance metrics and evaluation** (Wenjie Zou – 25 minutes)
- **Conclusions and future directions** (Marius Preda – 10 minutes)

Overview

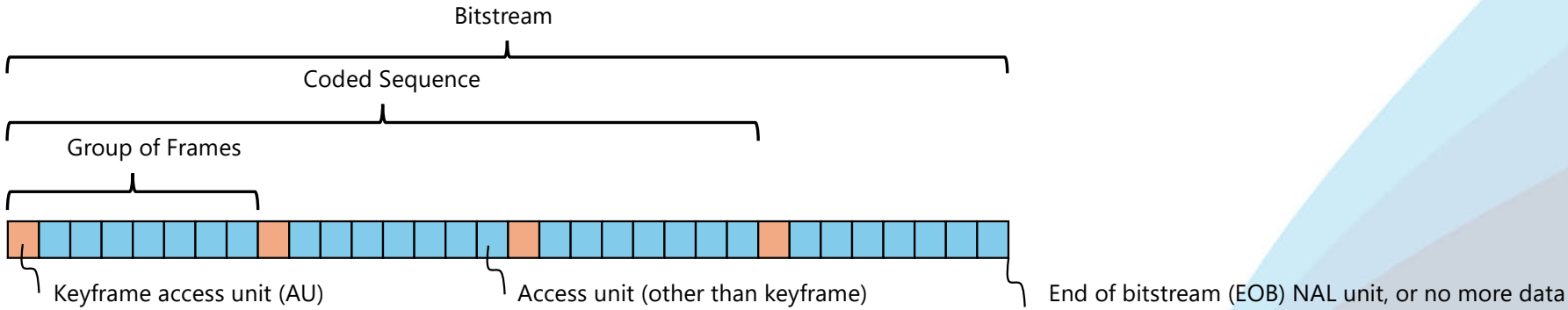
- V-DMC components sub-bitstream
  - atlas, basemesh, ac displacement HLS
  - profiles, tiers, levels
- V3C bitstream
  - V3C parameter set
  - profiles, tiers, levels

Encoder


Raw Data
Coded Data
Mesh preprocessing
Atlas Data
Atlas Encoder
Atlas Bitstream
Basemesh Data
Basemesh Encoder
Basemesh Bitstream
AC Displacement Data
AC Displacement Encoder
AC Displacement Bitstream
Video Data
Video Encoder
Video Bitstream
V3C Encapsulator
V3C Bitstream

Bitstream

- Bitstream may have one or more Coded Sequences
- Coded Sequence is composed from one or more GOFs
- Groups of Frames (GOF) is composed of access units (AUs), and GOF typically start with a keyframe
- access unit (AU) consists of coding layer NAL units for one timestamp and associated non-coding layer NAL units



Network Abstraction Layer (NAL)

- Acts as a bridge between the coding layer and the transport layer
- NAL unit consists of a
  - NAL unit header
  - NAL unit payload
- NAL unit types categorized into
  - Non-Coding Layer units
  - Coding Layer units

Non-coding layer NAL units

- Non-coding layer contains metadata:
  - essential metadata required to decode the actual content
    - Sequence Parameter Set - defines the overall structure and capabilities of bitstream
    - Picture/Frame Parameter Set - defines frame-specific decoding instructions
  - non-essential metadata information to assist in decoding, display, or post-processing of the encoded data
    - SEI (Supplemental Enhancement Information)
    - Filler data
    - Bitstream Information (access unit delimiter, end of sequence, end of bitstream)

Parameter Sets

	Atlas non-ACL	Basemesh non-BMCL	AC displacement non-DCL	Video non-VCL
Sequence paraments	ASPS	BMSPS	DSPS	SPS
	• Dimension (geometry and attribute atlas) • Bit depth information (2D/3D) • Max number of reference frames • Patch packing	• Profile and level • Intra/inter codec information • 3D bit depth conversion information (e.g., 10-bit, 12-bit) • Attribute information • Max number of reference frames • Frame rate (VUI)	• Bit depth (e.g., 10-bit, 12-bit) • Lod count • Sequence level quantization • Profile and level • Max number of reference frames	• Bit depth (e.g., 8-bit, 10-bit) • Chroma format (e.g., 4:2:0, 4:2:2) • Frame rate (via timing info) • Profile and level (e.g., Main, Main10) • Max number of reference frames • Conformance window (cropping info)
Frame parameters	AFPS	BMFPS	DFPS	PPS
	• Atlas Geometry and Attribute Tile mapping • Submesh information • Patch type allowed (e.g. raw, eom) • Raw patch information	• Submesh information	• Subdisplacement information • Frame level quantization	• Entropy coding mode (e.g., CABAC) • Deblocking filter settings • Slice group mapping • Tile and wavefront parallel processing (WPP) settings • Reference index settings

Supplemental Enhancement Information

	Atlas non-ACL	Basemesh non-BMCL	AC displacement non-DCL	Video Non-VCL
essential	• Component codec mapping • Buffering period • Frame timing • …	• Component codec mapping • Buffering period • Frame timing	• Buffering period • Frame timing • ..	• Buffering period • Frame timing • …
non-essential SEI	• Zippering • Submesh SOI relationship indication • Submesh distortion indication • LoD extraction information • Tile submesh mapping • Attribute extraction information • …	• Attribute transformation parameters • ..		

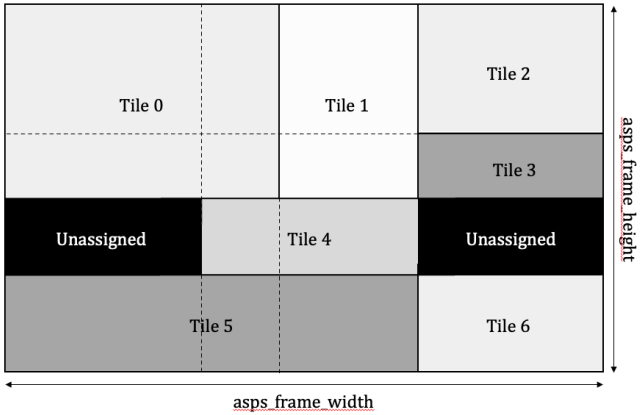
Coding Layer NAL units

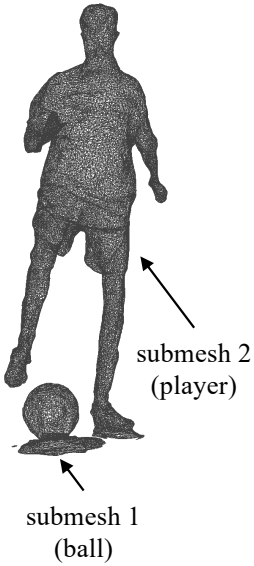
- Coding Layer NAL unit contains a compress unit of data that can be used for transmission and for independent parsing on the decoder
- Coding Layer NAL unit type can be
  - keyframe – e.g., IDR Instantaneous Decoder Refresh
  - non-keyframe – e.g., contains prediction information from one or more previous frames in sequence

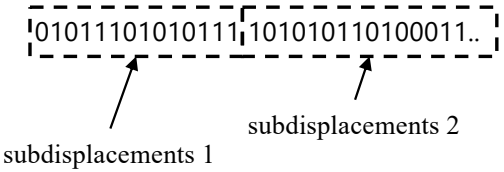
	Atlas (ACL)	Basemesh (BMCL)	AC displacement (DCL)	Video (VCL)
keyframe	NAL_IDR_N_LP	NAL_IDR_N_LP	NAL_IDR_N_LP	NAL_IDR_N_LP
non-keyframe	NAL_TRAIL_N/R NAL_SKIP_N/R	NAL_TRAIL_N/R NAL_SKIP_N/R	NAL_TRAIL_N/R	NAL_TRAIL_N/R NAL_SKIP_N/R

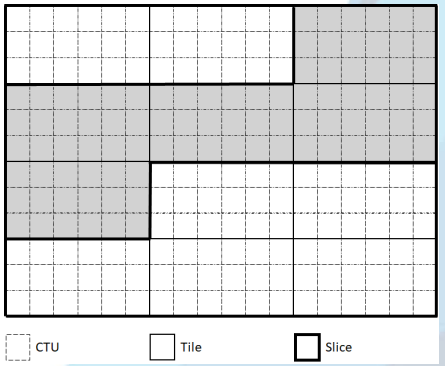
Atlas, basemesh, AC displacement, Video

	Atlas (ACL)	Basemesh (BMCL)	AC displacement (DCL)	Video (VCL)
	tiles	submesh	subdisplacements	slices









Atlas Tiles

- Atlas Tiles syntax structure has a header and a payload (data unit)
- Header contains metadata about tile:
  - linkage to parameters sets
  - type (I_TILE, P_TILE, SKIP_TILE, I_TILE_ATTR)
  - tile ID
  - frame order count
  - Information about the reference frames
- Payload contains patch information
  - Geometry tile: meshpatch_data_unit, skip_meshpatch_data_unit, merge_meshpatch_data_unit, inter_meshpatch_data_uni
  - Attribute tile: meshpatch_data_unit

Basemesh Submeshes

- Basemesh Submeshe syntax structure has a header and a payload (data unit)
- Header contains metadata about coded submesh:
  - linkage to parameters sets
  - type (I_SUBMESH, P_SUBMESH, SKIP_SUBMESH)
  - submesh ID
  - frame order count
  - information about the reference frames
- Payload contains encoded base mesh data, e.g.,
  - I_SUBMESH: bytes from the static mesh encoder as defined in annex I
  - P_SUBMESH: bytes from motion encoder as defined in annex L

AC Displacement subdisplacements

- AC displacement subdisplacement syntax structure has a header and a payload (data unit)
- Header contains metadata about coded AC displacement:
  - linkage to parameters sets
  - type (I_DISPLACEMENT , P_DISPLACEMENT)
  - displacement ID
  - frame order count
  - information about the reference frames
  - Information about quantization
- Payload contains arithmetically coded data

Profile, tier, and levels (PTL)

- PTL specify restrictions on the bitstreams and hence limits on the capabilities needed to decode the bitstreams.
- PTL may be used to indicate interoperability points between individual decoder implementations.
- Profile specifies a subset of features that should be supported by the decoder
- Level of a tier specifies a set of limits on the values that may be taken by the syntax elements

Atlas PTL

- Signal in V3C Parameter Set
- Atlas bitstream defines two V-DMC profiles
  - Toolset 0 - defined to have simple parsing of atlas bitstream with only intra patches allowed and also only one tile for attribute
  - Toolset 1 -defined to enable inter and merge mesh patches which would provide better compression of atlas bitstream
- Atlas has distinction for levels related to geometry and attribute

Geometry Level	Max submeshes per frame	Max CAB size in 1 000 bits					
1.0	1	16	1	16 384	16	240	30
2.0	1	16	1	65 536	16	240	30
2.1	2	32	2	524 288	32	480	60
2.2	4	64	2	1 048 576	64	960	120

Attribute Level	Max attribute atlas size
1.0	2 228 224
2.0	8 912 896
3.0	35 651 584

Basemesh PTL - Profile

- PTL is signal in Basemesh Sequence Parameter Set
- Basemesh bitstream defines two profiles
  - BMS Main profile specifies the basemesh bitstream with technologies defined in ISO/IEC 23090-29, i.e. the intra mesh coding (Annex I) and inter motion coding (Annex L) are used
  - MP4RA profile is defined to allow the use of any codec for intra and inter coding, identified through a component codec mapping SEI message, or through out of band means.

Basemesh PTL – Tier and Level

- Basemesh bitstream defines two tiers
  - Tier 0 aims for consumer devices and has three levels 1.0, 2.0, 3.0
  - Tier 1 aims for professional applications with very high quality meshes and we define levels 1.0 and 2.0.

Levels for Tier 0

Level	Max basemesh bit rate 1000 bits /s	Max submeshes rate submeshes /s	Max basemesh vertices rate 1000 vertices/s	Max vertices per basemesh frame	Max CBMB size in 1000 bits	Max # submeshes per basemesh frame	Max # basemesh attribute	Max # basemesh attribute dimension
1.0	3 840	30	120	4 000	3 840	1	1	2
2.0	7 680	120	360	12 000	7 680	4	2	3
3.0	15 360	480	720	24 000	15 360	16	3	3

Levels for Tier 1

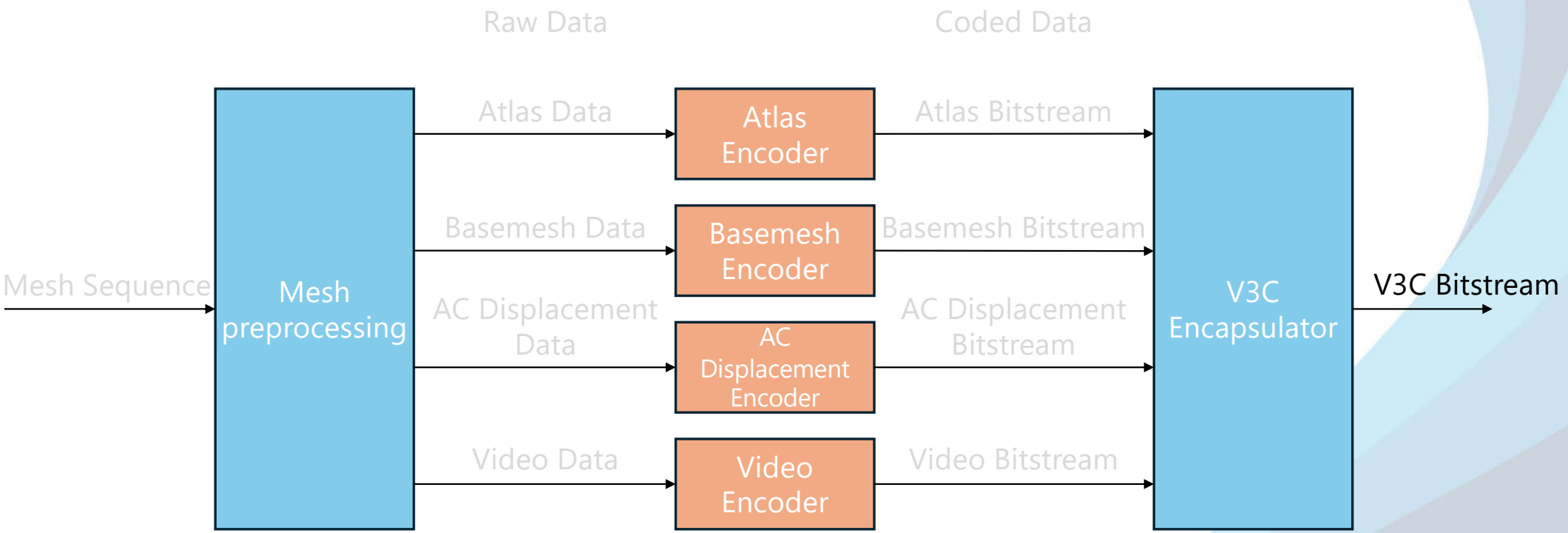
Level	Max basemesh bit rate in 1000 bits/s	Max submeshes rate submeshes/s	Max basemesh vertices rate 1000 vertices/s	Max vertices per basemesh frame	Max CBMB size in 1 000 bits	Max submeshes per basemesh frame	Max basemesh attribute	Max basemesh attribute dimension
1.0	18 000	480	3840	128 000	18 000	16	1	2
2.0	38 000	480	3840	128 000	38 000	16	2	3

AC Displacement PTL

- PTL is signal in Displacement Sequence Parameter Set
- AC displacement defined one profile. The only limitation is that one layer shall be present in the bitstream
- AC Displacement defines level only for tier 0 and the levels are aligned with atlas levels

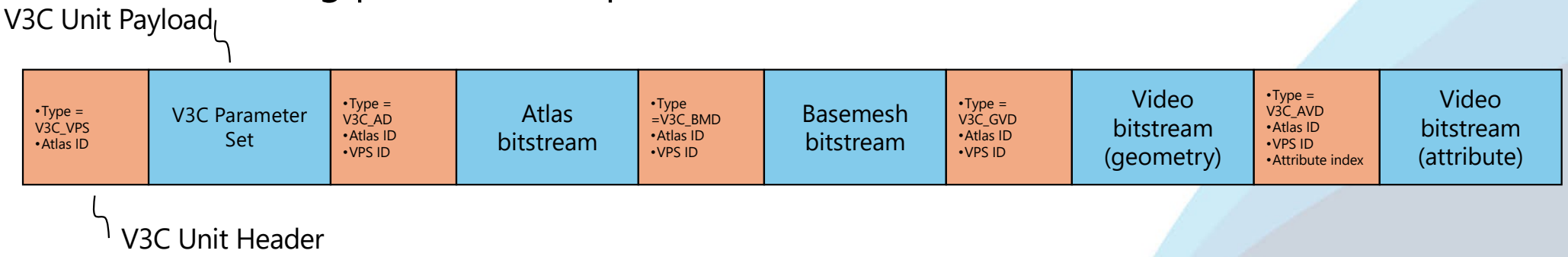
Level	Max displacements samples per displacement frame	Max CDB size in 1 000 000 bits	Max subdisplacements per displacement frame	Max displacement bit rate in 1 000 000 bits/s	Max subdisplacements rate subdisplacements/s	Max displacements rate vertices/s
1.0	16 384	1.5	1	1.5	30	491 520
2.0	65 536	2.5	1	2.5	30	1 966 080
2.1	524 288	4	2	4	60	15 728 640
2.2	1 048 576	7	4	7	120	31 457 280

Encoder


Raw Data
Coded Data
Mesh Sequence
Mesh preprocessing
Atlas Data
Basemesh Data
AC Displacement Data
Video Data
Atlas Encoder
Basemesh Encoder
AC Displacement Encoder
Video Encoder
Atlas Bitstream
Basemesh Bitstream
AC Displacement Bitstream
Video Bitstream
V3C Encapsulator
V3C Bitstream

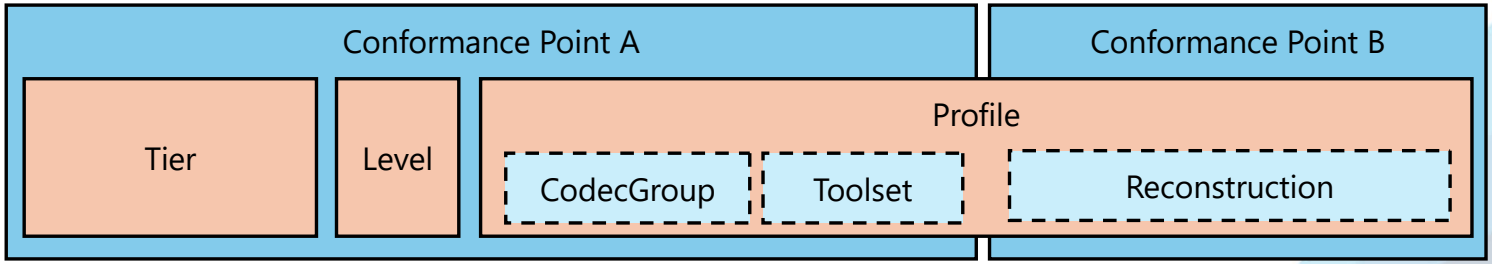
V3C bitstream

- Composed of V3C units, that have V3C unit header and V3C unit payload
- Provides linkage between the V3C Parameter Set and V-DMC components
- Storage, providing single bitstream to the transport layer for
  - file format encapsulation
  - streaming protocol encapsulation



Profiles of V3C/V-DMC

- Flexible structure identifying two distinct conformance points
  - Point A – covers decoded V-DMC components
  - Point B – covers the V-DMC reconstruction process
- Profile is split into three components
  - Toolset – restriction on the atlas plus allowed type of video components
  - CodecGroup – decoding specifications and their profiles for non-atlas components
  - Reconstruction – recommended tools to achieve conformance in terms of 3D reconstruction.



CodecGroup

- Split into video and non-video components

CodecGroup	PtlVideoCodecGroupIdc	4CC code
AVC Progressive High	0	'avc3'
HEVC Main 10	1	'hev1'
HEVC Main 4:4:4 10	2	'hev1'
VVC Main 10	3	'vvi1'
HEVC Main	4	'hev1'
Reserved	5..14	–
MP4RA	15	provided by external means or may be determined by component codec mapping SEI message

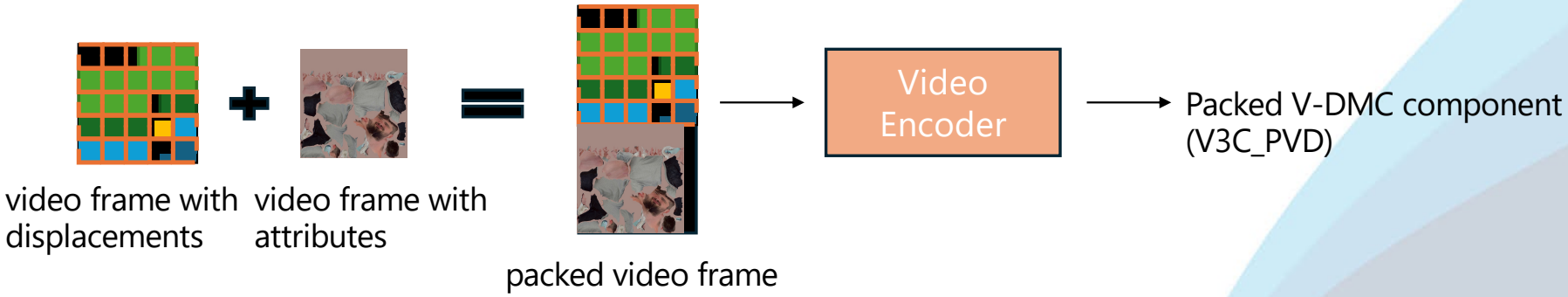
NonVideo CodecGroup	PtlNonVideoCodecGroupIdc	4CC code (V3C_BMD)	4CC code (V3C_ADD)
BMS Main	0	'bmsf'	–
BMS Main, DAC Main	1	'bmsf'	'dacf'
Reserved	2..6	–	–
MP4RA	7	provided by component codec mapping SEI message	

Reconstruction

Reconstruction Operation	Rec0	Rec1	Rec2	Rec3	Rec Unconstrained
chroma format conversion resolution upsampling frame rate conversion	ISO/IEC 23090-5, Annex B	ISO/IEC 23090-5, Annex B	ISO/IEC 23090-5, Annex B	ISO/IEC 23090-5, Annex B	unconstrained
texture coordinate derivation	OrthoAtlas	OrthoAtlas	unsupported	unsupported	unconstrained
subdivision	Midpoint	Midpoint, Loop, Normal, Pythagorean	Midpoint	Midpoint, Loop, Normal, Pythagorean	unconstrained
inverse image packing, quantization, transform normal, tangent, bitangent vertex coordinates reconstruction texture coordinate adaptation	ISO/IEC 23090-29, Clause 11	ISO/IEC 23090-29, Clause 11	ISO/IEC 23090-29, Clause 11	ISO/IEC 23090-29, Clause 11	unconstrained

Packed Video Component

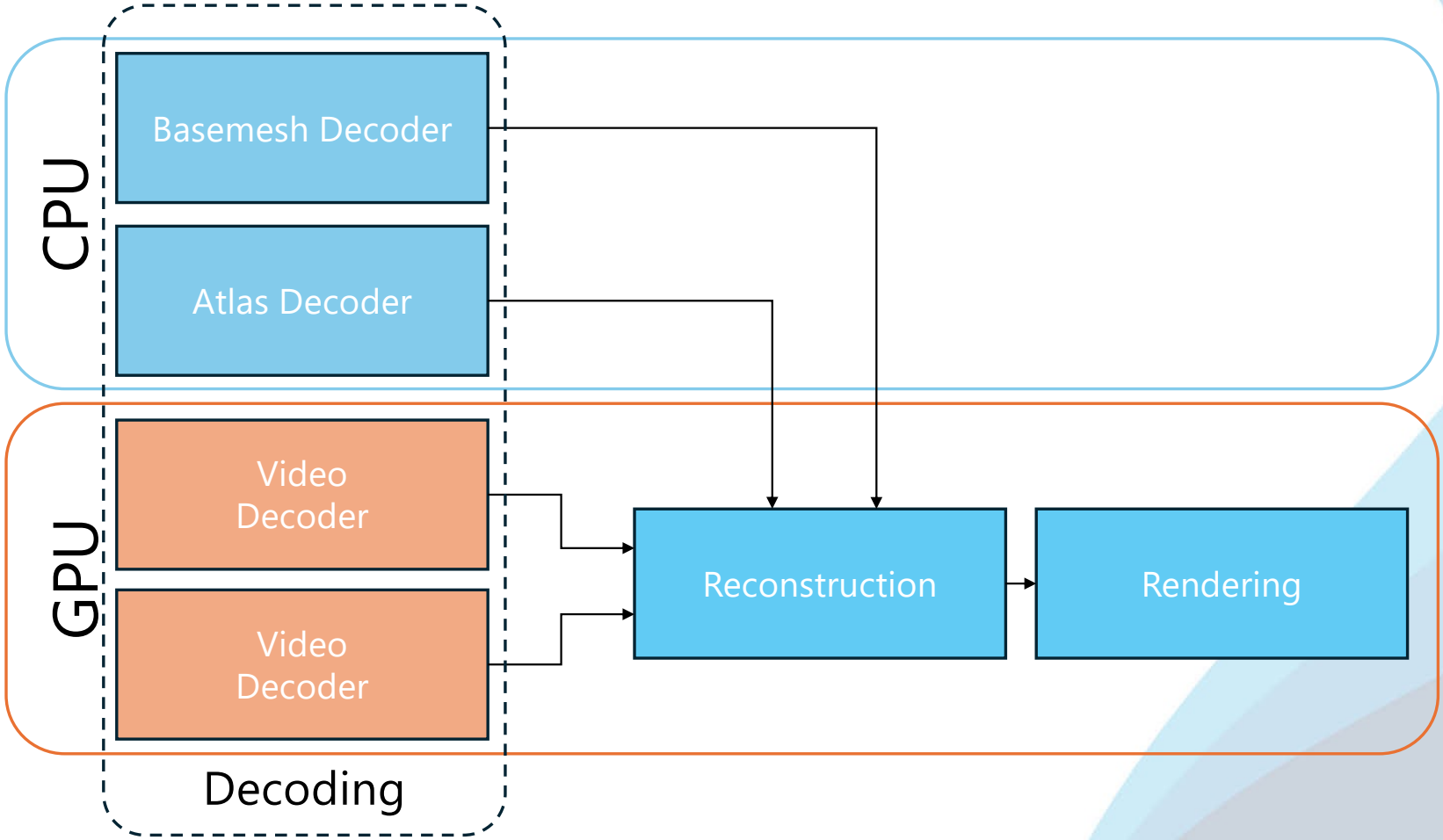
- V3C defines a packed V3C video component that allows to combine geometry and attribute video frame.
- Targeting platforms where the ability to run multiple parallel video decoding instances is limited
- Information on how to interpret the packed video component is provided in VPS



Agenda

- **Motivations for V-DMC and overall architecture** (Marius Preda – 15 minutes)
- **Core technologies and coding features** (Danillo Graziosi – 1 hour)
- **Syntax and semantics** (Lukasz Kondrad – 30 minutes)
- **Reconstruction techniques** (Lukasz Kondrad + Danillo Graziosi – 30 minutes)
- **Performance metrics and evaluation** (Wenjie Zou – 25 minutes)
- **Conclusions and future directions** (Marius Preda – 10 minutes)

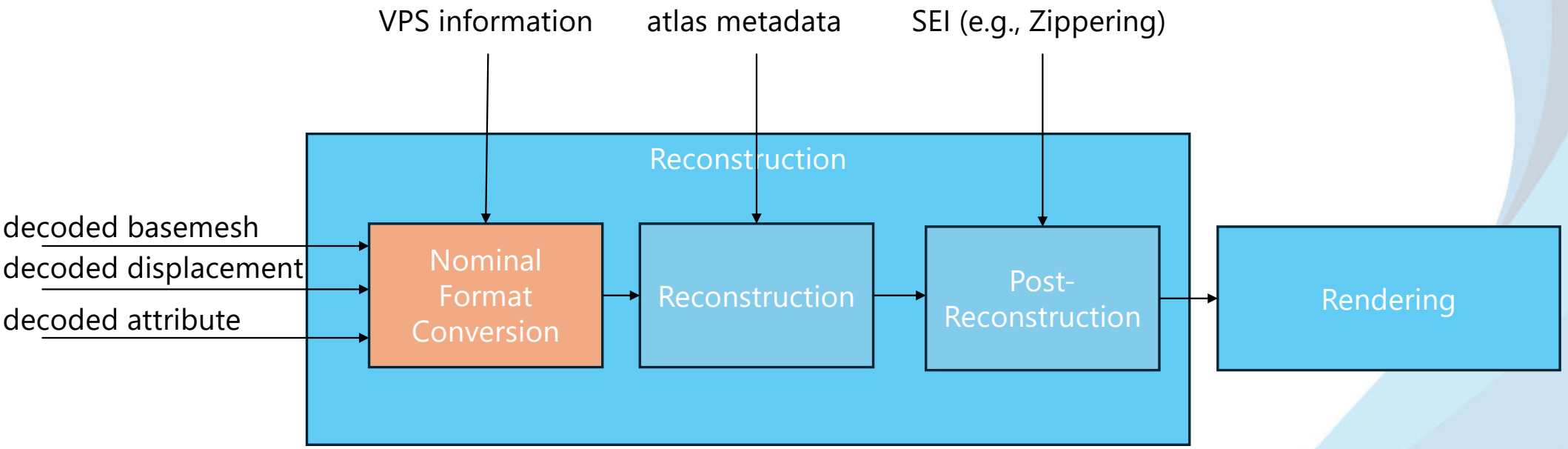
Decoder Pipeline



Sub-bitstream decoding

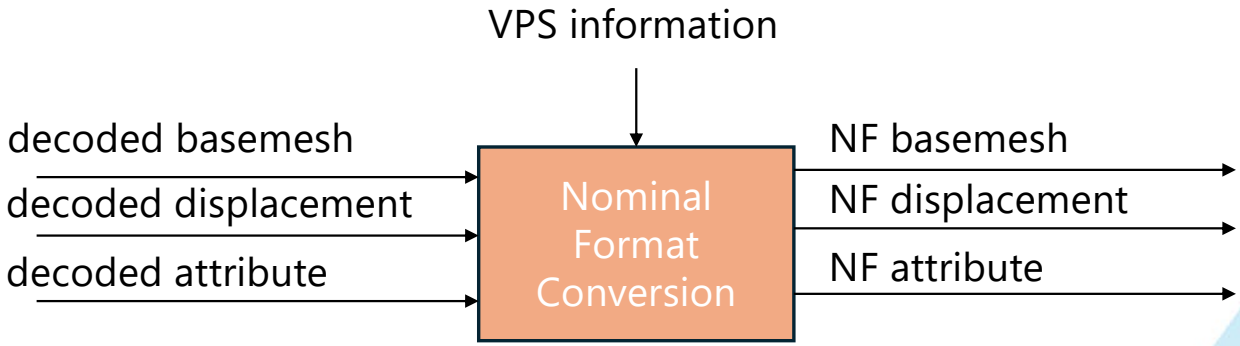
- What are the outputs of the video decoders
  - e.g. 4:2:0 for video to be converted to RGB
  - e.g. video resolution to be up/down-sampled according to the expected resolution at reconstruction
  - e.g. frames missing or in different order from the other bitstreams
  - etc.
- What are the outputs of the non-video decoders (basemesh and AC displacement)
  - e.g. number and order of components (mesh attributes, AC displacement components) are different than what is expected in reconstruction
  - e.g. frames missing or in different order from the other bitstreams
  - etc.

Reconstruction Pipeline

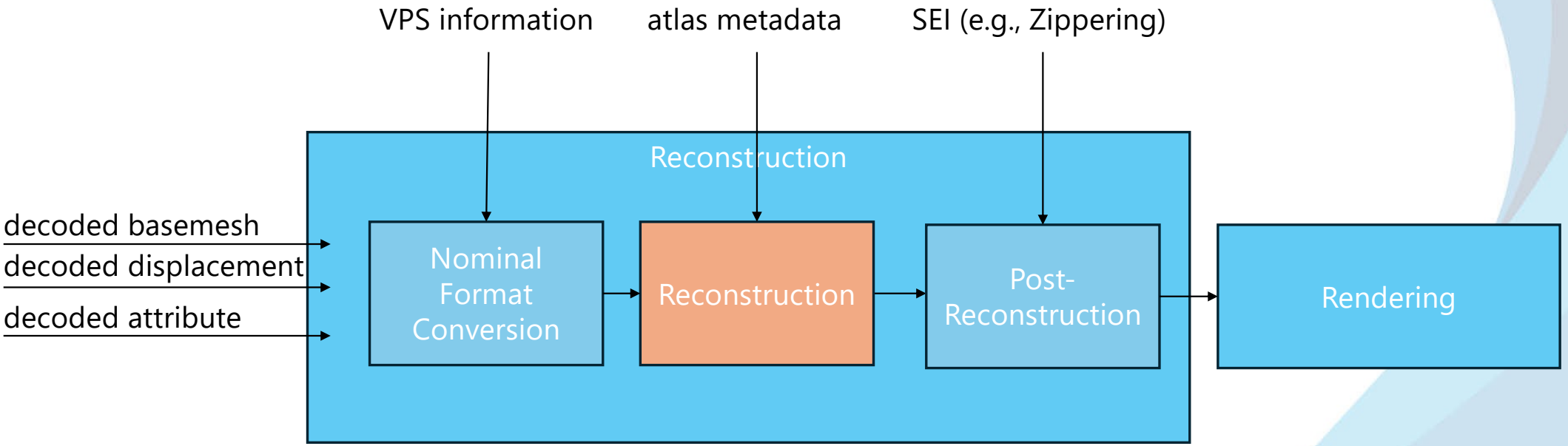


Nominal Format Conversion

- Overview of Annex B
  - Conversion process to align the decoded sub-bitstreams to the expected formats for reconstruction
    - Expected format is described by variables in VPS (e.g. bit depth, frame resolution, etc.)
  - V-DMC specification provides an informative way of converting the data
    - Alternative conversion processes are acceptable.



Reconstruction Pipeline

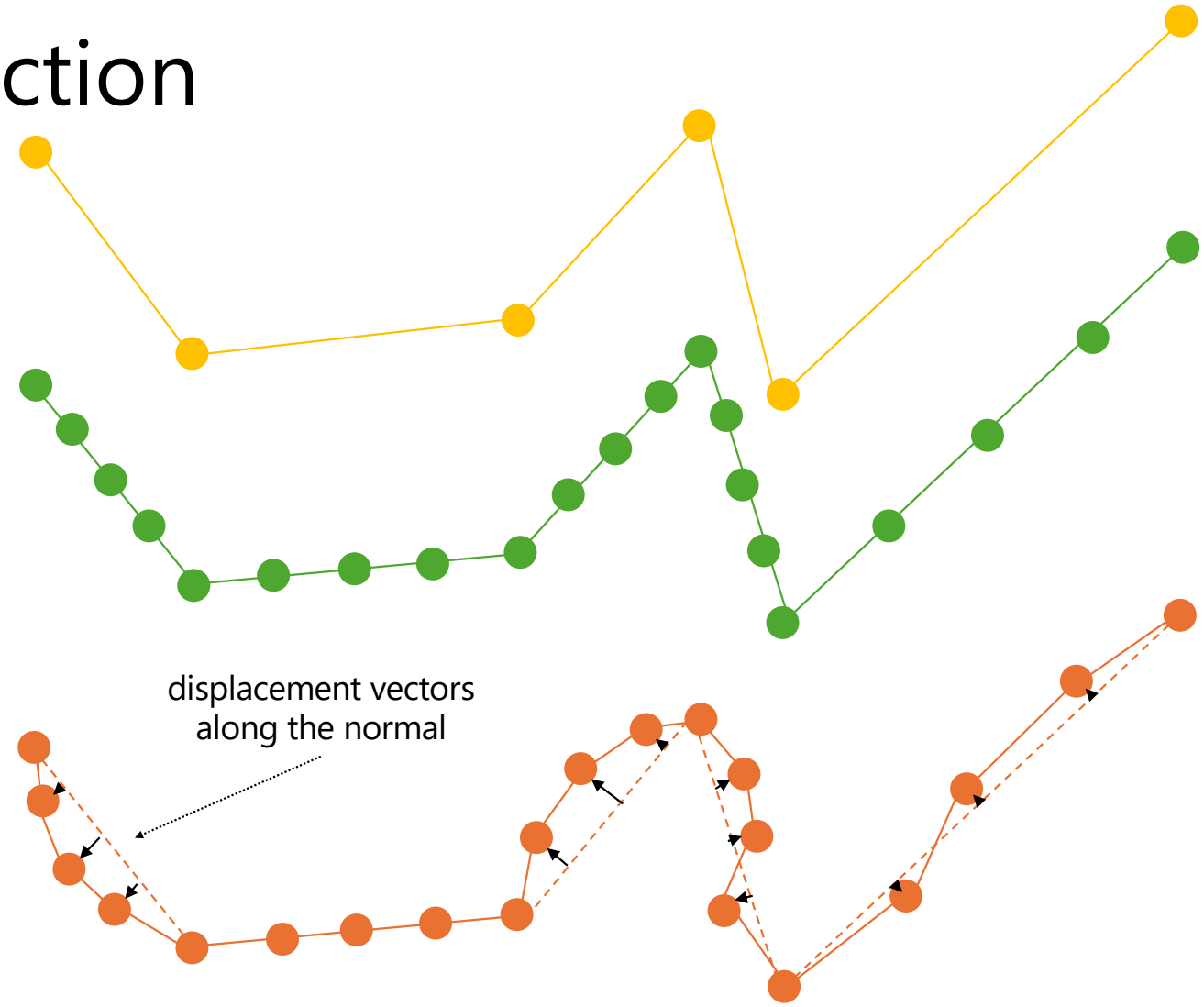


Reconstruction

decoded

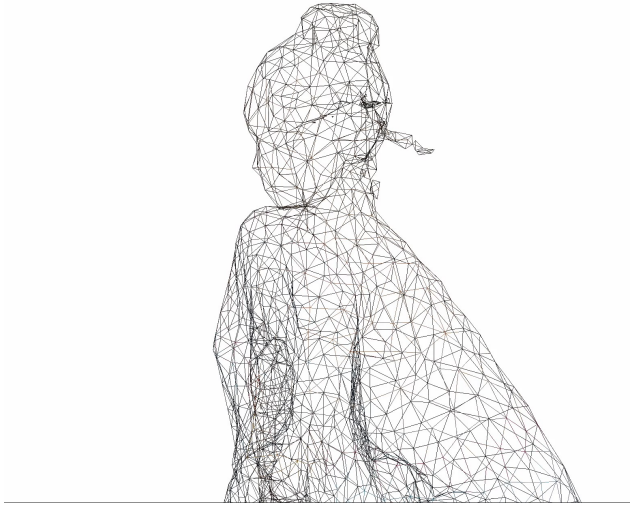
subdivided

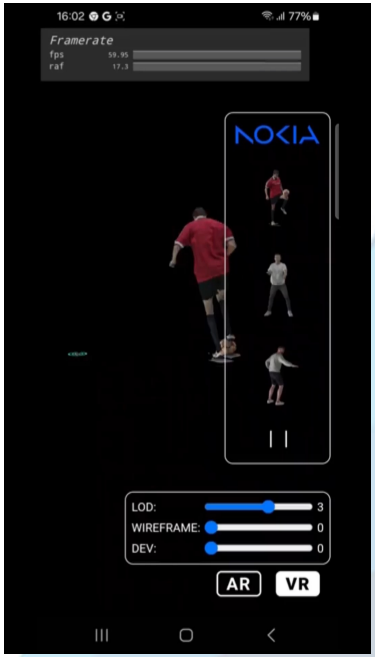
displaced



Reconstruction

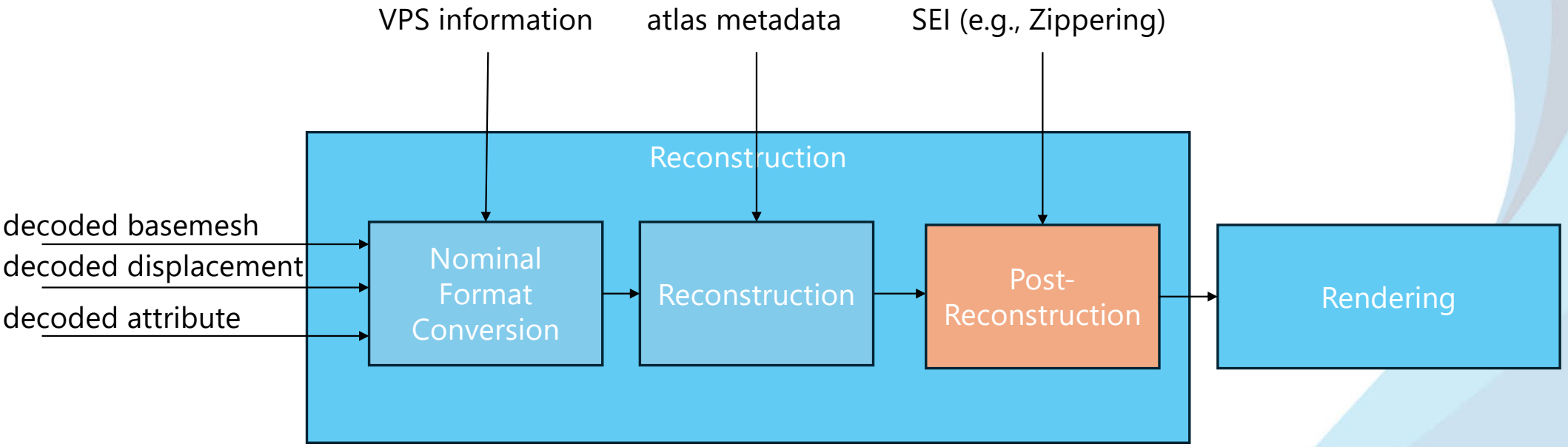
- number of implementation were showcase during standardization
- The mobile phone used Samsung S23 and OnePlus 11.
- Same chip in both: Qualcomm Snapdragon 8 Gen 2.





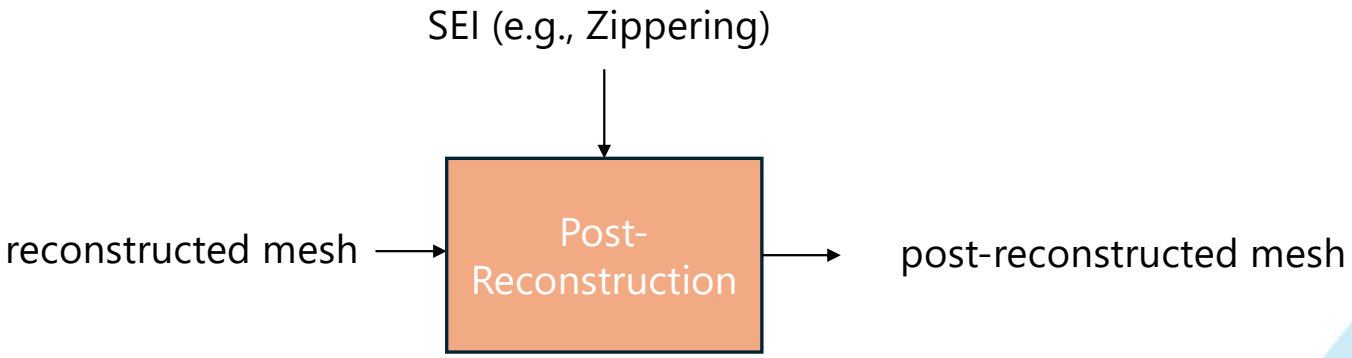
A. Martemianov et. al. "Real-Time Decoder and Player for MPEG V-DMC Dynamic Meshes," *2025 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*

Reconstruction Pipeline



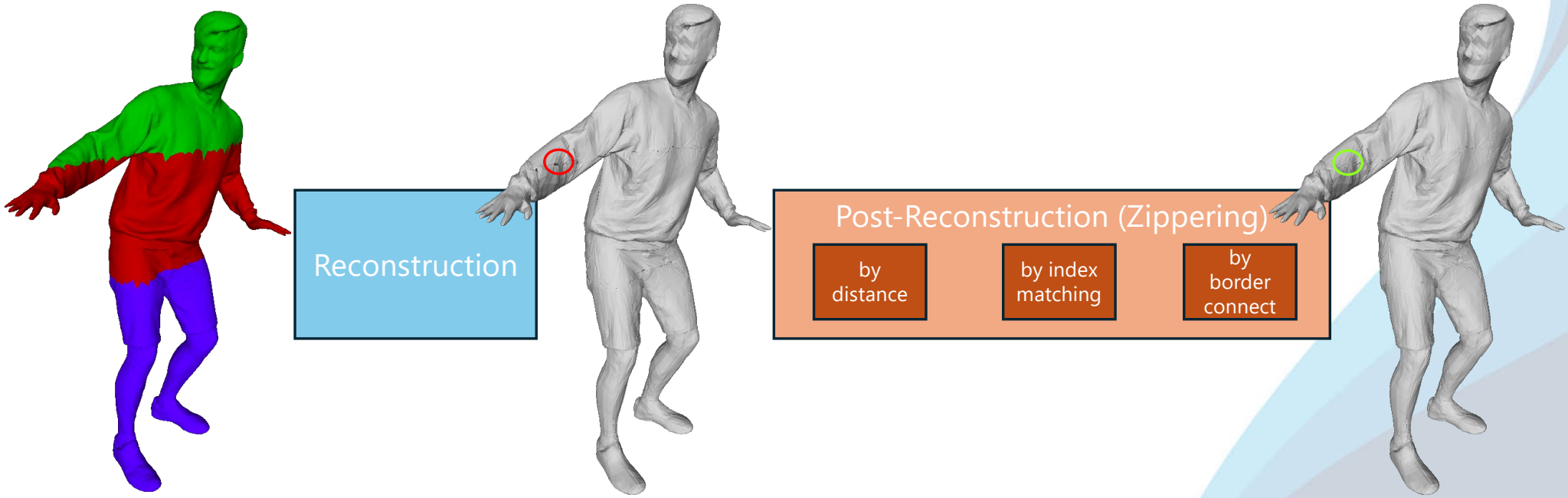
Post-reconstruction

- Overview of subclause 12
  - Description of methods that may be applied to the reconstructed mesh, usually to improve geometry/attribute quality
  - Methods are optional, and are signalled using SEI messages



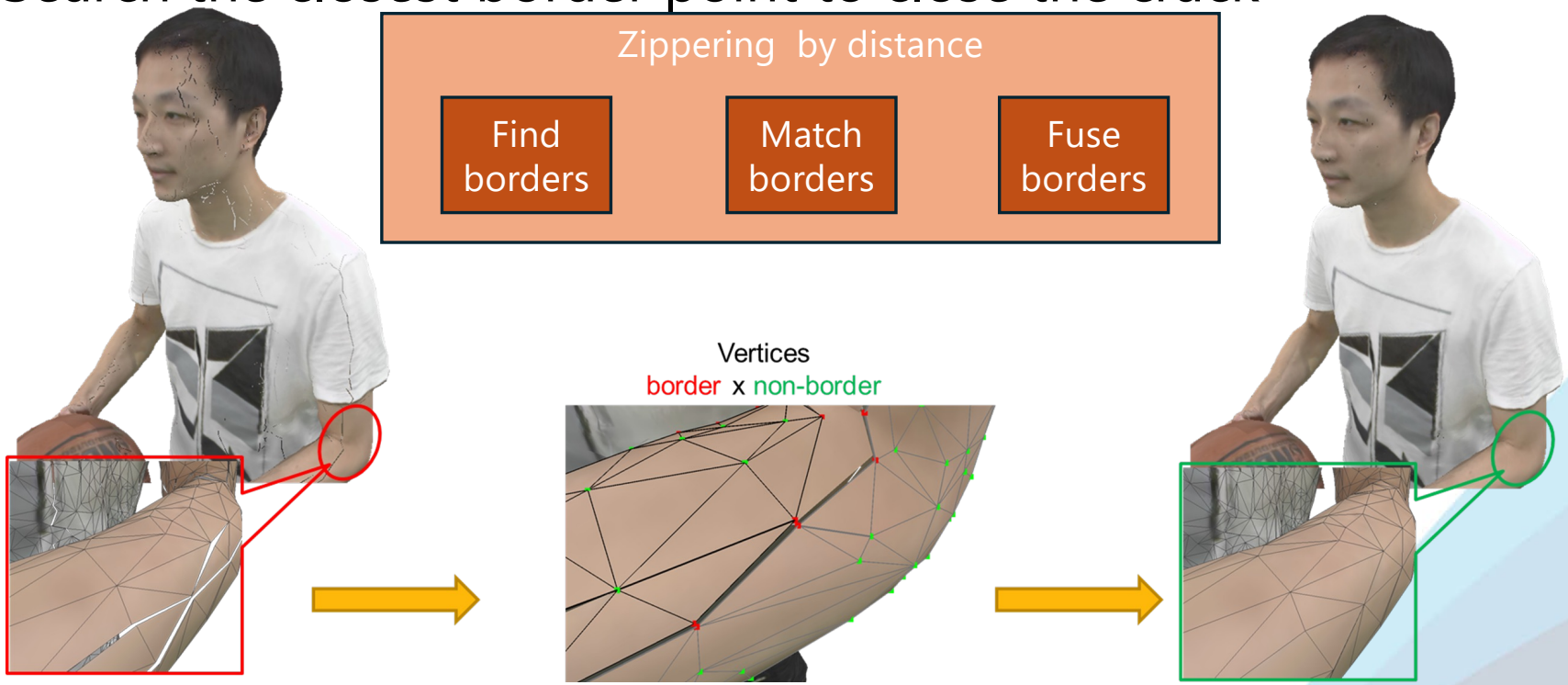
Post-reconstruction

- Zippering
  - Method to improve reconstruction of meshes segmented into submeshes by eliminating cracks due to compression



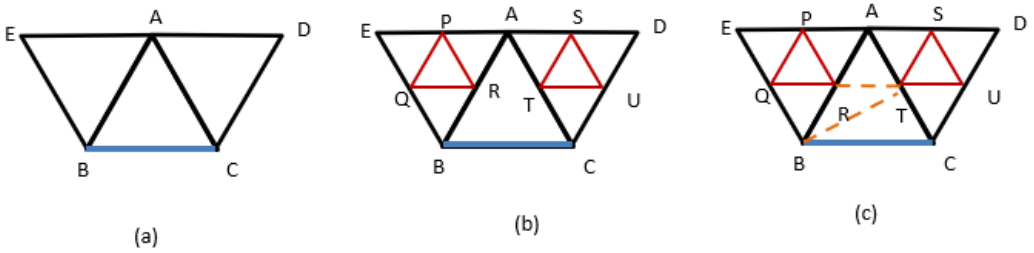
Zippering by Distance

- Search the closest border point to close the crack



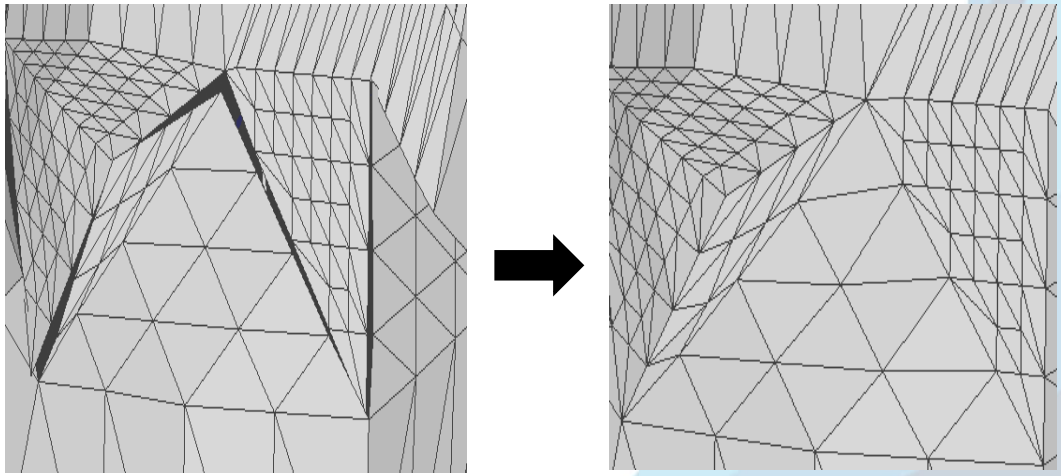
Zippering by Index Matching

- Uses transmitted index to close the crack
- Includes method to address different number of subdivisions at borders



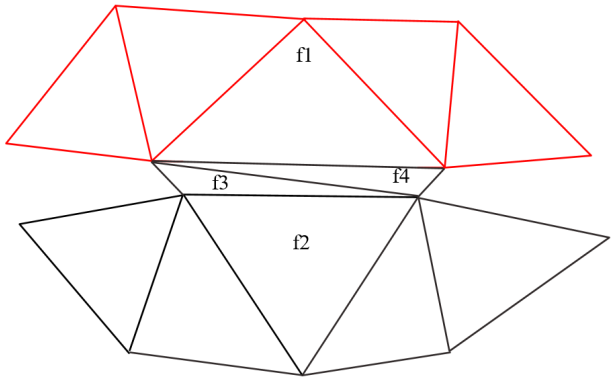
Internal edges:
Border edges:

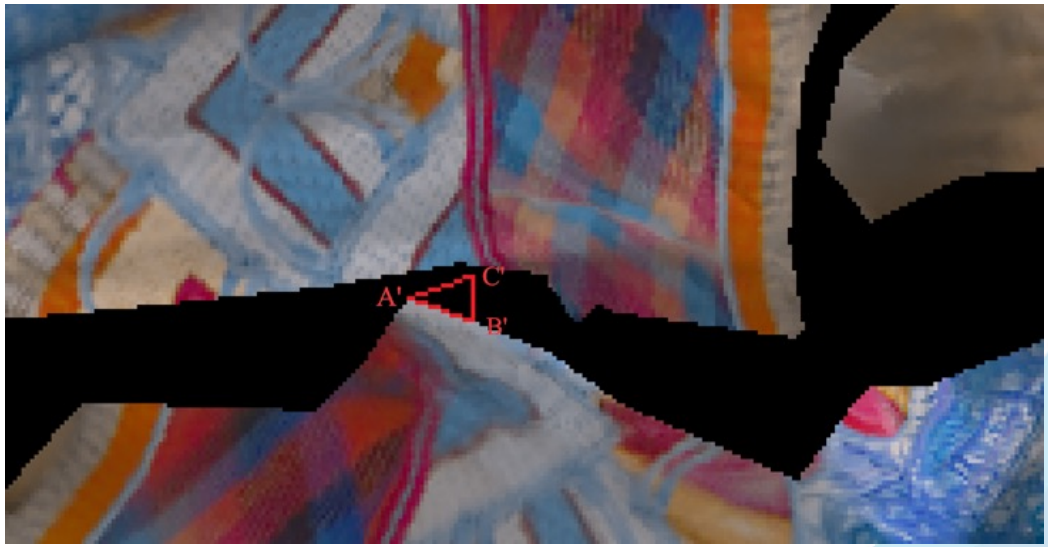
Figure 1 An example of connecting the border and internal vertices when number of subdivision iterations along the border is less than internal edges.



Zippering by Border Connect

- Add new triangles to close the crack between submeshes





Agenda

- **Motivations for V-DMC and overall architecture** (Marius Preda – 15 minutes)
- **Core technologies and coding features** (Danillo Graziosi – 1 hour)
- **Syntax and semantics** (Lukasz Kondrad – 30 minutes)
- **Reconstruction techniques** (Lukasz Kondrad + Danillo Graziosi – 30 minutes)
- **Performance metrics and evaluation** (Wenjie Zou – 25 minutes)
- **Conclusions and future directions** (Marius Preda – 10 minutes)

Overview

- V-DMC dataset
- Quality metrics
  - Objective metrics
  - Subjective metric
- Performance of V-DMC test model version 13.0
  - Lossy mesh compression
  - Lossless mesh compression

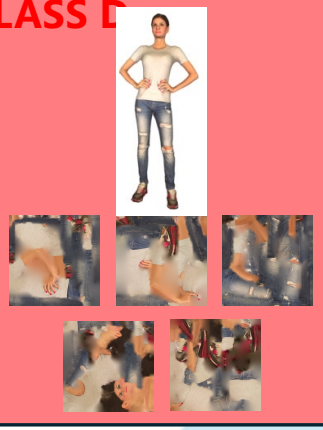
V-DMC Dataset

CLASS A

CLASS B

CLASS C

CLASS D

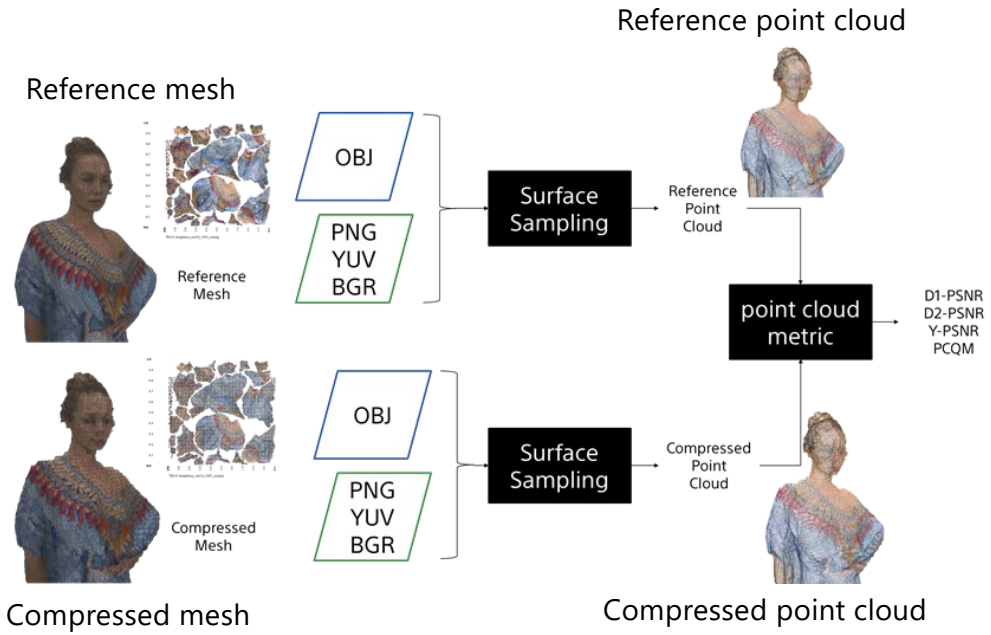


Test Category	Test Class	Test material dataset filename	# Frames	# Vertices	# Faces	Geometry Precision	Texture Coord. Precision	Texture Map Size	Color Attribute	Sequence Number
Dynamic Objects with Texture Mapping	A	longdress	300	22k	40k	10 bits	12 bits	2k x 2k	NA	1
		soldier	300	22k	40k	10 bits	12 bits	2k x 2k	NA	2
	B	basketball_player	300	20k	40k	12 bits	12 bits	2k x 2k	NA	3
		dancer	300	20k	40k	12 bits	12 bits	2k x 2k	NA	4
	C	mitch	300	16k	30k	12 bits	13 bits	4k x 4k	NA	5
		thomas	300	16k	30k	12 bits	13 bits	4k x 4k	NA	6
		football	300	25k	40k	12 bits	13 bits	4k x 4k	NA	7
		levi	150	20k	40k	12 bits	13 bits	4k x 4k	NA	8
Dynamic Objects with Multiple Texture Mapping	D*	rwtt422	1	200k	100k	12 bits	12 bits	1k x 1k x 4	NA	9

- Vertex attributes are associated with the surface of the mesh through a mapping process to a set of 2D attribute maps.

- The test class is an indicator of how complex a mesh is to encode, where A is the lowest and C the highest complexity.
- D has a single frame with multiple texture mapping.

Objective metrics - point-based



The point-based metric converts the reference and distorted meshes into two point clouds by applying the sampling procedure.

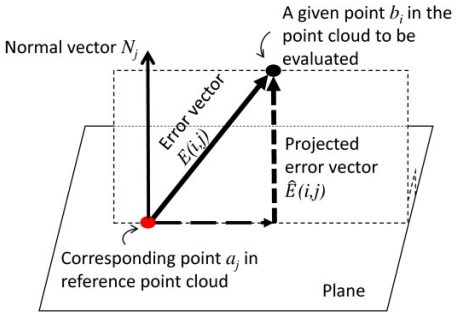


Illustration of point-to-point distance (**D1**) and point-to-plane distance (**D2**)

D1: $e^{D1}_{B,A}(i) = \|E(i,j)\|_2^2$ $\quad e^{D1}_{B,A} = \frac{1}{N_B} \sum_{\forall b_i \in B} e^{D1}_{B,A}(i)$

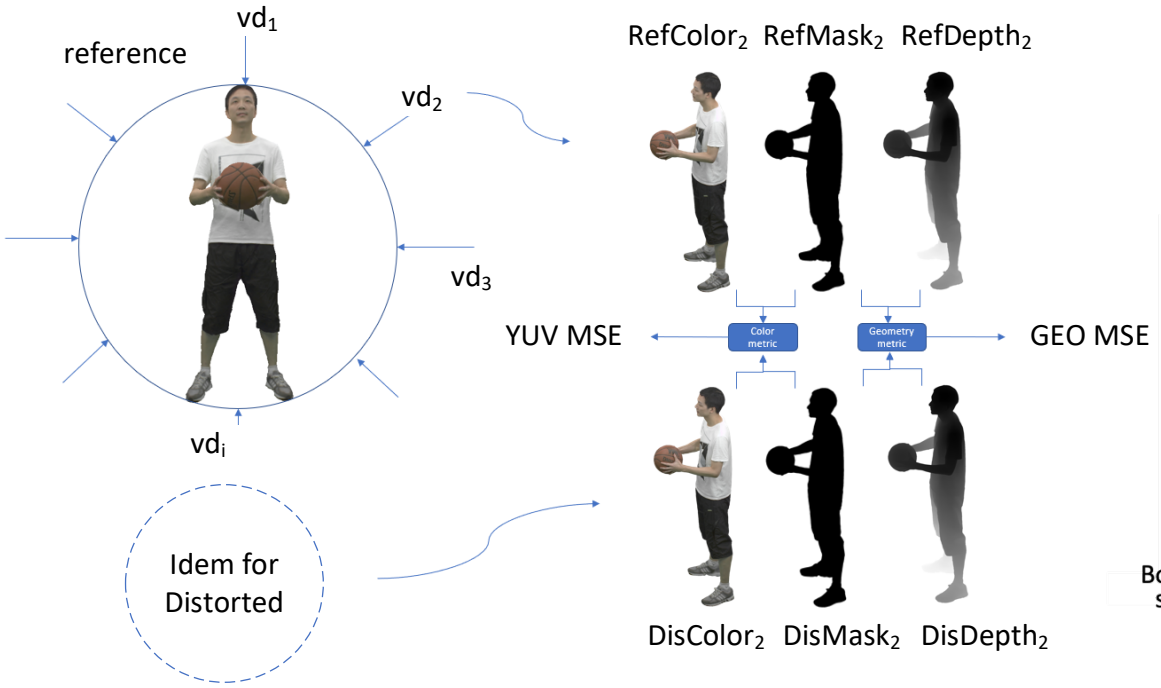
D2: $e^{D2}_{B,A}(i) = \left\|\hat{E}(i,j)\right\|_2^2 = (E(i,j) \cdot N_j)^2$ $\quad e^{D2}_{B,A} = \frac{1}{N_B} \sum_{\forall b_i \in B} e^{D2}_{B,A}(i)$

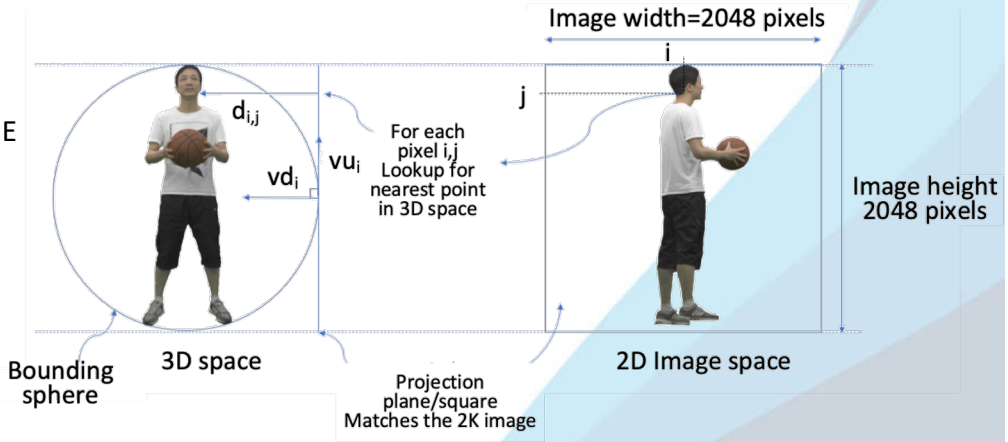
$$PSNR = 10 \log_{10}\left(\frac{3p^2}{MSE}\right) \quad e^{D1}_{symm} = \max\left(e^{D1}_{B,A}, e^{D1}_{A,B}\right) \text{ and } e^{D2}_{symm} = \max\left(e^{D2}_{B,A}, e^{D2}_{A,B}\right)$$

$$PSNR = 10 \log_{10}\left(\frac{p^2}{MSE}\right)$$

Objective metrics - image-based

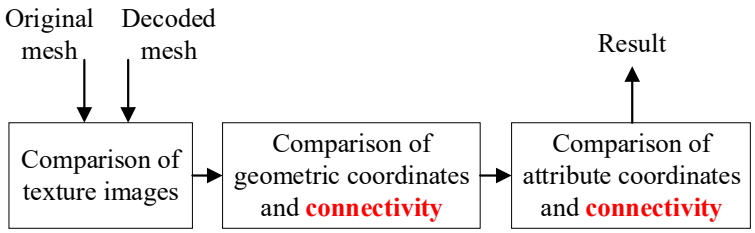
Image-based metrics





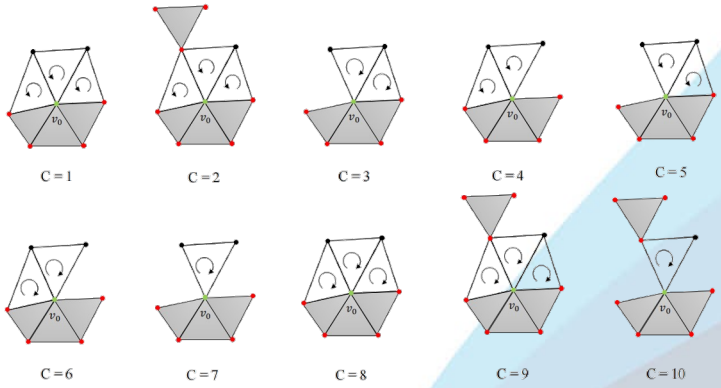
Objective metrics - lossless

Compared the geometry, texture coordinates, texture images, and **connectivity**.



Compared the connectivity by using TFAN 10 modes:

- a) Traverse the vertices of the original mesh and decoded mesh to build TFAN
- b) Compare the TFAN of original mesh and decoded mesh to achieve the comparison of connectivity.
- c) Add the solution for handling the independent connectivity of UV coordinates.



W. Zou, X. He, A. Trioux and F. Yang, "Mmetric++: An improved quality metric to judge the lossless compression of 3D meshes," *2023 IEEE International Conference on Visual Communications and Image Processing (VCIP)*, Jeju, Korea, Republic of, 2023, pp. 1-5,

Subjective metric

- Define a camera path around the object and provide subjective MOS score (similar procedure used with point-cloud)

Lossless performance

- No subdivision, i.e., no displacement, in lossless compression
- Original: OBJ + PNG files
- Compressed: TMM 13.0 (atlas + basemesh(MEB) + HM 16.21)

	Geometry			Color			Total		
Sequence	original (Mbps)	compressed (Mbps)	ratio	original (Mbps)	compressed (Mbps)	ratio	original (Mbps)	compressed (Mbps)	ratio
longdress	723.601	12.890	0.018	1252.084	852.290	0.681	1975.685	865.187	0.438
soldier	730.648	13.610	0.019	987.339	612.736	0.621	1717.987	626.352	0.365
basketball_player	497.864	15.283	0.031	962.425	589.818	0.613	1460.289	605.107	0.414
dancer	829.633	24.969	0.030	630.656	581.190	0.922	1460.289	596.178	0.408
football	528.734	16.607	0.031	4281.629	2895.990	0.676	4810.363	2912.602	0.605
levi	713.032	24.955	0.035	3400.765	2434.140	0.716	4113.797	2459.105	0.598
mitch	547.441	14.110	0.026	4778.318	3616.560	0.757	5325.759	3630.678	0.682
thomas	534.187	13.292	0.025	3760.780	2734.130	0.727	4294.967	2747.433	0.640
Average	638.142	16.964	0.027	2506.750	1789.607	0.714	3144.892	1805.330	0.574

Lossless performance

- Draco: Draco v1.5.7
- V-DMC: TMM 13.0 basemesh (MEB)

	Geometry		
Sequence	Draco (Mbps)	V-DMC (Mbps)	Ratio
longdress	101.837	12.890	0.127
soldier	102.735	13.610	0.132
basketball_player	98.757	15.283	0.155
dancer	98.618	24.969	0.253
football	76.544	16.607	0.217
levi	151.351	24.955	0.165
mitch	105.527	14.110	0.134
thomas	76.086	13.292	0.175
AVERAGE	101.432	16.964	0.167

Note: draco does not reconstruct the filtered duplicate vertices and degenerate faces, nor the differentiated non-manifold structures. Draco also performs only entropy coding without any prediction processing

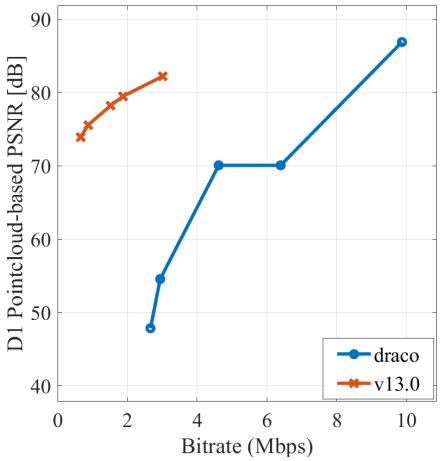
Lossy performance

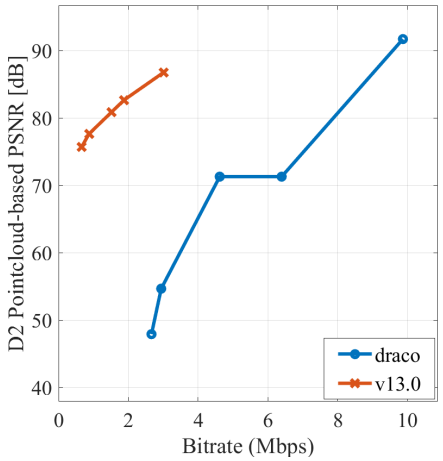
- Anchor: geometry (Draco v1.5.7) + color (HM 16.21)
- V-DMC Test Model v13.0: atlas + basemesh (MEB + Motion Coding) + color/displacement (HM 16.21)

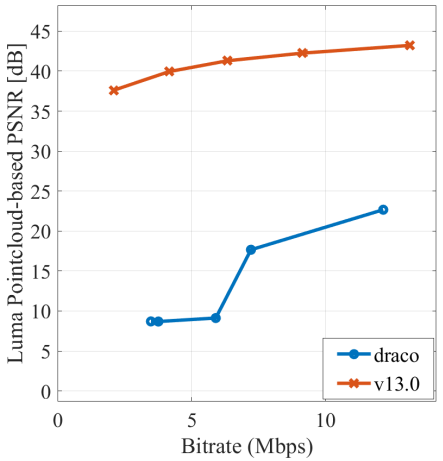
		C1_ai — lossy geometry, lossy attributes [all intra]							C2_ra — lossy geometry, lossy attributes [inter, random access]						
		Pointcloud-based BD-Rate [%]					Image-based BD-Rate		Pointcloud-based BD-Rate [%]					Image-based BD-Rate	
					Chroma							Chroma			
Class	Sequence	D1	D2	Luma	Cb	Chroma Cr	Geom	Luma	D1	D2	Luma	Cb	Chroma Cr	Geom	Luma
cat1-A	longdress_voxelised	-73.9%	-75.2%	-58.5%	-28.8%	-33.3%	-100.0%	-48.1%	-68.3%	-69.2%	-62.4%	-39.3%	-42.9%	-100.0%	-45.2%
	soldier_voxelised	-62.4%	-63.4%	-65.0%	-70.4%	-74.1%	-90.3%	-48.9%	-60.1%	-61.1%	-100.0%	-100.0%	-100.0%	-100.0%	-68.0%
cat1-B	basketball_player_voxelised	-71.3%	-71.7%	-82.6%	-100.0%	-84.3%	-91.9%	-67.2%	-67.7%	-68.5%	-78.6%	-80.9%	-77.5%	-92.4%	-64.8%
	dancer_voxelised	-67.6%	-68.8%	-77.3%	-77.3%	-75.7%	-100.0%	-62.8%	-68.8%	-69.4%	-76.5%	-76.0%	-75.2%	-100.0%	-63.7%
cat1-C	football_voxelised	-75.2%	-74.9%	-100.0%	-100.0%	-80.5%	-100.0%	-81.9%	-65.9%	-65.0%	-100.0%	-100.0%	-100.0%	-100.0%	-100.0%
	levi_voxelised	-78.8%	-77.8%	-100.0%	-100.0%	-100.0%	-100.0%	-100.0%	-81.9%	-81.4%	-100.0%	-100.0%	-100.0%	-100.0%	-100.0%
	mitch_voxelised	-59.7%	-61.2%	-74.3%	-72.1%	-69.7%	-100.0%	-61.7%	-78.3%	-79.3%	-100.0%	-71.2%	-79.6%	-100.0%	-72.9%
	thomas_voxelised	-67.0%	-67.7%	-67.8%	-100.0%	-79.0%	-100.0%	-43.6%	-83.3%	-83.8%	-100.0%	-100.0%	-100.0%	-100.0%	-66.9%
Cat1-A average		-68.1%	-69.3%	-61.8%	-49.6%	-53.7%	-95.2%	-48.5%	-64.2%	-65.2%	-81.2%	-69.6%	-71.5%	-100.0%	-56.6%
Cat1-B average		-69.5%	-70.2%	-79.9%	-88.6%	-80.0%	-95.9%	-65.0%	-68.3%	-68.9%	-77.5%	-78.5%	-76.4%	-96.2%	-64.3%
Cat1-C average		-70.2%	-70.4%	-85.5%	-93.0%	-82.3%	-100.0%	-71.8%	-77.4%	-77.4%	-100.0%	-92.8%	-94.9%	-100.0%	-85.0%
Overall average		-69.5%	-70.1%	-78.2%	-81.1%	-74.6%	-97.8%	-64.3%	-71.8%	-72.2%	-89.7%	-83.4%	-84.4%	-99.1%	-72.7%

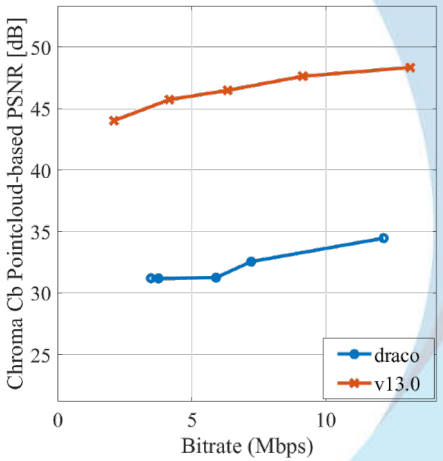
Note: Draco does not support RA mode. Anchor RA mode test is performed by encoding the color component with the RA mode of the video codec, while the geometry component is encoded in AI mode using Draco's.

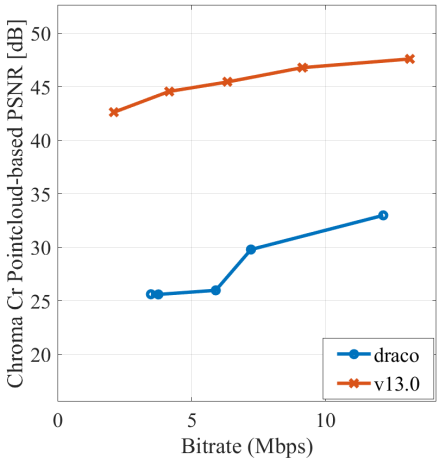
Anchor vs. V-DMC test model v13.0 - AI

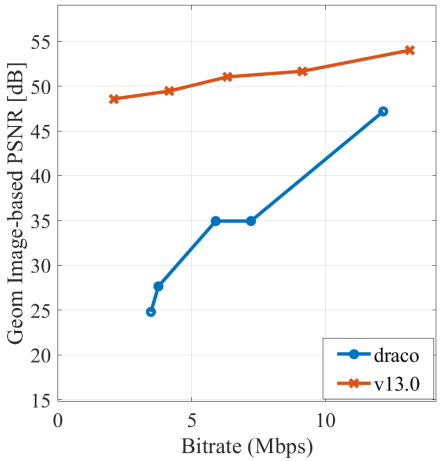


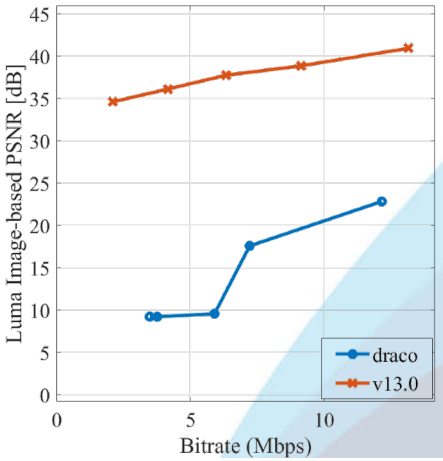


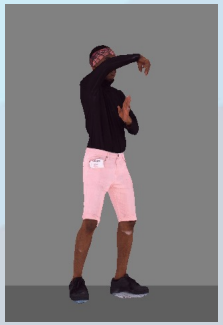




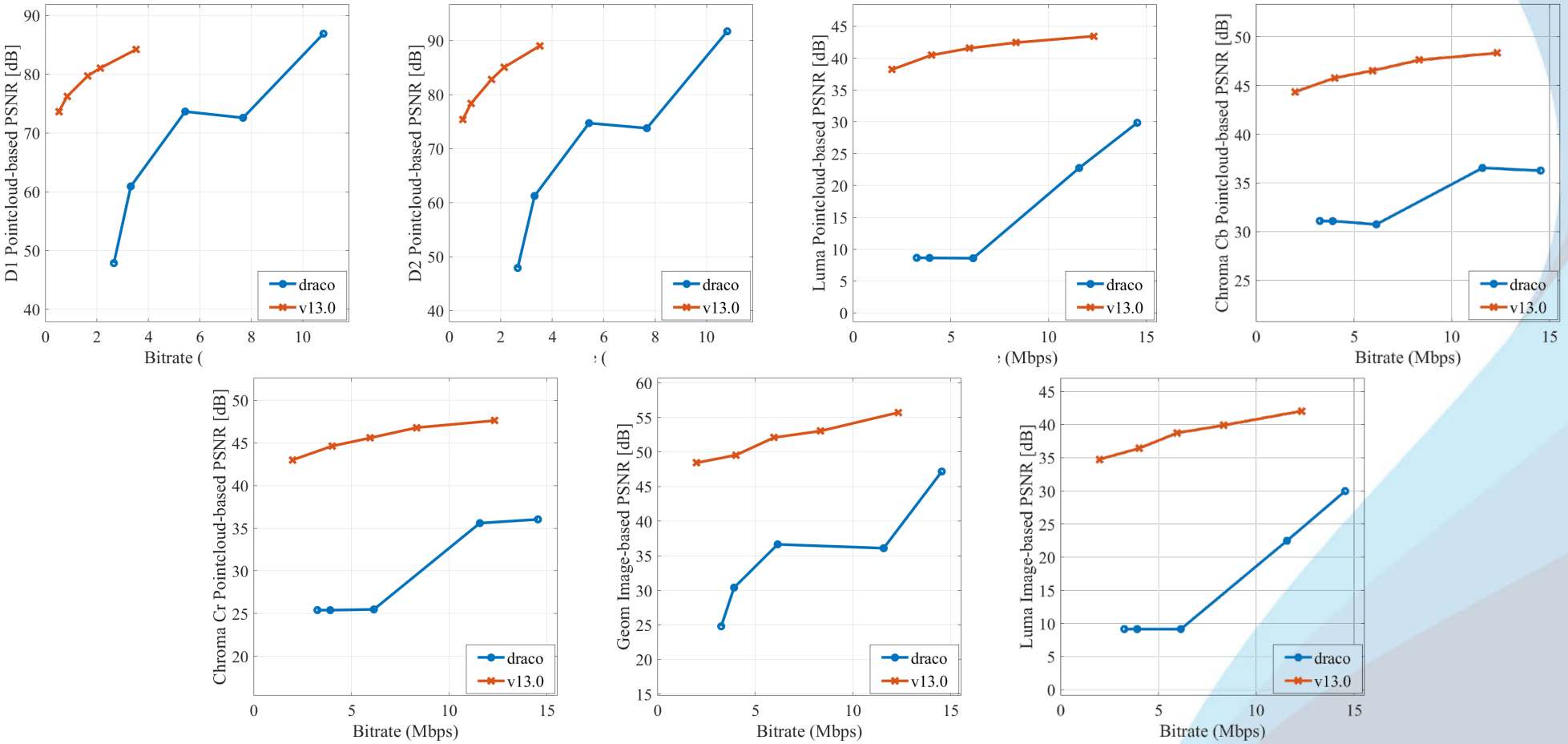








Anchor vs. V-DMC test model v13.0 - RA

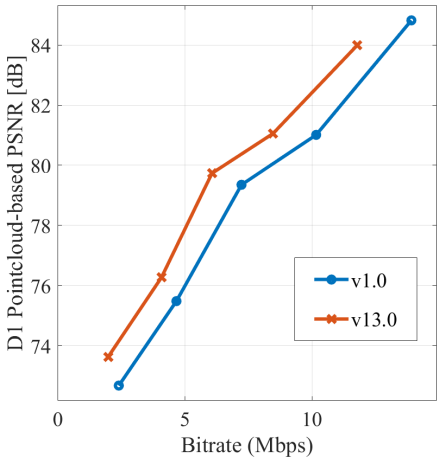


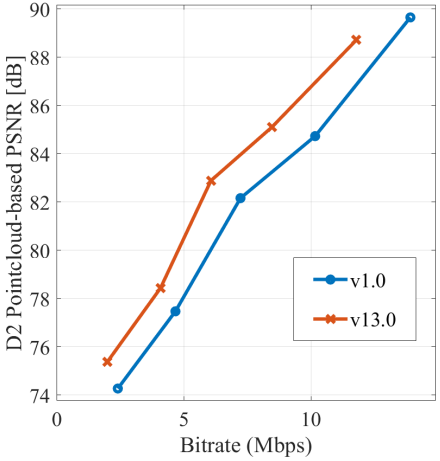
Lossy performance

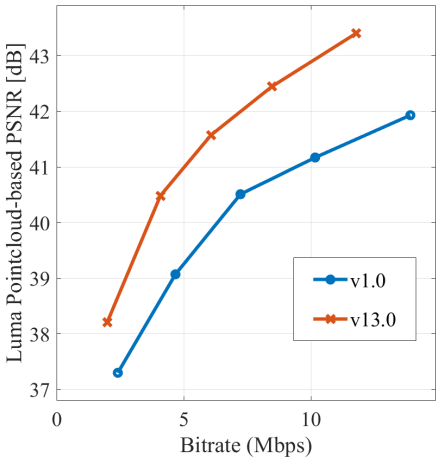
- V-DMC Test Model v1.0: atlas + basemesh (MEB + Motion Coding) + color/displacement (HM 16.21)
- V-DMC Test Model v13.0: atlas + basemesh (MEB + Motion Coding) + color/displacement (HM 16.21)

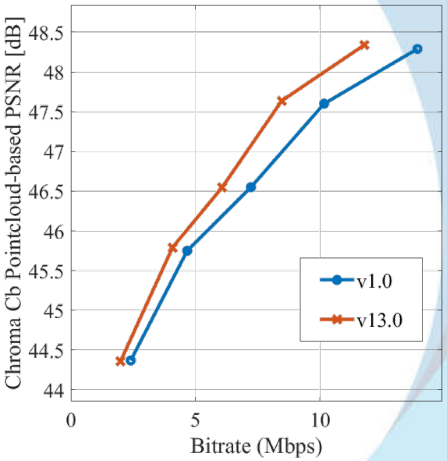
		C1_ai — lossy geometry, lossy attributes [all intra]							C2_ra — lossy geometry, lossy attributes [inter, random access]						
		Pointcloud-based BD-Rate [%]						Image-based BD-Rate	Pointcloud-based BD-Rate [%]						Image-based BD-Rate
					Chroma							Chroma			
Class	Sequence	D1	D2	Luma	Cb	Chroma Cr	Geom	Luma	D1	D2	Luma	Cb	Chroma Cr	Geom	Luma
cat1-A	longdress_voxelised	-30.5%	-27.0%	-24.0%	-9.4%	-12.2%	-34.1%	-27.1%	-36.8%	-36.3%	-22.2%	-10.0%	-12.2%	-33.3%	-26.4%
	soldier_voxelised	-31.0%	-31.1%	-21.8%	-8.2%	-8.7%	-31.0%	-27.4%	-64.0%	-32.7%	-46.3%	-13.2%	-13.5%	-39.2%	-59.1%
cat1-B	basketball_player_voxelised	-39.9%	-39.4%	-32.9%	-8.0%	-8.8%	-34.5%	-36.5%	-44.8%	-38.6%	-34.4%	-10.7%	-11.4%	-36.3%	-38.1%
	dancer_voxelised	-36.3%	-35.9%	-31.0%	-7.1%	-8.0%	-28.6%	-34.2%	-31.0%	-39.8%	-32.1%	-9.8%	-10.7%	-30.6%	-35.5%
cat1-C	football_voxelised	-37.4%	-30.7%	-28.9%	-8.4%	-11.0%	1.2%	-19.3%	-36.4%	-31.3%	-30.3%	-9.8%	-12.4%	-5.4%	-22.8%
	levi_voxelised	-15.5%	-16.1%	-41.3%	-12.7%	-13.5%	-14.8%	-23.0%	-20.8%	-21.9%	-45.4%	-16.2%	-17.4%	-19.0%	-26.1%
	mitch_voxelised	-39.6%	-35.7%	-35.3%	-16.7%	-18.4%	-41.1%	-29.0%	-22.9%	-21.1%	-30.5%	-8.4%	-17.5%	-23.9%	-29.7%
	thomas_voxelised	-29.3%	-36.5%	-42.7%	-14.7%	-13.5%	-26.9%	-41.5%	-29.0%	-25.2%	-52.7%	-13.8%	-16.4%	-24.5%	-47.1%
Cat1-A average		-30.7%	-29.0%	-22.9%	-8.8%	-10.5%	-32.6%	-27.2%	-50.4%	-34.5%	-34.2%	-11.6%	-12.8%	-36.2%	-42.7%
Cat1-B average		-38.1%	-37.6%	-32.0%	-7.6%	-8.4%	-31.6%	-35.4%	-37.9%	-39.2%	-33.2%	-10.3%	-11.1%	-33.5%	-36.8%
Cat1-C average		-30.4%	-29.7%	-37.1%	-13.1%	-14.1%	-20.4%	-28.2%	-27.3%	-24.9%	-39.8%	-12.0%	-15.9%	-18.2%	-31.4%
Overall average		-32.4%	-31.5%	-32.3%	-10.6%	-11.8%	-26.2%	-29.8%	-35.7%	-30.9%	-36.7%	-11.5%	-13.9%	-26.5%	-35.6%

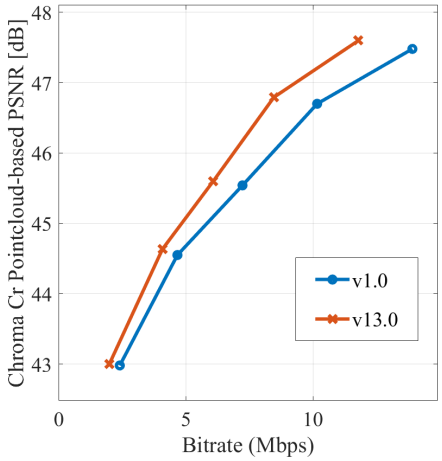
V-DMC test model v1.0 vs. v13.0 - AI

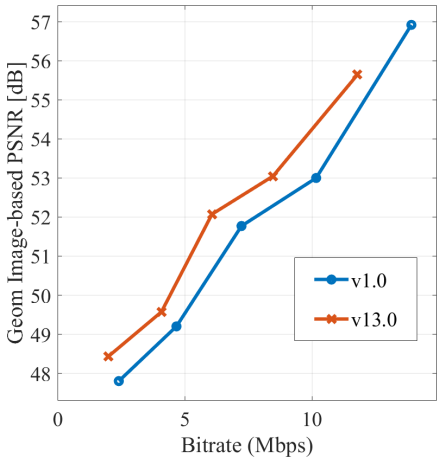


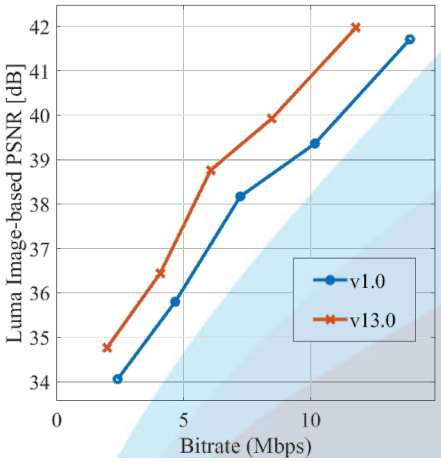




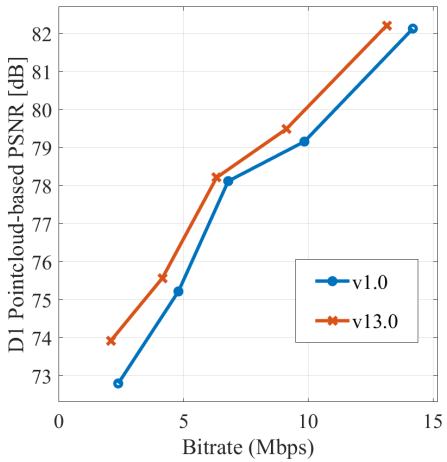


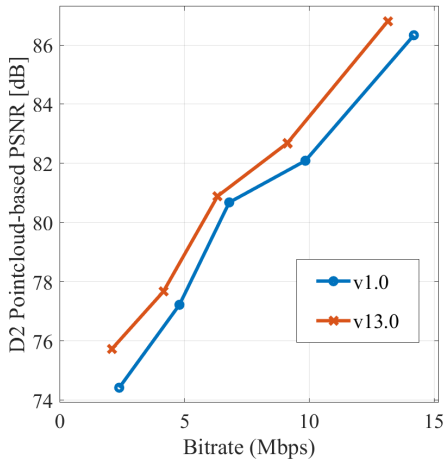


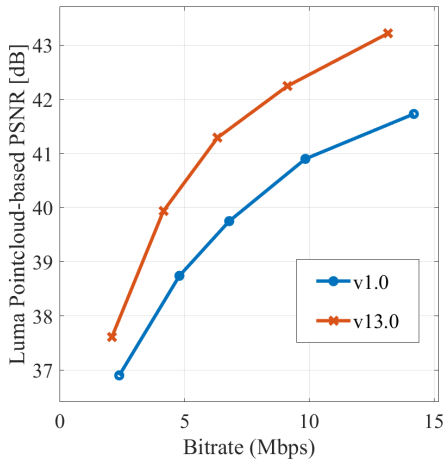


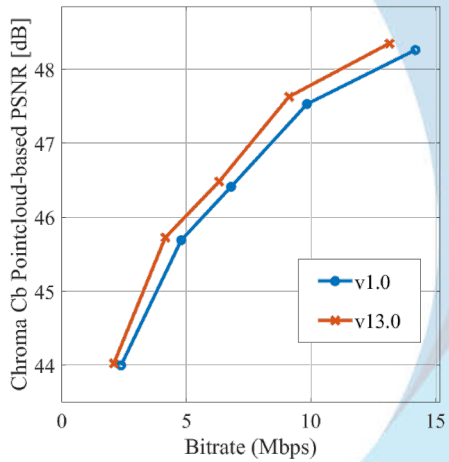


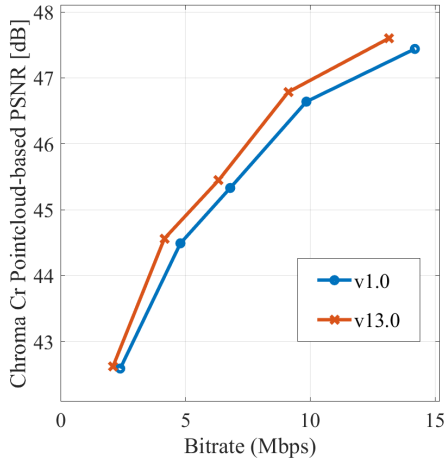
V-DMC test model v1.0 vs. v13.0 -RA

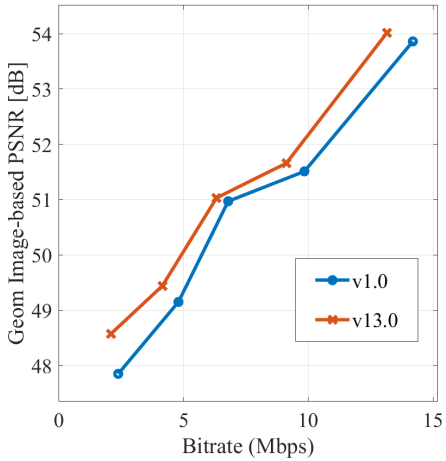


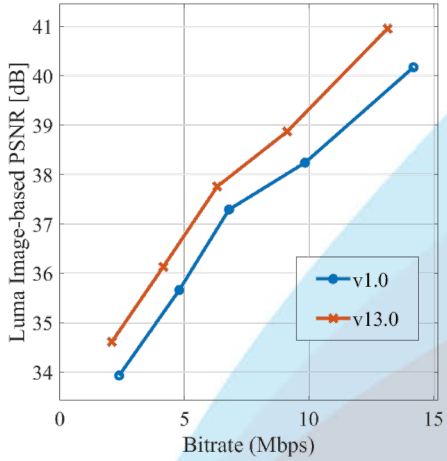












Draco vs. V-DMC test model

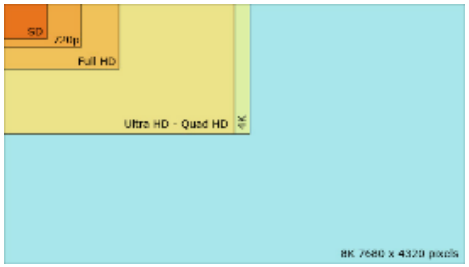
- Test condition of Draco v1.5.7

			AI					RA				
Test Class	Test material dataset filename	Rate	Draco QP	Draco QT	Mesh Resolution	HM QP	Texture Resolution	Draco QP	Draco QT	Mesh Resolution	HM QP	Texture Resolution
A	Longdress	R1	11	8	5	51	1	7	7	5	51	1
		R2	10	9	15	48	1	10	9	15	44	1
		R3	12	10	10	44	1	11	10	20	42	1
		R4	12	9	15	42	1	12	9	25	40	1
		R5	12	11	25	40	1	12	10	original	36	1
	Soldier	R1	9	8	10	51	1	7	7	10	51	1
		R2	10	9	20	48	1	10	8	20	44	1
		R3	12	8	20	44	1	11	9	25	42	1
		R4	12	9	25	42	1	12	9	25	36	1
		R5	12	11	original	38	1	12	10	original	32	1
B	Basketball_player	R1	8	7	5	51	1	7	8	5	51	1
		R2	9	8	10	48	1	10	8	10	48	1
		R3	12	10	15	44	1	12	9	20	42	1
		R4	12	10	25	42	1	12	10	25	36	1
		R5	12	10	original	38	1	12	11	original	32	1
	Dancer	R1	8	7	5	51	1	7	7	5	48	1
		R2	9	7	10	48	1	9	9	10	44	1
		R3	11	8	20	44	1	12	7	20	40	1
		R4	12	11	20	40	1	12	10	25	36	1
		R5	12	10	original	38	1	12	10	original	32	1
C	Mitch	R1	7	8	5	51	1	10	8	5	48	1
		R2	10	8	5	48	1	12	9	5	44	1
		R3	10	9	10	44	1	12	8	10	42	1
		R4	11	9	15	42	1	11	9	15	40	1
		R5	12	11	20	40	1	12	10	20	38	1
	Thomas	R1	11	9	5	51	1	12	8	5	51	1
		R2	11	7	10	48	1	10	9	10	48	1
		R3	12	11	15	44	1	11	9	15	44	1
		R4	12	11	25	42	1	12	11	15	40	1
		R5	12	12	original	38	1	12	11	25	38	1
	Football	R1	6	7	5	51	4	6	7	5	51	4
		R2	9	8	15	48	4	10	9	15	48	4
		R3	11	9	20	51	2	12	8	25	44	4
		R4	12	11	25	48	2	12	10	25	40	2
		R5	12	11	original	44	2	12	11	original	44	1
	Levi	R1	5	2	25	51	4	5	2	25	51	4
		R2	6	2	25	51	4	7	2	25	51	4
		R3	9	3	25	48	4	11	2	25	48	4
		R4	9	6	25	51	2	10	7	25	42	4
		R5	12	9	original	36	4	12	7	original	44	4

Agenda

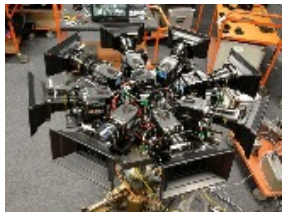
- **Motivations for V-DMC and overall architecture** (Marius Preda – 15 minutes)
- **Core technologies and coding features** (Danillo Graziosi – 1 hour)
- **Syntax and semantics** (Lukasz Kondrad – 30 minutes)
- **Reconstruction techniques** (Lukasz Kondrad + Danillo Graziosi – 30 minutes)
- **Performance metrics and evaluation** (Wenjie Zou – 25 minutes)
- **Conclusions and future directions** (Marius Preda – 10 minutes)

Capturing more and more from the reality



HD, Full HD, 4K, 8K

LDR, HDR



Multi-camera

Stereoscopy

What is the most appropriate representation format?

- Video + depth
- Many videos + depths

MPEG Video

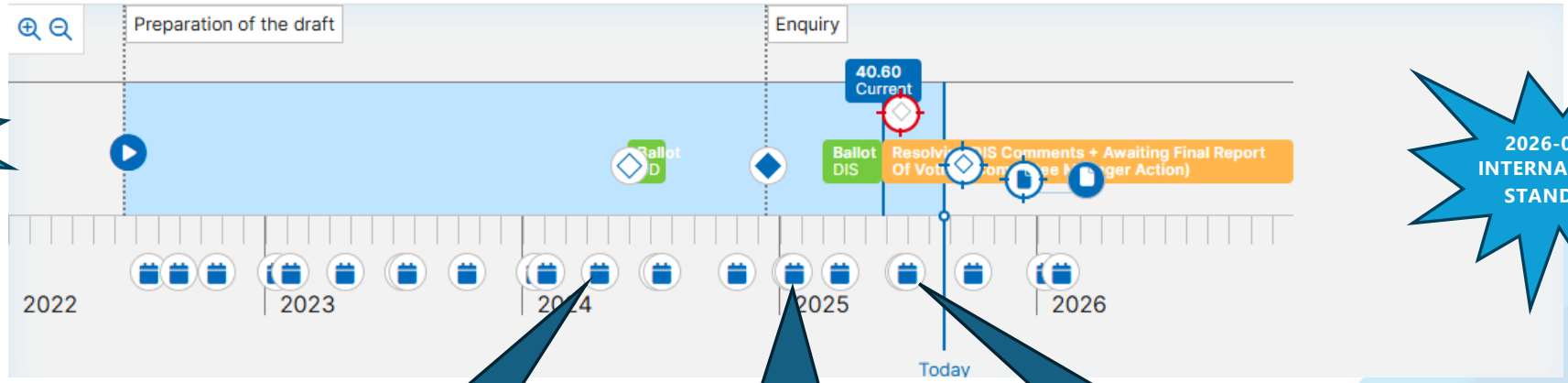
- Point clouds
- Meshes
- Gaussian Splatting

MPEG Graphics

V-DMC is finalized

Timeline

Registration date	Timeframe	Time since registration	In stage	STATUS
2022-06-16	57 months	38 months	40.60 for 2 months 24 days	en Resolving DIS comments + Awaiting final report of voting (Committee Manager action)



2022-06-15
START
PROJECT

2026-03-15
INTERNATIONAL
STANDARD

MPEG 146 Apr 2024
Committee Draft (CD)

Rennes

MPEG 149 Jan 2025
Draft of International Standards (DIS)

Geneva

MPEG 151 Jul 2025
Final Draft of International Standard (FDIS)

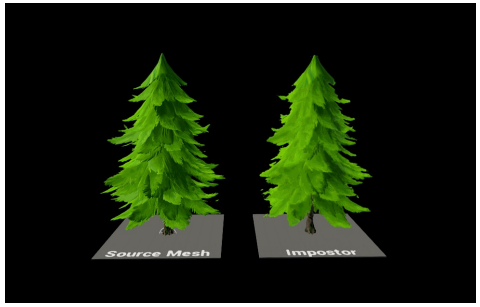
Daejeon

Next topic in graphics compression

- Gaussian Splats

Next topic in graphics compression

- Gaussian Splats
  - A new flavor of an old 3D Graphics mechanism: impostors



Gi :={ x,y,z,q,transparency,scale,f }

  - MPEG point cloud codecs (V-PCC and G-PCC) are able to handle such parameters
  - Ongoing work to adapt and improve MPEG tools for GSC

Conclusions

- Novel capturing systems and interactive 3D viewing technologies are creating new opportunities for realistic immersion
- Dynamic mesh compression enables interactive high quality 3D content by providing manageable bitrates and also reducing the constraints in creation and transmission of 3D content
- V-DMC leverages the existing hardware and software infrastructure for rapid deployment of new immersive experiences. It also provides a solid framework for the convergence between natural video content and synthetic 3D graphics.
- Gaussian Splat extension aim to improve rendering quality of realistic content

Acknowledgement

- The content used for the V-DMC standard is kindly provided by 8i, Owli, Volucap, and Interdigital

- Several pictures and videos used in this presentation are provided by Interdigital, Institut Mines Telecom, Nokia, Sony, Xidian University, Apple, KDDI, Qualcomm, Offino, Panasonic, and many others that contributed to the development of the V-DMC standard.

  - **We would like to thank all the experts that worked hard on V-DMC**