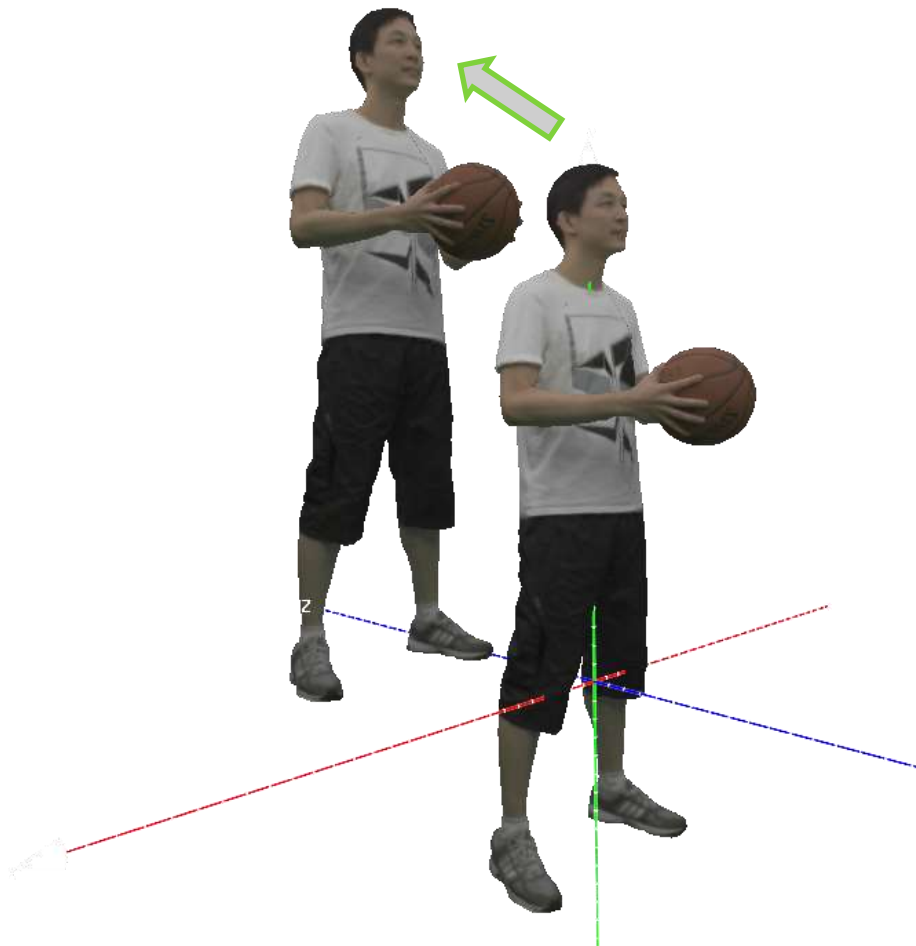


<m55367> Mesh voxelization

Danillo Graziosi, Alexandre Zaghetto, and Ali Tabatabai

<Problem statement>

- Mesh content may have floating point. Voxelization should happen before coding, but parameters should be sent in the bitstream.



No perceptual
difference



ORIGINAL



VOXELIZED

<NEW SYNTAX PROPOSAL>

- Send mesh scaling and offset in AAPS
 - Use available camera parameters in AAPS
 - OR
 - Introduce new syntax elements for voxelization
 - only scaling and offset are needed

	Descriptor
aaps_vpcc_extension() {	
aaps_vpcc_camera_parameters_present_flag	u(1)
if(aaps_vpcc_camera_parameters_present_flag)	
atlas_camera_parameters()	
aaps_voxelization_parameters_present_flag	u(1)
if(aaps_voxelization_parameters_present_flag)	
voxelization_parameters()	
}	

aaps_voxelization_parameters_present_flag equal to 1 specifies that voxelization parameters shall be present in the current atlas adaptation parameter set.
aaps_voxelization_parameters_present_flag equal to 0 specifies that voxelization parameters for the current adaptation parameter set shall not be present.

	Descriptor
voxelization_parameters() {	
vp_scale_enabled_flag	u(1)
vp_offset_enabled_flag	u(1)
if(vp_scale_enabled_flag)	
vp_scale	u(32)
if(vp_offset_enabled_flag)	
for(d = 0; d < 3; d++)	
vp_offset_on_axis[d]	i(32)
}	

See semantics in the next slide



<NEW SYNTAX PROPOSAL>

vp_scale_enabled_flag equal to 1 indicates that scale parameters for the current voxelization are present. **vp_scale_enabled_flag** equal to 0 indicates that scale parameters for the current voxelization are not present. When **vp_scale_enabled_flag** is not present, it shall be inferred to be equal to 0.

vp_offset_enabled_flag equal to 1 indicates that offset parameters for the current voxelization are present. **vp_offset_enabled_flag** equal to 0 indicates that offset parameters for the current voxelization are not present. When **vp_offset_enabled_flag** is not present, it shall be inferred to be equal to 0.

vp_scale specifies the value of the scale, *Scale*, for the current voxelization in increments of 2^{-16} . The value of **vp_scale** shall be in the range of 1 to $2^{32} - 1$, inclusive. When **vp_scale** is not present, it shall be inferred to be equal to 2^{16} . The value of *Scale* is computed as follows:

$$\text{Scale} = \text{vp_scale} \div 2^{16}$$

vp_offset_on_axis[d] indicates the value of the offset, *Offset[d]*, along the *d* axis for the current voxelization in increments of 2^{-16} . The value of **vp_offset_on_axis[d]** shall be in the range of -2^{31} to $2^{31} - 1$, inclusive, where *d* is in the range of 0 to 2, inclusive. The values of *d* equal to 0, 1, and 2 correspond to the X, Y, and Z axis, respectively. When **vp_offset_on_axis[d]** is not present, it shall be inferred to be equal to 0.

$$\text{Offset}[d] = \text{vp_offset_on_axis}[d] \div 2^{16}$$

This process specifies the reverse voxelization process, to go from voxelized decoded vertex values to floating point reconstructed values. The following applies:

```
for( n = 0; n < VertexCnt; n++)  
    for( k = 0; k < 3; k++ )  
        vertexReconstructed[ n ][ k ] = Scale * (decodedVertex[ n ][ k ]) + Offset[ k ]
```

Adaptation process is not
implemented in TMC2 yet

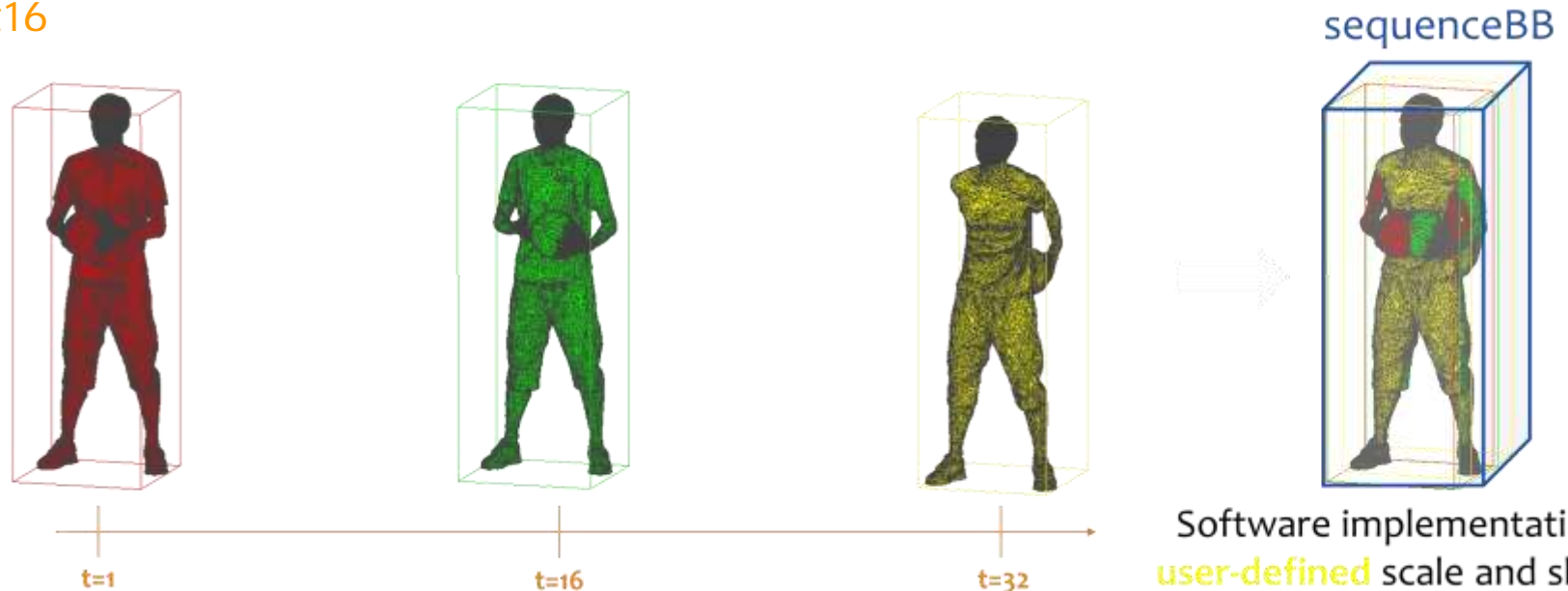
<VOXELIZATION PARAMETER CALCULATION>

- Bounding Box sequence normalization

- Get the bounding box of each frame, and calculate the sequence bounding box (see image below)
 - $\text{sequenceBB} = (\text{minPoint}, \text{maxPoint}) \rightarrow (\text{minPoint16}, \text{maxPoint16})$
- Calculate the scale to fit the maximum range in the range defined by the bit depth
 - $\text{maxRange} = \text{max}(\text{maxPoint16}[0..2] - \text{minPoint16}[0..2])$
 - $\text{scale} = (2^{\text{bitdepth}} - 1) / \text{maxRange} \rightarrow \text{scale16}$
- Scale and shift the result by the minimum value
 - $\text{voxelizedPoint} = \text{floor}((\text{originalPoint} - \text{minPoint16}) / \text{scale16})$
- Store the values in AAPS
 - Offset = **minPoint16**
 - Scale = **scale16**

Input Parameter (**modelScale**):

- (-1): per frame scale will be automatically calculated
- (0): sequence scale will be automatically calculated
- (>1): user-defined scale

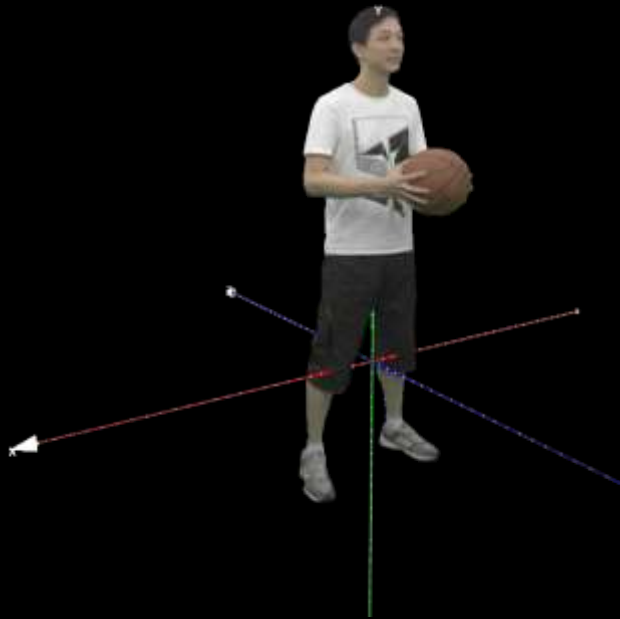


Software implementation allows **user-defined** scale and shift as well

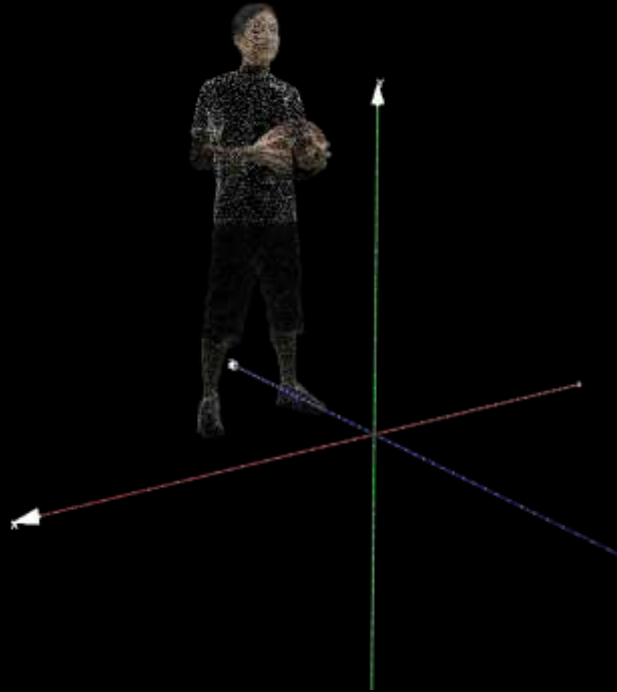
<EXPERIMENTAL RESULTS>

- **Voxelization done inside the TMC2**
 - Voxelization may degenerate triangles (vertices occupy the same position).
 - Coding procedure removes degenerate vertices and increases the number of vertices due to mesh segmentation
 - "Remove duplicate vertices" filter can be used to reduce the number of vertices.

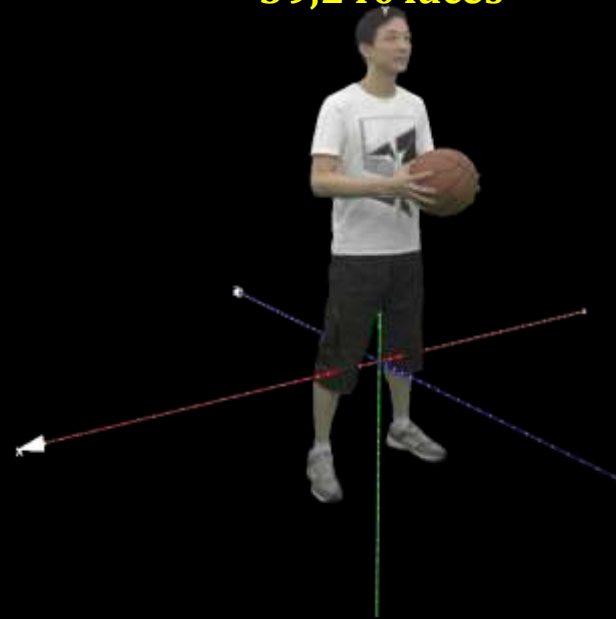
Original Mesh
20,692 vertices
39,455 faces



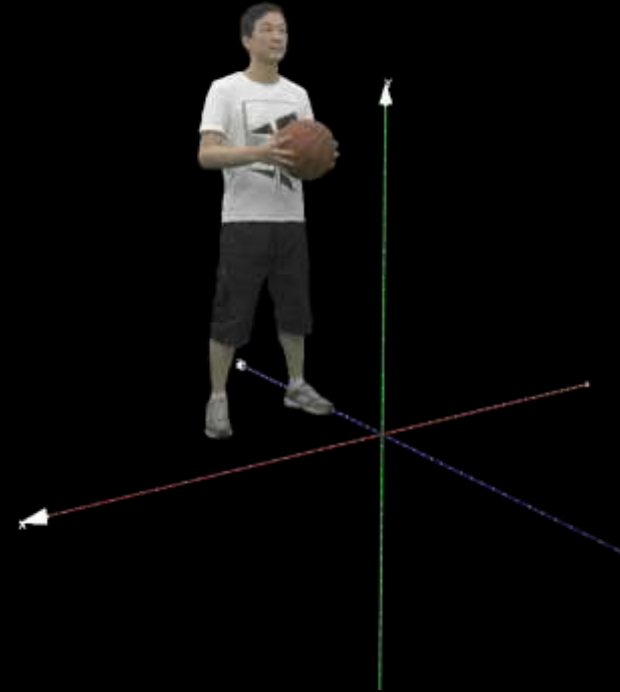
Voxelized vertices
20,692 points



Reconstructed Mesh
27,942 vertices
39,240 faces

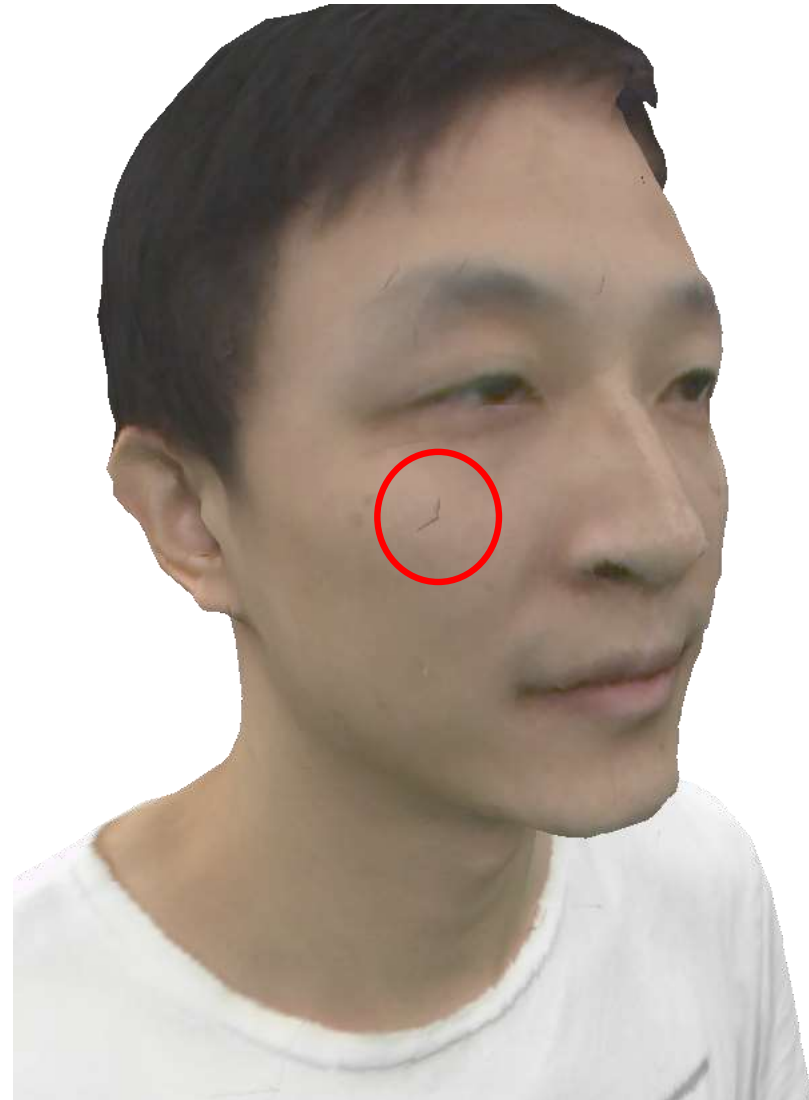


Reconstructed Point Cloud
1,938,384 points



<EXPERIMENTAL RESULTS>

- Voxelization may introduce artifacts. This should be studied further



<Conclusion>

- Implemented mesh voxelization in V3C-MESH software
 - Automatic generation of sequence voxelization parameters
 - User-defined input parameters
- Proposed signaling of voxelization using AAPS
 - Voxelization Parameters
- We suggest the group to:
 - Consider floating point meshes in the dataset
 - Consider the implementation of mesh voxelization
 - Discuss the signaling of the voxelization parameters