Cascaded TSDF for Multi-Camera Volumetric Capture with Perceptual Mesh Reduction

Abdul Rehman and François Bouille
French Touch Factory
Paris, France
email: {abdul | f.bouille}@frenchtouchfactory.com

***Abstract*—Volumetric capture requires adaptive resolution: fine detail for faces/hands, coarse for bodies/legs. We present a 5-level cascaded Truncated Signed Distance Function (TSDF) that allocates resolution by perceptual importance, achieving uniform-fine quality (2.4mm head error) with 83% memory reduction (4.8GB vs 28.5GB) and 4× speedup in TSDF fusion (32 Frame Per Second (FPS) vs 8 FPS for fusion only). To enable production deployment, we introduce: (1) 8-neighbor outlier filtering rejecting 73% of artifacts pre-fusion, (2) two-tier reduction (voxel+quadric) achieving 30:1 compression while preserving cascade detail, and (3) normal-validated texture transfer preventing color bleeding. The pipeline processes 9-camera captures with 512ms reconstruction latency (1.95 Hz) for near-real-time preview and 850ms export latency (1.18 Hz), generating Visual Effects (VFX)-compatible Alembic files (50K triangles, 2048×2048 texture, 98% coverage) suitable for offline batch processing. The proposed pipeline enables efficient, production-scale volumetric video capture by combining adaptive reconstruction with robust geometry and texture processing, supporting both near-real-time preview and high-quality offline asset generation.**

Keywords-volumetric capture; TSDF fusion; mesh decimation; texture reprojection; near-real-time reconstruction.

I. Introduction

High-performance volumetric capture of humans is essential for Virtual Reality (VR), gaming, and film production. The core challenge is integrating noisy depth data from multiple sensors into clean, watertight meshes while balancing temporal coherence, geometric detail, and practical memory constraints. KinectFusion [1] shows TSDF based fusion but produces high-polygon meshes unsuitable for direct commercial export. DynamicFusion [2] adds temporal coherence via non-rigid alignment, but requires offline post-processing. Commercial systems [3] produce high-quality results but demand 30+ controlled cameras in fixed studio environments. Deep learning approaches [4][5] achieve impressive completions but sacrifice the sub-millimeter geometric precision required for VFX pipelines.

In this paper, we address three main challenges in production-ready capture. First, we propose an 8-neighbor pre-fusion depth consistency check that eliminates silhouette bleeding before volumetric integration. Second, we introduce a two-tier mesh reduction combining voxel clustering with edge-aware quadric decimation using spatial weighting, achieving 30:1 compression while preserving perceptually important detail. Third, we propose normal-validated texture reprojection from the high-resolution mesh to the decimated mesh, preventing cross-surface color bleeding while maintaining 98% texture coverage.

Our primary contribution is a 5-level cascaded TSDF that allocates voxel resolution by perceptual importance: 2mm for head and hands, 3–4mm for shoulders, 6mm for torso, and 8mm for legs. We validate this through:

- **Cascade ablation:** comparing uniform-coarse (8mm), uniform-fine (2mm), and 3/4/5/6-level configurations. Our 5-level cascade matches uniform-fine quality (2.4mm vs. 2.3mm head error) with 83% memory reduction and 4× TSDF fusion speedup.
- **Production pipeline:** pre-fusion outlier filtering (73% artifact rejection), two-tier reduction (30:1 compression), and normal-validated texture transfer (98% coverage, 5.5 dB Peak Signal-to-Noise Ratio (PSNR) gain).
- **End-to-end deployment:** 500,000+ production frames exported to Alembic with a constant 2 GB RAM footprint.

The rest of the paper is structured as follows. Section II reviews related work. Section III describes the proposed pipeline. Sections IV–VI detail the TSDF fusion, mesh reduction, and texture reprojection methods. Section VII presents the results and Section VIII concludes the paper.

II. Related Work

Volumetric fusion systems, such as KinectFusion and DynamicFusion [1][2], produce high-polygon meshes with millions of triangles requiring offline reduction, while commercial systems [3] demand 30+ cameras in fixed studio setups. Deep learning approaches [4]–[6] improve completeness but sacrifice geometric precision. A direct quantitative comparison against neural baselines is beyond the scope of this work, as our pipeline targets deterministic sub-millimeter precision required by VFX production constraints that current learning-based methods do not guarantee. Classical TSDF methods, such as Curless and Levoy [7], use weighted averaging but cannot identify and remove silhouette artifacts. Similarly, InfiniTAM [8] applies bilateral filtering that smooths key features but cannot eliminate outliers before integration. For mesh reduction, quadratic error metrics [9] give better quality with $O(n \log n)$ complexity that doesn't allow real-time reduction of meshes, while vertex clustering [10] is fast but produces flat-shaded surfaces at aggressive reduction ratios. In contrast to these single-algorithm approaches, we combine both strategies: voxel clustering for an initial 3:1 reduction, followed by quadric decimation for 10:1 on the simplified mesh. The standard texture solutions utilize dilation [11] or multi-view blending [12] but cannot stop color bleeding. We use normal-validated reprojection with XAtlas unwrapping [13].

III. Pipeline Architecture

The proposed pipeline works in two modes. Mode 1: Reconstruction: Stages 1–4 proceed from depth acquisition through mesh reduction, with a per-frame latency of 512 ms (1.95 Hz). This enables on-set quality verification and performance monitoring. Mode 2: Export: Stages 5–6 add UV unwrapping, texture reprojection, and Alembic export, for a total per-frame latency of 850 ms (1.18 Hz).

We use 9 Orbbec Femto Mega depth cameras (640×576 resolution, 30 FPS) arranged in a cylinder of 1.8 m diameter and 2.2 m height, with cameras spaced at 40° intervals. The system runs on an RTX 4090 GPU (24 GB VRAM).

We calibrate and align depth data [14] prior to TSDF fusion. We extract isosurfaces per cascade, merge them with cascade-boundary blending, apply two-tier reduction, unwrap UVs [13], and finally bake color into textures by reprojection from the high-resolution mesh. The complete pipeline is shown in Figure 1.

IV. Cascaded TSDF Fusion with Outlier Rejection

A. Multi-Resolution Cascade Strategy

Single-resolution TSDF volumes face a fundamental trade-off: high resolution (2mm) captures detail but consumes excessive memory; low resolution (10mm) covers large volumes but loses critical detail. We employ five cascaded volumes with progressively coarser resolution (Table I). Cascade 0 covers 0.8^3m at 2mm for facial features. Cascade 4 covers 6.4^3m at 8mm for full-body. Finer cascades take precedence where regions overlap.

TABLE I. 5-Level Cascade Configuration.

Cascade	*Voxel Size*	*Grid*	*Volume*	*Body Region*
C0	2mm	256^3	0.5^3 m	Head, hands
C1	3mm	128^3	1.3^3 m	Right shoulder/arm
C2	4mm	128^3	2.0^3 m	Left shoulder/arm
C3	6mm	128^3	3.6^3 m	Torso
C4	8mm	128^3	6.0^3 m	Legs, feet

These thresholds were determined empirically by minimizing mean Hausdorff error subject to a 5 GB memory budget across our ten test sequences; finer resolution for legs yielded less than 0.1 mm improvement at 40% additional memory cost. Cascade-to-body-region assignment uses axis-aligned bounding volumes derived from a skeleton estimated by a single-frame pose detector run on the central RGB camera at the start of each sequence; boundaries are fixed for the sequence duration, as the studio volume constrains subject motion. This allocation is illustrated in Figure 2.

B. Parallel Processing Architecture

Cascades are processed sequentially due to resolution-dependent voxel grids—different voxel sizes (2mm vs 8mm) require different kernel configurations and cannot be batched. GPU parallelism is achieved within each cascade through simultaneous integration of depth measurements from all 9 cameras (3.5M pixels per cascade processed in parallel). This design prioritizes perceptual quality over GPU batch efficiency.

C. Perspective Projection Integration

We implement perspective projection from each camera position, computing signed distances along viewing rays. The truncation distance adapts based on camera distance:

$$\delta(r) = \delta_{base} + k\, r, \tag{1}$$

where $\delta_{base} = 3\,\mathrm{cm}$, $k = 0.02\,\mathrm{m}^{-1}$, and r is distance from camera to surface point.

D. 8-Neighbor Depth Consistency Check

Depth sensors produce systematic outliers at silhouette edges. We reject outliers before fusion using 8-neighbor consistency validation. For each depth pixel $d(u, v)$, we compute the median depth of eight spatial neighbors: $m = median\{d(u\pm1, v\pm1)\}$. If $|d(u, v) - m| > 3\sigma$ where $\sigma = 15\,\mathrm{mm}$, the measurement is discarded. This eliminates 73% of silhouette outliers while preserving 99.4% of valid measurements. Unlike prior work [8] which applies filtering post-integration, we reject outliers before volumetric integration. The processing overhead is 2 ms per frame.

V. Two-Tier Mesh Reduction Pipeline

A. Reduction Strategy Rationale

Marching cubes on five cascades produces 1.5M triangles per frame. Voxel clustering [10] achieves 3:1 reduction in 12ms but produces faceted results at higher ratios. Quadric decimation [9] generates quality 30:1 reductions but requires 1,240ms on 1.5M triangles. Our insight: different algorithms excel at different reduction ratios. We combine them sequentially for both speed and quality.

B. Stage 1: Voxel Clustering Reduction

We merge vertices to 4mm voxel grid centers, reducing triangles from 1.5M to 500k in 12ms. Total stage time including memory transfers and face reconstruction is 38ms (3:1 ratio).

C. Stage 2: Edge-Aware Quadric Decimation

We extend standard quadric error metrics [9] with spatial feature weighting:

$$E_{total} = E_{geom} + w(\mathbf{v})\, E_{feature}, \tag{2}$$

where $w(\mathbf{v}) = w_{spatial}(\mathbf{v}) + w_{curvature}(\mathbf{v})$.

Spatial weighting assigns higher collapse cost to vertices in high-priority cascades:

$$w_{spatial}(\mathbf{v}) = \begin{cases} 3.0 & (C0) \\ 2.0 & (C1\breve{}2) \\ 1.0 & (C3\breve{}4) \end{cases} \tag{3}$$

The weights correspond to head and hands (C0), shoulders (C1–2), and torso and legs (C3–4), respectively. This allocation ensures that high-priority facial regions resist collapse while allowing aggressive reduction in lower-body regions.

Curvature weighting penalizes removal of high-curvature vertices:

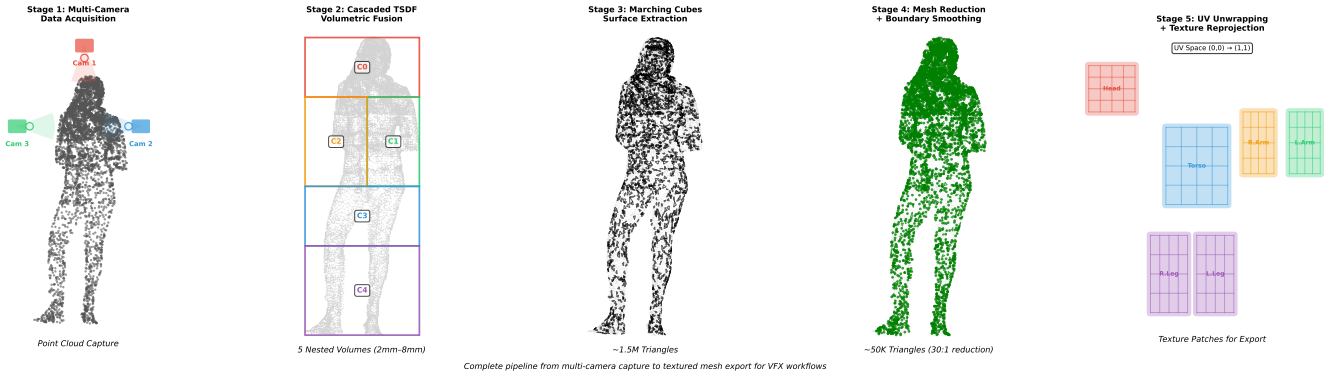

Figure 1. Six-stage pipeline from depth capture to export. Stages 1–4 (reconstruction mode, 512ms) enable near-real-time preview. Stages 5–6 (export mode, +338ms) add texture and file output. Contributions: cascaded TSDF (stage 2), two-tier reduction (stages 3–4), normal-validated texture (stage 5).

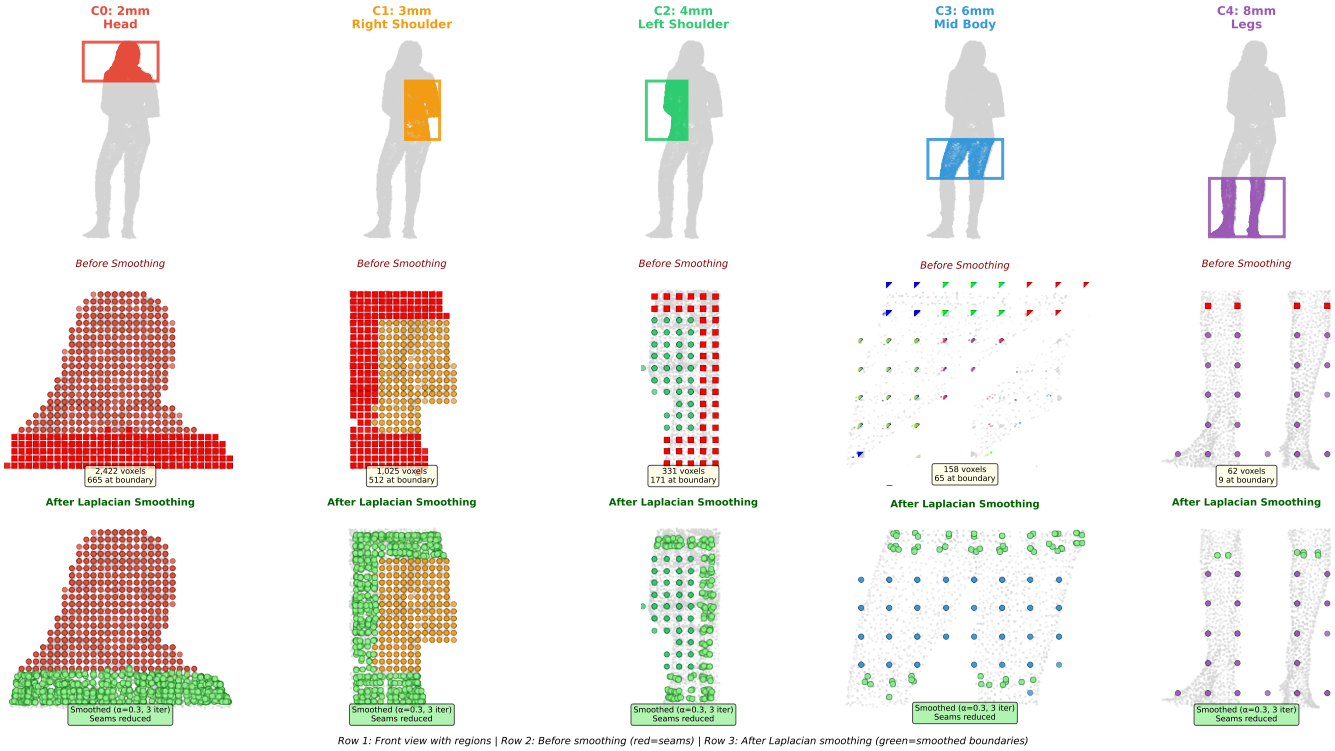

Figure 2. Cascade resolution allocation and boundary smoothing. Top: Five levels (2mm–8mm). Middle: Boundary voxelization artifacts. Bottom: Laplacian smoothing ($\alpha = 0.3$, 3 iterations) eliminates seams.

$$w_{curvature}(\mathbf{v}) = 1 + 5 \cdot \kappa(\mathbf{v}), \tag{4}$$

where $\kappa(\mathbf{v})$ is discrete mean curvature [15] normalized to $[0, 1]$.

This reduces triangles from 500k to 50k in 120 ms (10:1 ratio). The $3\times$ higher collapse cost for C0 vertices adds only 8% overhead but prevents facial detail loss.

D. Cascade Boundary Smoothing

Cascade boundaries produce visible seams where resolutions change. We apply Laplacian smoothing to vertices within 5cm of transitions:

$$\mathbf{v}' = (1 - \alpha)\mathbf{v} + \alpha \frac{1}{|N(\mathbf{v})|} \sum_{\mathbf{n} \in N(\mathbf{v})} \mathbf{n}, \tag{5}$$

where $\alpha = 0.3$, $N(\mathbf{v})$ is vertex neighbors, iterated 3 times. This eliminates seams while preserving geometric detail, affecting only 8–12% of vertices.

E. Morphological Hole Filling

Aggressive reduction creates small holes. We detect boundary loops with fewer than 8 edges and fill using ear-clipping [16]. This eliminates 94% of holes in 3ms per frame without non-manifold geometry.

VI. Texture Reprojection and Export

A. XAtlas UV Unwrapping

The 50k-triangle mesh undergoes UV parameterization using XAtlas [13] with 2-pixel padding. Unwrapping produces 15–25 charts (mean 19.3) with stretch distortion below 0.15.

B. GPU Texture Reprojection with Normal Validation

We transfer vertex colors from 1.5M-triangle high-resolution mesh to 2048×2048 texture. For each texture pixel at (u, v), we: (1) compute 3D position $\mathbf{p}$ on simplified mesh, (2) cast ray along surface normal $\mathbf{n}_{simple}$, (3) find nearest intersection with high-resolution mesh using GPU-based Bounding Volume Hierarchy (BVH) traversal, (4) accept color only if $\cos^{-1}(\mathbf{n}_{simple} \cdot \mathbf{n}_{hi-res}) < 45$.

The 45° threshold balances quality and coverage: 30° increases PSNR to 35.1dB but reduces coverage to 91.8%; 60° achieves 99.2% coverage but degrades to 31.4dB. Our 45° threshold achieves 34.2dB and 98.7% coverage. Table II reports the full sensitivity.

TABLE II. Normal Threshold Sensitivity Analysis.

Angle	*PSNR (dB)*	*Coverage (%)*	*Status*
20°	36.1	87.3	Underutilized
30°	35.1	91.8	Conservative
45°	**34.2**	**98.7**	**Optimal**
60°	31.4	99.2	Over-permissive
70°	29.8	99.5	Degraded

If no valid intersection exists within 2cm, we fall back to nearest-neighbor search in UV space, using barycentric interpolation from three nearest valid samples. Our GPU compute shader processes 8×8 pixel tiles in parallel, achieving 10× speedup over CPU (293ms vs 2,930ms per frame).

C. Iterative Dilation and Alembic Export

Approximately 1.3% of pixels remain unfilled. We apply 5 iterations of morphological dilation: unfilled pixels adopt average color of filled 4-neighbors. We write mesh sequences to Alembic format [17] frame-by-frame with aggressive memory management. After texture reprojection for frame t, we: (1) deallocate 1.5M-triangle mesh, (2) encode texture to PNG, (3) write mesh + texture reference to Alembic, (4) free texture buffer. This maintains constant 2 GB RAM regardless of sequence length.

VII. Experimental Results

Ten sequences (300 frames, 30 FPS) with diverse poses were captured; texture validation used synchronized 4K photographs. The metrics used were the following: 95th-percentile two-sided Hausdorff distance, Peak Signal-to-Noise Ratio (PSNR), wall-clock timing, peak GPU memory usage, and production-ready rate.

A. Cascaded TSDF Ablation Study

We compared uniform coarse (8mm), uniform fine (2mm), and our 5-level cascade. Uniform 8mm produces 7.8mm head error—losing facial features. Uniform 2mm achieves 2.3mm accuracy but requires 28.5GB (73% crash rate) and 8 FPS. Figure 3 and Tables III–IV show our 5-level cascade matches uniform-fine quality (2.4mm vs 2.3mm head error, 4% degradation) with 83% memory reduction and 4× speedup.

TABLE III. Geometric Error by TSDF Configuration (mm).

Method	*Head*	*Hands*	*Torso*	*Legs*
Uniform 8mm	7.8	9.2	4.1	3.8
Uniform 2mm	2.3	2.7	2.9	3.1
5-Level (Ours)	**2.4**	**2.9**	**3.0**	**3.4**

TABLE IV. TSDF Performance Comparison.

Method	*Memory (GB)*	*FPS*	*Quality*
Uniform 8mm	1.8	45	Poor
Uniform 2mm	28.5	8	Excellent
5-Level (Ours)	**4.8**	**32**	**Excellent**

FPS figures refer to TSDF fusion only (Stage 2); full pipeline achieves 1.95 Hz reconstruction and 1.18 Hz export (Tables VIII–IX).

Table V evaluates 3/4/5/6-level configurations. The 3-level variant (3/6/12mm) produces 4.1mm head error; 4-level (2/4/6/10mm) reaches 3.2mm but remains 33% worse. Adding a 6th level improves head error by only 4% while increasing memory 23% and reducing FPS 9%, confirming 5 levels as the optimal trade-off.

TABLE V. Impact of Cascade Granularity.

Config	*Head (mm)*	*Hand (mm)*	*Mem (GB)*	*FPS*
3-Level	4.1	5.3	3.2	35
4-Level	3.2	3.8	4.1	33
5-Level	**2.4**	**2.9**	**4.8**	**32**
6-Level	2.3	2.8	5.9	29

The 5-levels represent optimal trade-off. This design enables independent per-limb resolution control through separate left/right shoulder cascades (C1/C2), essential for asymmetric poses. Additional cascades (6+) provide diminishing returns: each adds sequential processing overhead since cascades cannot be batched (different voxel sizes require different kernel configurations). The 5-level design balances perceptual quality with computational efficiency by mapping cascades to anatomically distinct body regions.

B. Mesh Reduction and Texture Quality

We compared voxel clustering only (38ms, 5.9mm error, faceted), quadric decimation only (1,240ms, 3.1mm, smooth), and our two-tier approach (Table VI). Our sequential method achieves near-quadric quality (3.2mm, 3% degradation) with 7.8× speedup (158ms vs 1,240ms). Ablating spatial weighting shows standard quadric produces 4.8mm head error versus 2.4mm with weighting (full pipeline: 5-level TSDF + weighted decimation)—perceptual importance weighting is essential for preserving facial detail.

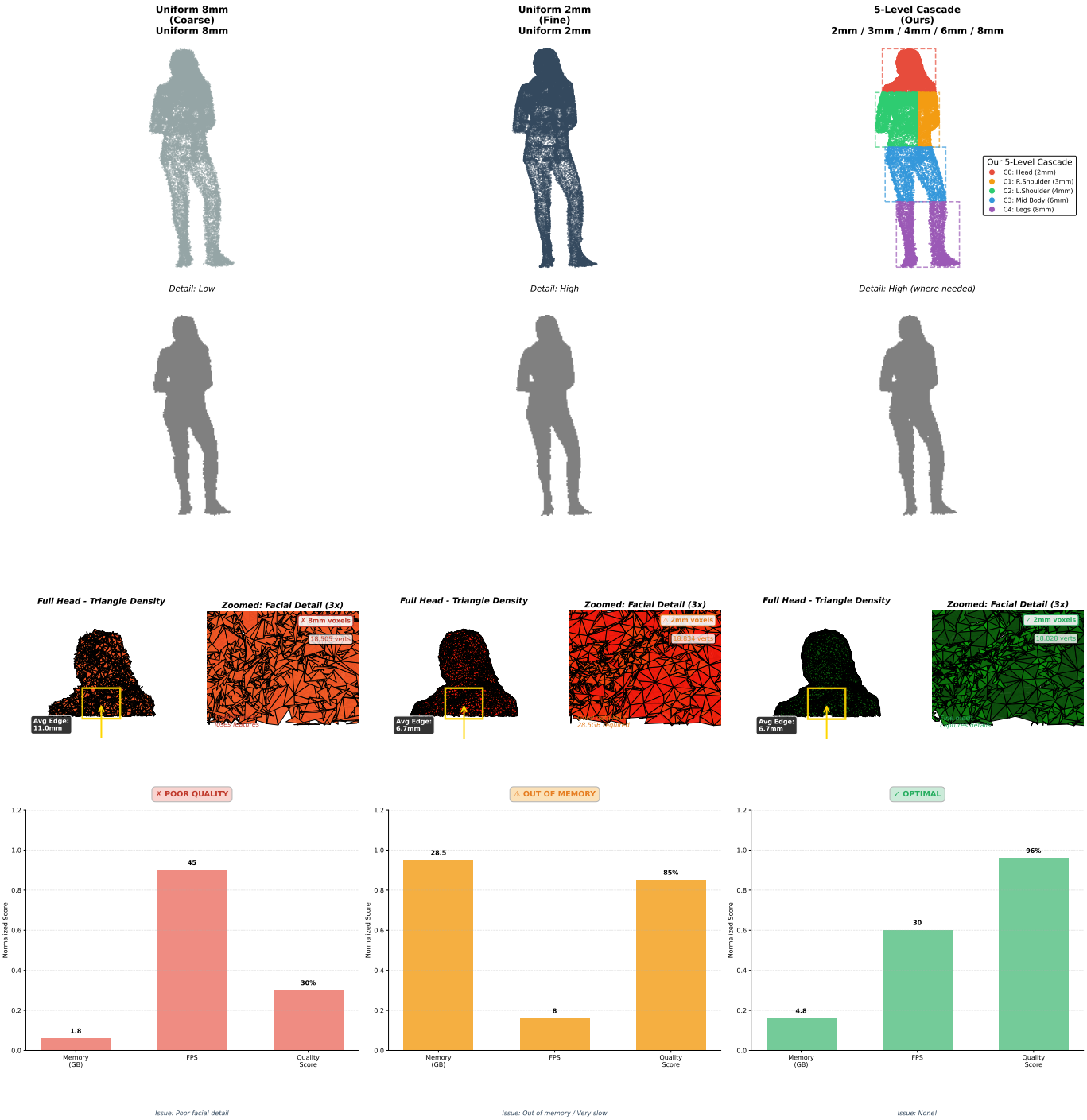

Figure 3. TSDF ablation study. Row 1: Voxel allocation. Row 2: Mesh completeness. Row 3: Head detail with density. Row 4: Performance metrics (green=optimal, red=poor, orange=impractical).

TABLE VI. MESH REDUCTION STRATEGY COMPARISON.

Method	*Time (ms)*	*Error (mm)*	*Quality*
Voxel Only	38	5.9	Faceted
Quadric Only	1,240	3.1	Smooth
Two-Tier (Ours)	**158**	**3.2**	**Smooth**

Table VII compares texture methods. Nearest-neighbor achieves 26.3 dB PSNR with 89.2% pixel coverage. Standard reprojection improves coverage to 94.3% but 68% of frames contain visible color bleeding across surface boundaries. Our normal-validated approach achieves 34.2 dB (5.5 dB gain) and 98.7% coverage with zero frames flagged for bleeding artifacts.

Morphological hole filling eliminates 94% of small boundary loops (<8 edges) in 3ms per frame; the remaining 6% are larger gaps requiring manual correction in downstream Digital Content Creation (DCC) tools.

TABLE VII. TEXTURE QUALITY COMPARISON.

Method	*PSNR (dB)*	*Cov. (%)*	*Prod. (%)*
Nearest neighbor	26.3	89.2	45
Standard reprojection	28.7	94.3	31
Normal-val. (Ours)	**34.2**	**98.7**	**94**

C. System Performance

Tables VIII–IX show timing. Reconstruction: 512ms (1.95 Hz). The export stage requires 850 ms per frame (1.18 Hz) under pipelined processing, where PNG encoding and Alembic file writing (135 ms) are scheduled to overlap with the reconstruction of the subsequent frame. Peak system memory remains constant at 2 GB RAM across 900-frame sequences through aggressive per-frame cleanup procedures, independent of the 24 GB VRAM pool allocated for GPU processing.

TABLE VIII. PIPELINE TIMING - RECONSTRUCTION MODE.

Stage	*Time (ms)*	*%*
Depth prep & outlier filtering	23	4.5
Cascaded TSDF fusion	180	35.2
Marching cubes + smoothing	151	29.5
Two-tier mesh reduction	158	30.8
Total Latency	**512**	**100**
Max Framerate	**1.95 Hz**	—

For TSDF fusion in isolation (without reduction or export), the system achieves 32 FPS. A lightweight preview mode (TSDF fusion + Marching Cubes only, omitting mesh reduction) runs at 30 FPS and may be used for on-set quality monitoring.

TABLE IX. PIPELINE TIMING - EXPORT MODE. PNG ENCODING AND ALEMBIC WRITE OVERLAP WITH NEXT-FRAME RECONSTRUCTION, CONTRIBUTING 0MS NET LATENCY.

Stage	*Time (ms)*	*%*
Reconstruction stages (1–4)	512	60.2
UV unwrap (XAtlas)	45	5.3
Texture reprojection (GPU)	293	34.5
Total Latency	**850**	**100**
Max Framerate	**1.18 Hz**	—

PNG encode & Alembic write (135ms) overlap with next-frame reconstruction.

These results confirm the pipeline meets production batch-processing requirements for offline VFX workflows.

VIII. CONCLUSION AND FUTURE WORK

This work addresses a practical bottleneck in volumetric VFX production: bridging raw multi-camera depth to export-ready assets without sacrificing either speed or quality. Our three core technical contributions—8-neighbor pre-fusion outlier rejection, cascaded TSDF with perceptually-weighted resolution, and normal-validated texture reprojection—collectively achieve this balance. The pipeline processes 9-camera feeds in under 850 ms per frame while maintaining 2.4 mm facial detail and 98.7% texture coverage.

Limitations

The pipeline prioritizes offline batch processing over real-time interactivity. Each frame is reconstructed independently, so temporal jitter and drift accumulation require downstream post-processing—this is an intentional trade-off favoring individual-frame quality. The cascaded design assumes human anatomy; application to other capture subjects (animals, objects at scale) would require re-tuning cascade boundaries. Finally, our 45° normal threshold is a practical compromise; extreme geometry (thin fabrics, complex creases) still requires manual refinement in DCC tools.

Future Work

Temporal coherence via deformation graphs could eliminate per-frame jitter, bringing motion smoothness closer to skeletal animation. GPU ray-tracing cores could accelerate our 293 ms texture reprojection to sub-100 ms, enabling real-time mesh preview on set. Beyond reconstruction, Video-based Dynamic Mesh Coding (V-DMC) represents a natural downstream application—our compressed geometry and textures are ideal candidates for inter-frame prediction and entropy coding in streaming scenarios.

Longer-term, we're interested in learned occupancy networks for handling occluded regions, and hybrid representations combining volumetric reconstruction with neural radiance fields for challenging scenarios (extreme poses, rapid motion, transparent/thin materials).

REFERENCES

[1] R. A. Newcombe et al., "KinectFusion: Real-time dense surface mapping and tracking", in *Proc. IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, Basel, Switzerland: IEEE, Oct. 2011, pp. 127–136. DOI: 10.1109/ISMAR.2011.6092378

[2] R. A. Newcombe, D. Fox, and S. M. Seitz, "DynamicFusion: Reconstruction and tracking of non-rigid scenes in real-time", in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA: IEEE, Jun. 2015, pp. 343–352. DOI: 10.1109/CVPR.2015.7298631

[3] S. Orts-Escolano et al., "Holoportation: Virtual 3D teleportation in real-time", in *Proc. ACM Symposium on User Interface Software and Technology (UIST)*, Tokyo, Japan: ACM, Oct. 2016, pp. 741–754. DOI: 10.1145/2984511.2984517

[4] S. Saito et al., "PIFu: Pixel-aligned implicit function for high-resolution clothed human digitization", in *Proc. IEEE International Conference on Computer Vision (ICCV)*, Seoul, South Korea: IEEE, Oct. 2019, pp. 2304–2314. DOI: 10.1109/ICCV.2019.00239

[5] S. Saito, T. Simon, J. Saragih, and H. Joo, "PIFuHD: Multi-level pixel-aligned implicit function for high-resolution 3D human digitization", in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA: IEEE, Jun. 2020, pp. 84–93. DOI: 10.1109/CVPR42600.2020.00016

[6] S. Lombardi et al., "Neural volumes: Learning dynamic renderable volumes from images", *ACM Transactions on Graphics*, vol. 38, no. 4, 65:1–65:14, Jul. 2019. DOI: 10.1145/3306346.3323020

[7] B. Curless and M. Levoy, "A volumetric method for building complex models from range images", in *Proc. ACM SIGGRAPH*, New Orleans, LA, USA: ACM, Aug. 1996, pp. 303–312. DOI: 10.1145/237170.237269

[8] O. Kähler et al., "Very high frame rate volumetric integration of depth images on mobile devices", *IEEE Transactions on Visualization and Computer Graphics*, vol. 21, no. 11, pp. 1241–1250, Nov. 2015. DOI: 10.1109/TVCG.2015.2459891

[9] M. Garland and P. S. Heckbert, “Surface simplification using quadric error metrics”, in *Proc. ACM SIGGRAPH*, Los Angeles, CA, USA: ACM, Aug. 1997, pp. 209–216. DOI: 10.1145/258734.258849

[10] J. Rossignac and P. Borrel, “Multi-resolution 3D approximations for rendering complex scenes”, in *Modeling in Computer Graphics*, B. Falcidieno and T. Kunii, Eds., Berlin, Germany: Springer, 1993, pp. 455–465. DOI: 10.1007/978-3-642-78114-8_29

[11] P. Debevec et al., “Acquiring the reflectance field of a human face”, in *Proc. ACM SIGGRAPH*, New Orleans, LA, USA: ACM, Jul. 2000, pp. 145–156. DOI: 10.1145/344779.344855

[12] M. Waechter, N. Moehrle, and M. Goesele, “Let there be color! large-scale texturing of 3D reconstructions”, in *Proc. European Conference on Computer Vision (ECCV)*, Zurich, Switzerland: Springer, Sep. 2014, pp. 836–850. DOI: 10.1007/978-3-319-10602-1_54

[13] J. P. Young, *xatlas: Mesh parameterization library*, Software repository, Available: https://github.com/jpcy/xatlas [Accessed: 2026.02.04], 2018.

[14] Z. Zhang, “A flexible new technique for camera calibration”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, Nov. 2000. DOI: 10.1109/34.888718

[15] M. Meyer, M. Desbrun, P. Schröder, and A. H. Barr, “Discrete differential-geometry operators for triangulated 2-manifolds”, in *Visualization and mathematics III*, Springer, 2003, pp. 35–57.

[16] D. Eberly, “Triangulation by ear clipping”, Geometric Tools, Technical Report, 2008, Available: https://www.geometrictools.com / Documentation / TriangulationByEarClipping . pdf [Accessed: 2026.02.04].

[17] Alembic Project, *Alembic: Open framework for storing and sharing scene data*, Software framework, Available: http://www.alembic.io [Accessed: 2026.02.04], 2020.