An Octree-Based Approach towards Efficient Variational Range Data Fusion

Wadim Kehl [1]
kehl@in.tum.de
Tobias Holl [1]
holl@in.tum.de
Federico Tombari [1,2]
tombari@in.tum.de
Slobodan Ilic [1,3]
slobodan.ilic@siemens.com
Nassir Navab [1]
navab@cs.tum.edu

[1] Computer-Aided Medical Procedures,
TU Munich, Germany
[2] Computer Vision Lab (DISI),
University of Bologna, Italy
[3] Siemens AG
Research & Technology Center
Munich, Germany

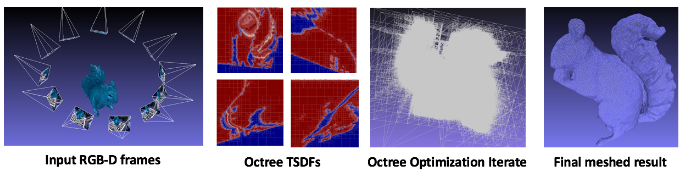

Figure 1: Our work deals with robust variational fusion of range scans. Given a sequence of input frames, we optimize over Octree-structures representing transformed input TSDFs into a common evolving Octree to finally retrieve an accurate and smooth meshed reconstruction.

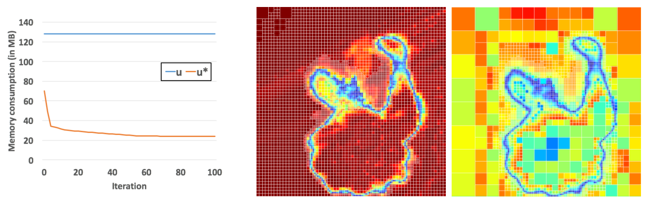

Figure 2: Left: Memory usage of the dense iterate u and Octree u^* during the optimization for the 'head' sequence. The usage goes down quickly for the Octree-variant as the surface evolves in the TSDF, leading to many block joins. Center/Right: Slicing through u^* at iterations 1 and 100.

Volume-based reconstruction is usually expensive both in terms of memory consumption and runtime. Especially for sparse geometric structures, volumetric representations produce a huge computational overhead. We present an efficient way to fuse range data via a variational Octree-based minimization approach by taking the actual range data geometry into account. We transform the data into Octree-based truncated signed distance fields and show how the optimization can be conducted on the newly created structures. The main challenge is to uphold speed and a low memory footprint without sacrificing the solutions' accuracy during optimization. We explain how to dynamically adjust the optimizer's geometric structure via joining/splitting of Octree nodes and how to define the operators. We evaluate on various datasets and outline the suitability in terms of performance and geometric accuracy.

SDF construction The generation of the TSDFs $f_i : \Omega_3 \subset \mathbb{R}^3 \to \mathbb{R}$ follows related work [1, 2, 3]: given the i-th frame of range data $D_i : \Omega_2 \to \mathbb{R}^+$ together with the projection $\pi_i : \mathbb{R}^3 \to \Omega_2$, the idea is to compute the signed distance ϕ_i between the surface and each point $\mathbf{x} \in \Omega_3$ of the reconstruction volume along the line of sight. Furthermore, scaling with δ and truncation to $[-1,+1]$ is performed to retrieve the final TSDF f_i

$$f_i(\mathbf{x}) = \begin{cases} \operatorname{sgn}(\phi_i(\mathbf{x})) & \text{if } |\phi_i(\mathbf{x})| > \delta \\ \phi_i(\mathbf{x})/\delta & \text{else} \end{cases}, \phi_i(\mathbf{x}) = D_i(\pi_i(\mathbf{x})) - ||\mathbf{x} - C_i||. \quad (1)$$

Octree construction We construct TSDF-Octrees f_i^* from f_i in a top-to-bottom manner. Starting from root node n, we define the spread s of values subsumed by node n in f as

$$s_f(n) = \left| \max_{\mathbf{x} \in \Omega_3(n)} f(\mathbf{x}) - \min_{\mathbf{x} \in \Omega_3(n)} f(\mathbf{x}) \right| \quad (2)$$

with $\Omega_3(n)$ being the subvolume that node n represents. Initially, $\Omega_3(n) = \Omega_3$ and the spread will be maximal. From here we recursively apply a splitting rule: if the spread $s_f(n)$ at node n is higher than a threshold $\tau = 0.1$, we subdivide n into eight children and proceed further down.

Variational Octree optimization We define a strictly-convex functional based on our Octree-representation

$$\mathcal{E}(u^*) := \int_{\Omega_3} \frac{D(\mathbf{f}^*, \mathbf{w}^*, u^*)}{\sum_i w_i^* + \gamma} + \lambda S(\nabla u^*) \, d\mathbf{x}, \quad (3)$$

and solve for u^* with a small γ in the normalizer to avoid division problems for unseen voxels. To optimize Equation 3, we determine the steady state of our PDE with a constantly evolving Octree u^*

$$\frac{\partial \mathcal{E}}{u^*} = \lambda \operatorname{div}(S_{\nabla u^*}(\nabla u^*)) - \frac{D_{u^*}(\mathbf{f}^*, \mathbf{w}^*, u^*)}{\sum_i w_i^* + \gamma}. \quad (4)$$

We conduct the optimization by having at all times only one version of u^* in memory and adjusting the structure while we recursively traverse into each node of u^*. This means that instead of integrating point-wise over the volume, we start from the root and run along the tree while conducting our computations/restructuring on it before proceeding to the next node in the volume in the same pass. Furthermore, we compute the gradient and the divergence in a manner that the induced error from the Octree partitioning has a negligible effect on the numerical update.

Results For all employed sequences (synthetic, low-cost RGB-D sensor, high-precision depth sensor) we constantly retrieve very accurate solutions that are on par with their dense versions in terms of metric fidelity (Figure 3). Furthermore, the total amount of memory needed is drastically reduced with the Octree approach. To give another interesting insight we plot the memory consumption of u^* during the optimization in Figure 2. While in the dense approach each iterate u_t is constant in memory, its Octree-variant quickly decreases its memory footprint after more and more blocks get joined.

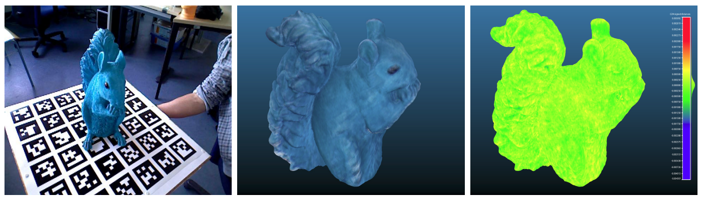

Figure 3: Example frame from one sequence, meshed result and reconstruction difference between the dense and the Octree version.

[1] Wadim Kehl, Nassir Navab, and Slobodan Ilic. Coloured signed distance fields for full 3D object reconstruction. In *BMVC*, 2014.

[2] Christopher Schroers, Henning Zimmer, Levi Valgaerts, Oliver Demetz, and Joachim Weickert. Anisotropic Range Image Integration. *Pattern Recognition, LNCS*, 2012.

[3] Christopher Zach, Thomas Pock, and Horst Bischof. A globally optimal algorithm for robust TV-L1 range image integration. In *ICCV*, 2007.