Variational Level Set Evolution for Non-rigid 3D Reconstruction from a Single Depth Camera
Supplementary Material

Miroslava Slavcheva [1,2], Maximilian Baust [1], Slobodan Ilic [1,2]

[1] Technical University of Munich, Germany
[2] Siemens Corporate Technology, Munich, Germany

This supplementary document contains complete derivations of the components of our non-rigid registration model. In addition, we provide details on how to compute separable 1D Sobolev filters.

1 Signed Distance Field Evolution Energy Components

Here we provide the derivatives of the energy terms following standard calculus of variations.

We are seeking a 3D vector field $\Psi = (U, V, W)$, where U, V and W are its x-, y- and z-components respectively, each of which is a scalar grid $\mathbb{N}^3 \mapsto \mathbb{R}$. Voxel coordinates are denoted by (x, y, z) and the respective flow vector applied at a voxel is denoted by (u, v, w).

1.1 Data term

The data term aligns the projective TSDF ϕ_{proj} of the current frame with the cumulative TSDF ϕ_{model}, driving their voxel-wise difference to be minimal:

$$E_{data}(\Psi) = \frac{1}{2} \sum_{x,y,z} \big(\phi_{proj}(x+u, y+v, z+w) - \phi_{model}(x, y, z)\big)^2 . \tag{1}$$

$$\begin{aligned}
\frac{\partial E_{data}}{\partial u} &= \frac{1}{2}\left[\frac{\partial \big(\phi_{proj}(x+u, y+v, z+w) - \phi_{model}(x, y, z)\big)^2}{\partial u} - \mathrm{div}\frac{\partial \big(\phi_{proj}(x+u, y+v, z+w) - \phi_{model}(x, y, z)\big)^2}{\partial \nabla u}\right] = \\
&= \frac{1}{2}\frac{\partial \big(\phi_{proj}(x+u, y+v, z+w) - \phi_{model}(x, y, z)\big)^2}{\partial u} = \\
&= \frac{1}{2}\, 2\, \big(\phi_{proj}(x+u, y+v, z+w) - \phi_{model}(x, y, z)\big)\frac{\partial \big(\phi_{proj}(x+u, y+v, z+w) - \phi_{model}(x, y, z)\big)}{\partial u} = \\
&= \big(\phi_{proj}(x+u, y+v, z+w) - \phi_{model}(x, y, z)\big)\frac{\partial \phi_{proj}(x+u, y+v, z+w)}{\partial u} = \\
&= \big(\phi_{proj}(x+u, y+v, z+w) - \phi_{model}(x, y, z)\big)\, \nabla_x \phi_{proj}(x+u, y+v, z+w)
\end{aligned} \tag{2}$$

Above $\nabla_x \phi$ denotes the x-component of the spatial gradient of the TSDF ϕ, which is obtained numerically via central differences. We will use analogous notation for the y- and z-components. The full TSDF gradient is therefore written as $\nabla \phi = (\nabla_x \phi, \nabla_y \phi, \nabla_z \phi)^\top$.

We also use the nabla symbol ∇ to denote energy derivatives. Thus:

$$\begin{aligned}
\nabla E_{data}(\Psi) &= \big(\phi_{proj}(x+u, y+v, z+w) - \phi_{model}(x, y, z)\big) \begin{pmatrix} \nabla_x \phi_{proj}(x+u, y+v, z+w) \\ \nabla_y \phi_{proj}(x+u, y+v, z+w) \\ \nabla_z \phi_{proj}(x+u, y+v, z+w) \end{pmatrix} = \\
&= \big(\phi_{proj}(x+u, y+v, z+w) - \phi_{model}(x, y, z)\big)\, \nabla \phi_{proj}(x+u, y+v, z+w) = \\
&= \big(\phi_{proj}(\Psi) - \phi_{model}\big)\, \nabla \phi_{proj}(\Psi)
\end{aligned} \tag{3}$$

We use $\phi_{proj}(\Psi)$ to refer to the evolved TSDF after the application of the warp field vector (u, v, w), *i.e.* equivalently to $\phi_{proj}(x+u, y+v, z+w)$. We will use this shorthand notation from here onwards.

1.2 Uniform motion term

The term which encourages nearby vectors to be similar is the Tikhonov-type regularizer:

$$E_{smooth}(\Psi) = \frac{1}{2}\sum_{x,y,z}\left(|\nabla U(x,y,z)|^2 + \ + |\nabla V(x,y,z)|^2 + |\nabla W(x,y,z)|^2\right). \tag{4}$$

$$\begin{aligned}\frac{\partial E_{smooth}}{\partial u} &= \frac{1}{2}\Bigg[\frac{\partial\left(|\nabla U(x,y,z)|^2 + |\nabla V(x,y,z)|^2 + |\nabla W(x,y,z)|^2\right)}{\partial u} - \\ &- \operatorname{div}\frac{\partial\left(|\nabla U(x,y,z)|^2 + |\nabla V(x,y,z)|^2 + |\nabla W(x,y,z)|^2\right)}{\partial \nabla u}\Bigg] = \\ &= \frac{1}{2}\Bigg[0 - \operatorname{div}\frac{\partial\left(|\nabla U(x,y,z)|^2 + |\nabla V(x,y,z)|^2 + |\nabla W(x,y,z)|^2\right)}{\partial \nabla u}\Bigg] = \\ &= \ -\frac{1}{2}\operatorname{div}\frac{\partial|\nabla U(x,y,z)|^2}{\partial \nabla u} = -\frac{1}{2}\operatorname{div}2\nabla U(x,y,z) = -\operatorname{div}\nabla U = -\Delta U\end{aligned} \tag{5}$$

The symbol Δ denotes the Laplacian of its operand. Thus:

$$\nabla E_{smooth}(\Psi) = -(\Delta U, \Delta V, \Delta W)^\top \tag{6}$$

1.3 Approximately Killing vector field term

The approximately Killing vector field term (AKVF) enforces the warp field to be divergence free by minimizing the Frobenius norm of the Killing condition:

$$E_{akvf}(\Psi) = \frac{1}{2}\sum_{x,y,z}\left\|J_\Psi + J_\Psi^\top\right\|_F^2. \tag{7}$$

The Jacobian of the vector field is: $J_\Psi = \begin{pmatrix}\partial U/\partial x & \partial U/\partial y & \partial U/\partial z \\ \partial V/\partial x & \partial V/\partial y & \partial V/\partial z \\ \partial W/\partial x & \partial W/\partial y & \partial W/\partial z\end{pmatrix} = \begin{pmatrix}U_x & U_y & U_z \\ V_x & V_y & V_z \\ W_x & W_y & W_z\end{pmatrix}$ and its transpose is denoted by $J_\Psi^\top$.

Next, let us rewrite Eq. (7) using the column-wise stacking operator $vec(A)$, which denotes the vectorized matrix A. Thus, $vec(J_\Psi) \in \mathbb{R}^{9\times 1}$ is the 9-element vector of stacked elements from J_Ψ, and similarly $vec(J_\Psi^\top) \in \mathbb{R}^{9\times 1}$ contains the elements from $J_\Psi^\top$. Finally, $vec(J_\Psi)^\top \in \mathbb{R}^{1\times 9}$ denotes the transpose of $vec(J_\Psi)$.

$$vec(J_\Psi) = \begin{pmatrix}U_x & V_x & W_x & U_y & V_y & W_y & U_z & V_z & W_z\end{pmatrix}^\top \tag{8}$$

$$\begin{aligned}E_{akvf}(\Psi) &= \frac{1}{2}\sum_{x,y,z}\left\|\begin{pmatrix}2U_x & V_x+U_y & W_x+U_z \\ V_x+U_y & 2V_y & W_y+V_z \\ W_x+U_z & W_y+V_z & 2W_z\end{pmatrix}\right\|_F^2 = \\ &= \frac{1}{2}\sum_{x,y,z} vec(J_\Psi + J_\Psi^\top)^\top vec(J_\Psi + J_\Psi^\top) = \\ &= \frac{1}{2}\sum_{x,y,z}\left(vec(J_\Psi)^\top vec(J_\Psi) + 2vec(J_\Psi^\top)^\top vec(J_\Psi) + vec(J_\Psi^\top)^\top vec(J_\Psi^\top)\right) = \\ &= \sum_{x,y,z} vec(J_\Psi)^\top vec(J_\Psi) + vec(J_\Psi^\top)^\top vec(J_\Psi) = \\ &= \sum_{x,y,z}\left(2U_x^2 + 2V_y^2 + 2W_z^2 + U_y^2 + U_z^2 + V_x^2 + V_z^2 + W_x^2 + W_y^2 + 2V_xU_y + 2W_xU_z + 2W_yV_z\right)\end{aligned} \tag{9}$$

$$
\begin{aligned}
\frac{\partial E_{akvf}}{\partial u} &= \frac{\partial(2U_x^2+2V_y^2+2W_z^2+U_y^2+U_z^2+V_x^2+V_z^2+W_x^2+W_y^2+2V_xU_y+2W_xU_z+2W_yV_z)}{\partial u} - \\
&- \frac{\partial}{\partial x}\frac{\partial(2U_x^2+2V_y^2+2W_z^2+U_y^2+U_z^2+V_x^2+V_z^2+W_x^2+W_y^2+2V_xU_y+2W_xU_z+2W_yV_z)}{\partial U_x} - \\
&- \frac{\partial}{\partial y}\frac{\partial(2U_x^2+2V_y^2+2W_z^2+U_y^2+U_z^2+V_x^2+V_z^2+W_x^2+W_y^2+2V_xU_y+2W_xU_z+2W_yV_z)}{\partial U_y} - \\
&- \frac{\partial}{\partial z}\frac{\partial(2U_x^2+2V_y^2+2W_z^2+U_y^2+U_z^2+V_x^2+V_z^2+W_x^2+W_y^2+2V_xU_y+2W_xU_z+2W_yV_z)}{\partial U_z} = \\
&= 0 - \frac{\partial}{\partial x}(4U_x) - \frac{\partial}{\partial y}(2U_y+2V_x) - \frac{\partial}{\partial z}(2U_z+2W_x) = \\
&= -4U_{xx} - (2U_{yy}+2V_{xy}) - (2U_{zz}+2W_{xz}) = -2(2U_{xx}+U_{yy}+U_{zz}+V_{xy}+W_{xz})
\end{aligned}
\tag{10}
$$

Similarly:

$$
\begin{aligned}
\frac{\partial E_{akvf}}{\partial v} &= -2(V_{xx}+2V_{yy}+V_{zz}+U_{xy}+W_{yz}) \\
\frac{\partial E_{akvf}}{\partial w} &= -2(W_{xx}+W_{yy}+2W_{zz}+U_{xz}+V_{yz})
\end{aligned}
\tag{11}
$$

Finally,

$$
\begin{aligned}
\nabla E_{akvf}(\Psi) &= -2\begin{pmatrix} 2U_{xx}+U_{yy}+U_{zz}+V_{xy}+W_{xz} \\ V_{xx}+2V_{yy}+V_{zz}+U_{xy}+W_{yz} \\ W_{xx}+W_{yy}+2W_{zz}+U_{xz}+V_{yz} \end{pmatrix} = \\
&= -2\begin{pmatrix} U_{xx}+U_{yy}+U_{zz} \\ V_{xx}+V_{yy}+V_{zz} \\ W_{xx}+W_{yy}+W_{zz} \end{pmatrix} - 2\begin{pmatrix} U_{xx}+V_{xy}+W_{xz} \\ U_{xy}+V_{yy}+W_{yz} \\ U_{xz}+V_{yz}+W_{zz} \end{pmatrix} = \\
&= -2\begin{pmatrix} \Delta U \\ \Delta V \\ \Delta W \end{pmatrix} - 2\begin{pmatrix} \partial(\mathrm{div}\Psi)/\partial x \\ \partial(\mathrm{div}\Psi)/\partial y \\ \partial(\mathrm{div}\Psi)/\partial z \end{pmatrix},
\end{aligned}
\tag{12}
$$

where $\mathrm{div}\Psi = U_x + V_y + W_z$ is the divergence of the warp field Ψ.

1.4 Damped Killing term

As discussed, the condition from Eq. (7) is too strong to account for large deformations. Re-writing the first term from the vectorized form in Eq. (9) sum leads to:

$$
\begin{aligned}
\sum_{x,y,z} vec(J_\Psi)^\top vec(J_\Psi) &= \sum_{x,y,z} \left(U_x^2+U_y^2+U_x^2+V_x^2+V_y^2+V_z^2+W_x^2+W_y^2+W_z^2\right) = \\
&= \sum_{x,y,z} \left(|\nabla U|^2+|\nabla V|^2+|\nabla W|^2\right) = E_{smooth}(\Psi)
\end{aligned}
\tag{13}
$$

Thus increasing the weight of the motion smoothness component and decreasing the weight of the rigidity component leads to the damped Killing condition:

$$
E_{Killing}(\Psi) = \sum_{x,y,z} \left(vec(J_\Psi)^\top vec(J_\Psi) + \gamma vec(J_\Psi^\top)^\top vec(J_\Psi)\right). \tag{14}
$$

The factor γ controls the balance between the strictly rigid and non-rigid components of the regularization. A choice of $\gamma = 1$ would lead to the AKVF condition from the previous section. As we aim to alleviate the effect of the rigidity constraint, we use values of $\gamma < 1$ in our optimization. The combined functional derivative is then:

$$
\nabla E_{Killing}(\Psi) = -2(\Delta U, \Delta V, \Delta W)^\top - 2\gamma\left(\frac{\partial}{\partial x}(\mathrm{div}\Psi), \frac{\partial}{\partial y}(\mathrm{div}\Psi), \frac{\partial}{\partial z}(\mathrm{div}\Psi)\right)^\top. \tag{15}
$$

1.5 Level set term

Maintaining the property of unity gradient ensures geometrically correct TSDF evolution:

$$E_{level\,set}(\Psi) = \frac{1}{2}\sum_{x,y,z}\left(|\nabla\phi_{proj}(x+u,y+v,z+w)|-1\right)^2. \tag{16}$$

Note that when the implementation is over a truncated signed distance field, the gradient magnitude is unit in the narrow band and 0 in the truncated ± 1 regions. If the TSDF is also scaled, the scale δ has to be applied also to the unity in the narrow band. Furthermore, values on the border between truncated and non-truncated region will be between 0 and $1/\delta$, so additional care has to be taken there.

The functional derivative is then:

$$\begin{aligned}
\frac{\partial E_{level\,set}}{\partial u} &= \frac{1}{2}\left[\frac{\partial\left(|\nabla\phi_{proj}(x+u,y+v,z+w)|-1\right)^2}{\partial u} - \mathrm{div}\frac{\partial\left(|\nabla\phi_{proj}(x+u,y+v,z+w)|-1\right)^2}{\partial\nabla u}\right] = \\
&= \frac{1}{2}\frac{\partial\left(|\nabla\phi_{proj}(x+u,y+v,z+w)|-1\right)^2}{\partial u} = \\
&= \frac{1}{2}\,2\,\left(|\nabla\phi_{proj}(x+u,y+v,z+w)|-1\right)\frac{\partial\left(|\nabla\phi_{proj}(x+u,y+v,z+w)|-1\right)}{\partial u} = \\
&= \left(|\nabla\phi_{proj}(\Psi)|-1\right)\frac{\partial\left(\left(\frac{\partial\phi_{proj}(\Psi)}{\partial x}\right)^2+\left(\frac{\partial\phi_{proj}(\Psi)}{\partial y}\right)^2+\left(\frac{\partial\phi_{proj}(\Psi)}{\partial z}\right)^2\right)^{1/2}}{\partial u} = \\
&= \frac{|\nabla\phi_{proj}(\Psi)|-1}{2|\nabla\phi_{proj}(\Psi)|_\varepsilon}\left(2\frac{\partial\phi_{proj}(\Psi)}{\partial x}\frac{\partial}{\partial u}\frac{\partial\phi_{proj}(\Psi)}{\partial x}+2\frac{\partial\phi_{proj}(\Psi)}{\partial y}\frac{\partial}{\partial u}\frac{\partial\phi_{proj}(\Psi)}{\partial y}+2\frac{\partial\phi_{proj}(\Psi)}{\partial z}\frac{\partial}{\partial u}\frac{\partial\phi_{proj}(\Psi)}{\partial z}\right) = \\
&= \frac{|\nabla\phi_{proj}(\Psi)|-1}{|\nabla\phi_{proj}(\Psi)|_\varepsilon}\left(\nabla_x\phi_{proj}(\Psi)\nabla_{xx}\phi_{proj}(\Psi)+\nabla_y\phi_{proj}(\Psi)\nabla_{xy}\phi_{proj}(\Psi)+\nabla_z\phi_{proj}(\Psi)\nabla_{xz}\phi_{proj}(\Psi)\right) = \\
&= \frac{|\nabla\phi_{proj}(\Psi)|-1}{|\nabla\phi_{proj}(\Psi)|_\varepsilon}\begin{pmatrix}\nabla_{xx}\phi_{proj}(\Psi) & \nabla_{xy}\phi_{proj}(\Psi) & \nabla_{xz}\phi_{proj}(\Psi)\end{pmatrix}\nabla\phi_{proj}(\Psi),
\end{aligned} \tag{17}$$

where $|\cdot|_\epsilon$ denotes the norm plus a small constant ϵ which avoids division by zero. Similarly we obtain:

$$\nabla E_{\substack{level\\set}}(\Psi) = \frac{|\nabla\phi_{proj}(\Psi)|-1}{|\nabla\phi_{proj}(\Psi)|_\varepsilon}\begin{pmatrix}\nabla_{xx}\phi_{proj}(\Psi) & \nabla_{xy}\phi_{proj}(\Psi) & \nabla_{xz}\phi_{proj}(\Psi)\\ \nabla_{yx}\phi_{proj}(\Psi) & \nabla_{yy}\phi_{proj}(\Psi) & \nabla_{yz}\phi_{proj}(\Psi)\\ \nabla_{zx}\phi_{proj}(\Psi) & \nabla_{zy}\phi_{proj}(\Psi) & \nabla_{zz}\phi_{proj}(\Psi)\end{pmatrix}\nabla\phi_{proj}(\Psi), \tag{18}$$

where the 3×3 matrix in the middle is the Hessian $H_{\phi_{proj}(\Psi)}$ of the warped TSDF.

2 Sobolev Kernels

Here we explain how to obtain the three separable 1D filters starting with from the following equation from the paper:

$$(Id - \lambda\Delta)S = v\,. \tag{19}$$

Let the size of the 3D Sobolev filter we are interested in be $s \times s \times s$. Then the terms in the above equation are as follows:

- Id is the $s^3 \times s^3$ identity matrix.
- Δ is the s-point stencil finite difference Laplacian matrix describing neighbouring voxels, resulting in the occupancy shown in Figure 1.
- v is a one-hot s^3-element vector with 1 at the middle index $\left\lfloor \frac{s^3}{2} \right\rfloor$ (assuming indexing starting at 0). It corresponds to a discretized Dirac impulse of size $s \times s \times s$ voxels.
- S is the s^3-element solution of the linear system that we are looking for. By restructuring it into a $s \times s \times s$ volume, we obtain the sought 3D Sobolev filter.

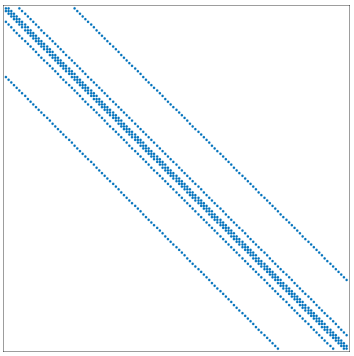

Figure 1: Occupancy of a $s^3 \times s^3$ matrix Δ.

In order to obtain the corresponding 1D filters, we make an approximation using the higher-order SVD decomposition of the tensor S. It yields three $s \times s$ U-matrices with equal elements. We take the first singular vector from each of these matrices, obtaining the approximated 1D filters S_x, S_y and S_z. Note that they have equal entries, but we use the subscript to indicate the spatial direction in which they are applied.

This procedure needs to be done only once for selected neighbourhood size s and Sobolev parameter λ, after which the 1D filter entries can be stored. The separable convolutions are then applied over the energy derivative in each gradient descent step.

3 Accelerated Optimization

3.1 Theory

As defined in the paper, we have an action integral J_{def} that consists of a kinetic energy K_{def} and a potential energy equal to the chosen deformation energy E_{def}:

$$K_{def}(\Psi) = \frac{1}{2} \sum_{x,y,z} \left(\rho(\Psi, \nabla\Psi)\, \Psi_t^2 \right), \tag{20}$$

$$J_{def}(\Psi) = \int k(t) \left(K_{def}(\Psi) - b(t) E_{def}(\Psi) \right) \, dt \,, \tag{21}$$

with $\rho(\Psi, \nabla\Psi)$ being the mass density and $a(t)$, $b(t)$ time-dependent weights. Note that ρ is a scalar field, *i.e.* in general settings it also depends on the voxel location, but we have omitted this for consistency with the notation used in Section 4 of the paper (and especially the data term and its derivative).

While ρ could be chosen as the Dirac delta function in case the optimization variable is the zero level set of a distance field [1], here we are estimating a warp field Ψ. As is customary in the case of diffeomorphisms, we set ρ to a constant $\rho_0 \in \mathbb{R}$ throughout the entire volume. Thus ∇K_{def} vanishes:

$$\frac{\partial K_{def}}{\partial \Psi} = \frac{1}{2} \left[\frac{\partial \left(\rho(\Psi, \nabla\Psi)\, \Psi_t^2 \right)}{\partial \Psi} - \mathrm{div} \frac{\partial \left(\rho(\Psi, \nabla\Psi)\, \Psi_t^2 \right)}{\partial \nabla\Psi} \right] = \frac{1}{2} \left[\Psi_t^2 \frac{\partial \rho_0}{\partial \Psi} - div \left(\Psi_t^2 \frac{\partial \rho_0}{\partial \nabla\Psi} \right) \right] = 0 \,. \tag{22}$$

The equations of motion for Ψ are the Euler-Lagrange equations for J_{def}. We refer the reader to [1] (*cf.* eq. 12) for a complete derivation that yields the following:

$$\frac{d}{dt} \left(\rho \Psi_t \right) + a(t) \rho \Psi_t = \nabla K_{def}(\Psi) - b(t) \nabla E_{def}(\Psi) \,, \tag{23}$$

which becomes the following in our case:

$$\Psi_{tt} + a(t) \Psi_t = -\frac{b(t)}{\rho_0} \nabla E_{def}(\Psi) \,. \tag{24}$$

3.2 Parameter Analysis

We carried out our investigation of the $b(t)$ parameter on the relatively small-motion *Andrew-Chair* sequence from the paper of Dou *et al.* [2] and on the large-motion *Alex* sequence from KillingFusion. Table 1 summarizes the average number of iterations required for $E_{accelerated}$ to converge, *i.e.* for the energy update to fall below 10^{-6} (we measure the energy as SSD divided by the total number of voxels). Likewise, Figure 2 shows the progression of number of required iterations during the sequences. Note how the iteration number is consistent throughout the *Andrew-Chair* sequence, while the large motion of the arms can be noticed as more iterations are needed at the end of the *Alex* one.

Table 1: Iterations to convergence with accelerated optimization depending on $b(t)$.

	$\rho_0 = 2$	$\rho_0 = 1$	$\rho_0 = 1/2$	$\rho_0 = 1/3$	$\rho_0 = 1/4$	$\rho_0 = 1/5$	$\rho_0 = 1/6$	$\rho_0 = 1/7$	$\rho_0 = 1/8$	$\rho_0 = 1/9$	$\rho_0 = 1/10$
Andrew	37.1	31.5	23.4	19.6	17.5	16.1	15.0	14.7	14.4	13.6	13.6
Alex	49.7	45.5	36.2	31.4	28.2	25.6	23.6	22.2	21.1	20.1	19.5

3.3 Accelerated Optimization Experiments

Figure 3 shows the CloudCompare error plots for the objects from the *Deformable 3D Reconstruction Dataset*.

Figure 4 compares the results with voxel size 6 mm and 1 cm. The larger voxel size may lead to some washing out of details, such as the nose.

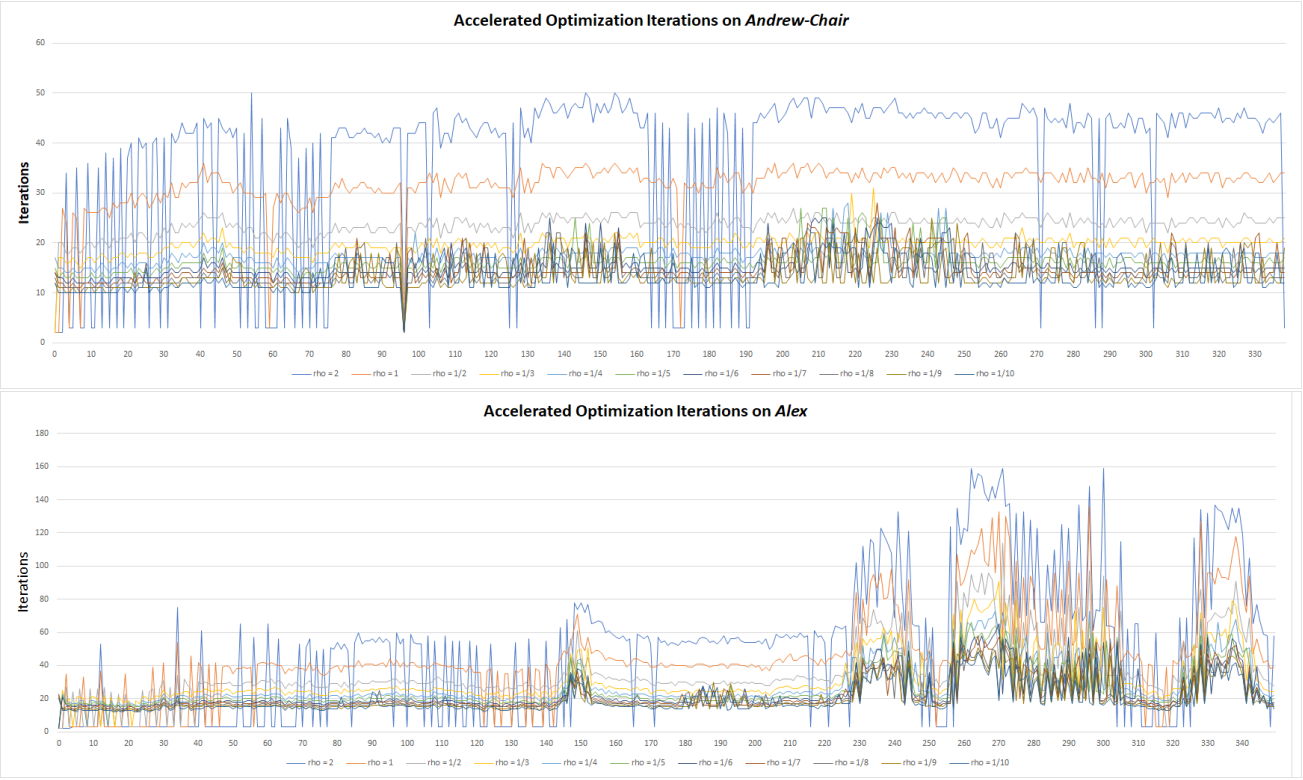

Figure 2: Accelerated optimization iterations to converge throughout the *Andrew-Chair* and *Alex* sequences for different values of $b(t)$.

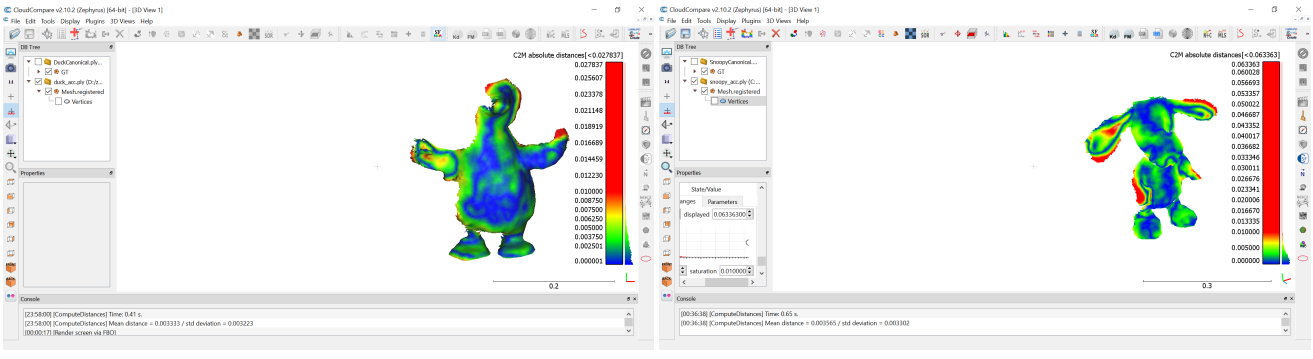

Figure 3: *CloudCompare* cloud-to-mesh error evaluation for the objects from the *Deformable 3D Reconstruction Dataset*. Red is saturated at 1 cm.

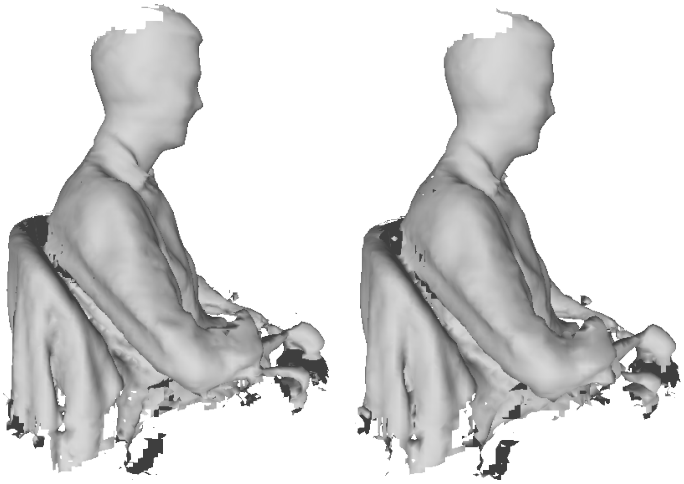

Figure 4: Comparison of the effect of voxel size: 6 mm (left) versus 1 cm (right). Although the results are similar, the larger voxel size washes out some fine-scale details such as the nose.

References

[1] J. Calder and A. Yezzi. PDE Acceleration: A Convergence Rate Analysis and Applications to Obstacle Problems. *arXiv preprint arXiv:1810.01066*, 2018.

[2] M. Dou, J. Taylor, H. Fuchs, A. Fitzgibbon, and S. Izadi. 3D Scanning Deformable Objects with a Single RGBD Sensor. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.