

# KillingFusion: Non-rigid 3D Reconstruction without Correspondences

## Supplementary Material

Miroslava Slavcheva<sup>1,2</sup> Maximilian Baust<sup>1</sup> Daniel Cremers<sup>1</sup> Slobodan Ilic<sup>1,2</sup>  
{mira.slavcheva,maximilian.baust,cremers}@tum.de, slobodan.ilic@siemens.com

<sup>1</sup> Technische Universität München, Munich, Germany

<sup>2</sup> Siemens Corporate Technology, Munich, Germany

This supplementary document gives a more detailed derivation of our non-rigid reconstruction model in Sections 1 and 2. Then it provides more insights into the properties of approximately Killing vector fields in Section 3. Finally, further results are included in Section 4.

### 1. Non-rigid Registration Energy

Recall that we are estimating a 3D vector field  $\Psi: \mathbb{N}^3 \mapsto \mathbb{R}^3$  and our energy is formulated as an SDF-based data alignment term, regularized by motion smoothness and rigidity, and a level set property preservation term:

$$E_{rigid}^{non}(\Psi) = E_{data}(\Psi) + \omega_k E_{Killing}(\Psi) + \omega_s E_{level}^{set}(\Psi). \quad (1)$$

#### 1.1. Data Term

The data term aligns the projective SDF  $\phi_n$  of frame number  $n$  with the cumulative SDF  $\phi_{global}$ , driving their voxel-wise difference to be minimal:

$$E_{data}(\Psi) = \frac{1}{2} \sum_{x,y,z} (\phi_n(x+u, y+v, z+w) - \phi_{global}(x, y, z))^2. \quad (2)$$

Note that in the above formula, and elsewhere, the dependence of  $u$ ,  $v$  and  $w$  on the voxel location  $(x, y, z)$  and on the frame number  $n$  has been omitted for brevity.

#### 1.2. Level Set Property

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

$$E_{level}^{set}(\Psi) = \frac{1}{2} \sum_{x,y,z} (|\nabla \phi_n(x+u, y+v, z+w)| - 1)^2. \quad (3)$$

Here  $\nabla \phi$  denotes the spatial gradient of the SDF  $\phi$ .

Note that when the implementation is over a truncated signed distance field, the gradient magnitude is unity in the narrow band and 0 in the truncated  $\pm 1$  regions. We do not write this explicitly in the equations, since the deformation field is calculated over the narrow band only.

#### 1.3. Motion Regularization

Let us denote the Jacobian of the vector field  $\Psi$  as  $J_\Psi = \begin{pmatrix} \partial u/\partial x & \partial u/\partial y & \partial u/\partial z \\ \partial v/\partial x & \partial v/\partial y & \partial v/\partial z \\ \partial w/\partial x & \partial w/\partial y & \partial w/\partial z \end{pmatrix}$ . Its transpose is  $J_\Psi^\top$ .

As explained in the main paper, a Killing vector field generates isometric motion and satisfies the Killing condition  $\|J_\Psi + J_\Psi^\top\|_F^2 = 0$ . An approximately Killing vector field (AKVF) generates locally nearly isometric motion, thus balancing volume and angular distortion, and minimizes the Frobenius norm of the Killing condition:

$$E_{AKVF}(\Psi) = \frac{1}{2} \sum_{x,y,z} \|J_\Psi + J_\Psi^\top\|_F^2. \quad (4)$$

Next, let us rewrite Eq. 4 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  $J_\Psi$ , 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  $vec(J_\Psi)$ .

$$vec(J_\Psi) = (u_x \ v_x \ w_x \ u_y \ v_y \ w_y \ u_z \ v_z \ w_z)^\top \quad (5)$$

We obtain the following:

$$\begin{aligned} E_{AKVF}(\Psi) &= \frac{1}{2} \sum_{x,y,z} vec(J_\Psi + J_\Psi^\top)^\top vec(J_\Psi + J_\Psi^\top) = \\ &= \frac{1}{2} \sum_{x,y,z} \left( vec(J_\Psi)^\top vec(J_\Psi) + 2vec(J_\Psi^\top)^\top vec(J_\Psi) + vec(J_\Psi^\top)^\top vec(J_\Psi^\top) \right) = \\ &= \sum_{x,y,z} vec(J_\Psi)^\top vec(J_\Psi) + vec(J_\Psi^\top)^\top vec(J_\Psi). \end{aligned} \quad (6)$$

However, this condition is too restrictive for cases of large deformation. We notice that the first term of Eq. 6 can be rewritten as follows:

$$\begin{aligned} \sum_{x,y,z} vec(J_\Psi)^\top vec(J_\Psi) &= \sum_{x,y,z} (u_x^2 + u_y^2 + u_z^2 + v_x^2 + v_y^2 + v_z^2 + w_x^2 + w_y^2 + w_z^2) = \\ &= \sum_{x,y,z} (|\nabla u|^2 + |\nabla v|^2 + |\nabla w|^2), \end{aligned} \quad (7)$$

which is the motion regularizer typically used in scene flow. This regularizer requires smoothness of the motion, but not necessarily rigidity. Therefore, we propose to reduce the weight of the other term in Eq. 6 in order to be able to capture non-rigid motions. Thus we obtain our damped Killing regularizer:

$$E_{Killing}(\Psi) = \sum_{x,y,z} (vec(J_\Psi)^\top vec(J_\Psi) + \gamma vec(J_\Psi^\top)^\top vec(J_\Psi)). \quad (8)$$

The factor  $\gamma$  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 of Eq. 4. As we aim to alleviate the effect of the rigidity constraint, we use values of  $\gamma < 1$  in our optimization.

## 2. Solution

Here we give the detailed derivations of the Euler-Lagrange equations.

### 2.1. Data Term

$$\begin{aligned} \frac{\partial E_{data}}{\partial u} &= \frac{1}{2} \left[ \frac{\partial (\phi_n(x+u, y+v, z+w) - \phi_{global}(x, y, z))^2}{\partial u} - \text{div} \frac{\partial (\phi_n(x+u, y+v, z+w) - \phi_{global}(x, y, z))^2}{\partial \nabla u} \right] = \\ &= \frac{1}{2} \frac{\partial (\phi_n(x+u, y+v, z+w) - \phi_{global}(x, y, z))^2}{\partial u} = \end{aligned}$$

$$\begin{aligned}
&= \frac{1}{2} 2 (\phi_n(x+u, y+v, z+w) - \phi_{global}(x, y, z)) \frac{\partial(\phi_n(x+u, y+v, z+w) - \phi_{global}(x, y, z))}{\partial u} = \\
&= (\phi_n(x+u, y+v, z+w) - \phi_{global}(x, y, z)) \frac{\partial\phi_n(x+u, y+v, z+w)}{\partial u} = \\
&= (\phi_n(x+u, y+v, z+w) - \phi_{global}(x, y, z)) \nabla_x \phi_n(x+u, y+v, z+w)
\end{aligned} \tag{9}$$

In the above  $\nabla_x \phi$  is the  $x$ -component of the spatial gradient of the SDF  $\phi$ .

$$\begin{aligned}
\begin{pmatrix} \partial E_{data}/\partial u \\ \partial E_{data}/\partial v \\ \partial E_{data}/\partial w \end{pmatrix} &= (\phi_n(x+u, y+v, z+w) - \phi_{global}(x, y, z)) \begin{pmatrix} \nabla_x \phi_n(x+u, y+v, z+w) \\ \nabla_y \phi_n(x+u, y+v, z+w) \\ \nabla_z \phi_n(x+u, y+v, z+w) \end{pmatrix} = \\
&= (\phi_n(x+u, y+v, z+w) - \phi_{global}(x, y, z)) \nabla \phi_n(x+u, y+v, z+w) = \\
&= (\phi_n(\Psi) - \phi_{global}) \nabla \phi_n(\Psi)
\end{aligned} \tag{10}$$

Above we used  $\phi_n(\Psi)$  to refer to the evolved SDF after the application of the deformation field vector  $(u, v, w)$ , *i.e.* equivalently to  $\phi_n(x+u, y+v, z+w)$ . We will use this shorthand notation from here onwards.

## 2.2. Level Set Property

$$\begin{aligned}
\frac{\partial E_{level_{set}}}{\partial u} &= \frac{1}{2} \left[ \frac{\partial(|\nabla \phi_n(x+u, y+v, z+w)| - 1)^2}{\partial u} - \text{div} \frac{\partial(|\nabla \phi_n(x+u, y+v, z+w)| - 1)^2}{\partial \nabla u} \right] = \\
&= \frac{1}{2} \frac{\partial(|\nabla \phi_n(x+u, y+v, z+w)| - 1)^2}{\partial u} = \\
&= \frac{1}{2} 2 (|\nabla \phi_n(x+u, y+v, z+w)| - 1) \frac{\partial(|\nabla \phi_n(x+u, y+v, z+w)| - 1)}{\partial u} = \\
&= (|\nabla \phi_n(\Psi)| - 1) \frac{\partial \left( \left( \frac{\partial \phi_n(\Psi)}{\partial x} \right)^2 + \left( \frac{\partial \phi_n(\Psi)}{\partial y} \right)^2 + \left( \frac{\partial \phi_n(\Psi)}{\partial z} \right)^2 \right)^{1/2}}{\partial u} = \\
&= (|\nabla \phi_n(\Psi)| - 1) \frac{1}{2|\nabla \phi_n(\Psi)|_\epsilon} \frac{\partial \left( \left( \frac{\partial \phi_n(\Psi)}{\partial x} \right)^2 + \left( \frac{\partial \phi_n(\Psi)}{\partial y} \right)^2 + \left( \frac{\partial \phi_n(\Psi)}{\partial z} \right)^2 \right)}{\partial u} = \\
&= \frac{|\nabla \phi_n(\Psi)| - 1}{2|\nabla \phi_n(\Psi)|_\epsilon} \left( 2 \frac{\partial \phi_n(\Psi)}{\partial x} \frac{\partial}{\partial u} \left( \frac{\partial \phi_n(\Psi)}{\partial x} \right) + 2 \frac{\partial \phi_n(\Psi)}{\partial y} \frac{\partial}{\partial u} \left( \frac{\partial \phi_n(\Psi)}{\partial y} \right) + 2 \frac{\partial \phi_n(\Psi)}{\partial z} \frac{\partial}{\partial u} \left( \frac{\partial \phi_n(\Psi)}{\partial z} \right) \right) = \\
&= \frac{|\nabla \phi_n(\Psi)| - 1}{|\nabla \phi_n(\Psi)|_\epsilon} \left( \nabla_x \phi_n(\Psi) \nabla_{xx} \phi_n(\Psi) + \nabla_y \phi_n(\Psi) \nabla_{yy} \phi_n(\Psi) + \nabla_z \phi_n(x+u, y+v, z+w) \nabla_{zz} \phi_n(\Psi) \right) = \\
&= \frac{|\nabla \phi_n(\Psi)| - 1}{|\nabla \phi_n(\Psi)|_\epsilon} (\nabla_{xx} \phi_n(\Psi) \quad \nabla_{yy} \phi_n(\Psi) \quad \nabla_{zz} \phi_n(\Psi)) \nabla \phi_n(\Psi)
\end{aligned} \tag{11}$$

Here  $|\cdot|_\epsilon$  denotes the norm plus a small constant  $\epsilon$  which avoids division by zero.

$$\begin{aligned}
\begin{pmatrix} \partial E_{level_{set}}/\partial u \\ \partial E_{level_{set}}/\partial v \\ \partial E_{level_{set}}/\partial w \end{pmatrix} &= \frac{|\nabla \phi_n(\Psi)| - 1}{|\nabla \phi_n(\Psi)|_\epsilon} \begin{pmatrix} \nabla_{xx} \phi_n(\Psi) & \nabla_{xy} \phi_n(\Psi) & \nabla_{xz} \phi_n(\Psi) \\ \nabla_{yx} \phi_n(\Psi) & \nabla_{yy} \phi_n(\Psi) & \nabla_{yz} \phi_n(\Psi) \\ \nabla_{zx} \phi_n(\Psi) & \nabla_{zy} \phi_n(\Psi) & \nabla_{zz} \phi_n(\Psi) \end{pmatrix} \nabla \phi_n(\Psi) = \\
&= \frac{|\nabla \phi_n(\Psi)| - 1}{|\nabla \phi_n(\Psi)|_\epsilon} H_{\phi_n(\Psi)} \nabla \phi_n(\Psi)
\end{aligned} \tag{12}$$

Above we have denoted the Hessian matrix of  $\phi_n(\Psi)$  as  $H_{\phi_n(\Psi)}$ .

### 2.3. Motion Regularization

Expanding the terms in Eq. 8, we obtain:

$$E_{Killing}(\Psi) = \sum_{x,y,z} \left( (1+\gamma)u_x^2 + u_y^2 + u_z^2 + v_x^2 + (1+\gamma)v_y^2 + v_z^2 + w_x^2 + w_y^2 + (1+\gamma)w_z^2 + 2\gamma u_y v_x + 2\gamma u_z w_x + 2\gamma w_y v_z \right). \quad (13)$$

Then,

$$\begin{aligned} \frac{\partial E_{Killing}}{\partial u} &= 2(1+\gamma)u_x u_{xx} + 2(u_y + \gamma v_x)u_{xy} + 2(u_z + \gamma w_x)u_{xz} + \\ &\quad 2(v_x + \gamma u_y)v_{xx} + 2(1+\gamma)v_y v_{xy} + 2(v_z + \gamma w_y)v_{xz} + \\ &\quad 2(w_x + \gamma u_z)w_{xx} + 2(w_y + \gamma v_z)w_{xy} + 2(1+\gamma)w_z w_{xz}, \end{aligned} \quad (14)$$

$$\begin{aligned} \frac{\partial E_{Killing}}{\partial v} &= 2(1+\gamma)u_x u_{xy} + 2(u_y + \gamma v_x)u_{yy} + 2(u_z + \gamma w_x)u_{yz} + \\ &\quad 2(v_x + \gamma u_y)v_{xy} + 2(1+\gamma)v_y v_{yy} + 2(v_z + \gamma w_y)v_{yz} + \\ &\quad 2(w_x + \gamma u_z)w_{xy} + 2(w_y + \gamma v_z)w_{yy} + 2(1+\gamma)w_z w_{yz}, \end{aligned} \quad (15)$$

$$\begin{aligned} \frac{\partial E_{Killing}}{\partial w} &= 2(1+\gamma)u_x u_{xz} + 2(u_y + \gamma v_x)u_{yz} + 2(u_z + \gamma w_x)u_{zz} + \\ &\quad 2(v_x + \gamma u_y)v_{xz} + 2(1+\gamma)v_y v_{yz} + 2(v_z + \gamma w_y)v_{zz} + \\ &\quad 2(w_x + \gamma u_z)w_{xz} + 2(w_y + \gamma v_z)w_{yz} + 2(1+\gamma)w_z w_{zz}. \end{aligned} \quad (16)$$

Finally,

$$\begin{aligned} E'_{Killing}(\Psi) &= 2 \begin{pmatrix} u_{xx} & u_{xy} & u_{xz} & v_{xx} & v_{xy} & v_{xz} & w_{xx} & w_{xy} & w_{xz} \\ u_{xy} & u_{yy} & u_{yz} & v_{xy} & v_{yy} & v_{yz} & w_{xy} & w_{yy} & w_{yz} \\ u_{xz} & u_{yz} & u_{zz} & v_{xz} & v_{yz} & v_{zz} & w_{xz} & w_{yz} & w_{zz} \end{pmatrix} \begin{pmatrix} (1+\gamma)u_x \\ u_y + \gamma v_x \\ u_z + \gamma w_x \\ v_x + \gamma u_y \\ (1+\gamma)v_y \\ v_z + \gamma w_y \\ w_x + \gamma u_z \\ w_y + \gamma v_z \\ (1+\gamma)w_z \end{pmatrix} = \\ &= 2 \begin{pmatrix} H_u & H_v & H_w \end{pmatrix} \begin{pmatrix} u_x & u_x \\ u_y & v_x \\ u_z & w_x \\ v_x & u_y \\ v_y & v_y \\ v_z & w_y \\ w_x & u_z \\ w_y & v_z \\ w_z & w_z \end{pmatrix} \begin{pmatrix} 1 \\ \gamma \end{pmatrix} = 2 \begin{pmatrix} H_u & H_v & H_w \end{pmatrix} \left( \text{vec}(J_{\Psi}^T) \quad \text{vec}(J_{\Psi}) \right) \begin{pmatrix} 1 \\ \gamma \end{pmatrix}, \end{aligned} \quad (17)$$

where  $H_u$  is the Hessian of the x-component of the motion field  $H_u = \begin{pmatrix} u_{xx} & u_{xy} & u_{xz} \\ u_{xy} & u_{yy} & u_{yz} \\ u_{xz} & u_{yz} & u_{zz} \end{pmatrix}$  and  $H_v$  and  $H_w$  are the respective Hessians of the y- and z-component.

### 3. AKVF Visualization

The Killing constraint has advantages over classical motion smoothness because it enforces a divergence-free flow field, *i.e.* a vector field with no sources or sinks. According to the Helmholtz theorem, any vector field can be decomposed into a combination of curl-free and a divergence-free term. Therefore, our regularizer enforces the curl-free part to vanish, as shown below in a 2D example (visualized via [2]). The stronger the influence of the Killing regularizer, the closer to zero the curl-free part will become.

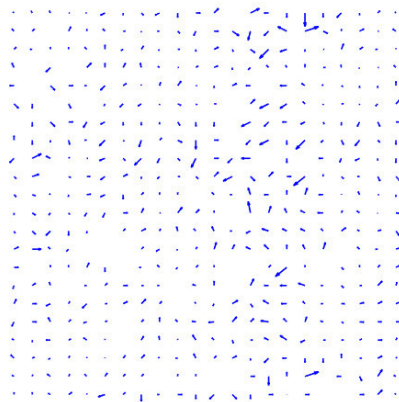


Figure 1. Curl-free component of a 2D approximately Killing vector field.

### 4. Additional Results

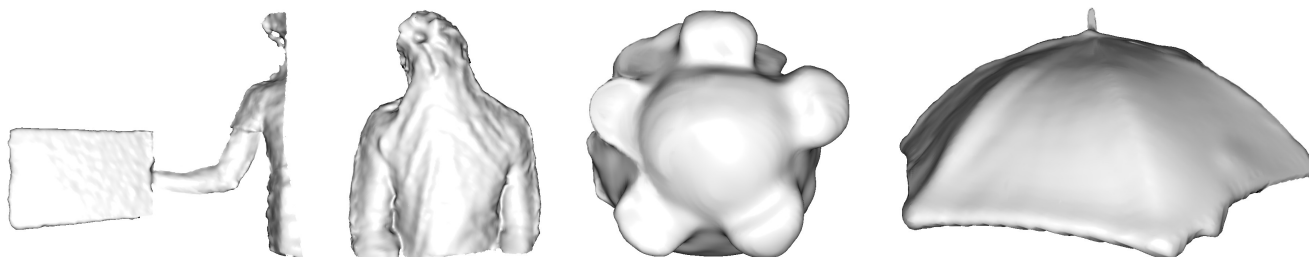


Figure 2. More canonical pose reconstructions of sequences from the VolumeDeform paper [1].

### References

- [1] M. Innmann, M. Zollhöfer, M. Nießner, C. Theobalt, and M. Stamminger. VolumeDeform: Real-time Volumetric Non-rigid Reconstruction. In *European Conference on Computer Vision (ECCV)*, 2016. 5
- [2] G. Peyré. The Numerical Tours of Signal Processing - Advanced Computational Signal and Image Processing. *IEEE Computing in Science and Engineering*, 13(4):94–97, 2011. 5