DROID-SLAM: Supplementary Material

A Additional Results

<table>
<thead>
<tr>
<th></th>
<th>MH01</th>
<th>MH02</th>
<th>MH03</th>
<th>MH04</th>
<th>MH05</th>
<th>V101</th>
<th>V102</th>
<th>V103</th>
<th>V201</th>
<th>V202</th>
<th>V203</th>
<th>Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>D3VO + DSO</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ORB-SLAM2</td>
<td>0.035</td>
<td>0.018</td>
<td>0.028</td>
<td>0.119</td>
<td>0.060</td>
<td>0.035</td>
<td>0.020</td>
<td>0.048</td>
<td>0.037</td>
<td>0.035</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>VINS-Fusion</td>
<td>0.540</td>
<td>0.460</td>
<td>0.330</td>
<td>0.780</td>
<td>0.500</td>
<td>0.550</td>
<td>0.230</td>
<td>-</td>
<td>0.230</td>
<td>0.200</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>SVO</td>
<td>0.040</td>
<td>0.070</td>
<td>0.270</td>
<td>0.170</td>
<td>0.120</td>
<td>0.040</td>
<td>0.040</td>
<td>0.070</td>
<td>0.050</td>
<td>0.090</td>
<td>0.790</td>
<td>0.159</td>
</tr>
<tr>
<td>ORB-SLAM3</td>
<td>0.029</td>
<td>0.019</td>
<td>0.024</td>
<td>0.085</td>
<td>0.052</td>
<td>0.035</td>
<td>0.025</td>
<td>0.230</td>
<td>0.041</td>
<td>0.028</td>
<td>0.521</td>
<td>0.084</td>
</tr>
<tr>
<td>Ours</td>
<td>0.015</td>
<td>0.013</td>
<td>0.035</td>
<td>0.048</td>
<td>0.040</td>
<td>0.037</td>
<td>0.011</td>
<td>0.020</td>
<td>0.018</td>
<td>0.015</td>
<td>0.017</td>
<td>0.024</td>
</tr>
</tbody>
</table>

Table 1: Stereo SLAM on the EuRoC datasets, ATE[m].

We provide stereo results on the EuRoC dataset[1] in Tab.1 using our network trained on synthetic, monocular video. In the stereo setting, it is possible to recover the trajectory of the camera up to scale. Compared to ORB-SLAM3[2] we reduce the average ATE by 71%.

B Ablations

Figure 1: Visualizations of keyframe image, depth, optical flow and confidence estimates.

Figure 2: (Left) we show the performance of the system with different inputs (monocular vs. stereo) and whether global optimization is performed in addition to local BA (local vs. full). (Right) Tracking accuracy as a function of the number of keyframes. We use 5 keyframes (bold) in our experiments.

Ablations We ablate various design choices regarding our SLAM system and network architecture. Ablations are performed on our validation split of the TartanAir dataset. In Fig.1 we show visualizations on the validation set of keyframe depth estimates alongside optical flow and associated confidence weights.

In Fig.2 (left) we show how the system benefits from both stereo video and global optimization. Although our network is only trained on monocular video, it can readily leverage stereo frames if available. In Fig.2 (right) we show how the number of keyframe affects odometry performance. In Fig.3 we ablate components of the network architecture. Fig.3 (left) shows the impact of using global context in the GRU through spatial pooling while Fig.3 (right) demonstrates the importance of
Figure 3: (Left) Impact of global context in the update operator. (Right) Impact of using the bundle adjustment layer during training vs training directly on optical flow, then applying BA at test time.

training with DBA as opposed to training on flow and applying BA at inference. We find that the SLAM system is unstable and prone to failure if the DBA is not used during training.

C Camera Model and Jacobians

We represent 3D points using homogeneous coordinates $X = (X, Y, Z, W)^T$. An image point $p$ with inverse depth $d$ is re-projected from frame $i$ into frame $j$ according to the warping function

$$ p' = \Pi_c(G_{ij} \cdot \Pi^{-1}_i(p, d)) \quad G_{ij} = G_j \circ G_i^{-1} $$

(1)

where $\Pi_c$ is the pinhole projection function, and $\Pi^{-1}_c$ is the inverse projection

$$ \Pi_c(X) = \begin{pmatrix} f_x X + c_y \\ f_y Z + c_y \end{pmatrix} \quad \Pi^{-1}_c(p, d) = \begin{pmatrix} \frac{p_y - c_y}{f_y} \\ \frac{p_x - c_x}{f_x} \\ 1 \end{pmatrix}. $$

(2)

given camera intrinsic parameters $c = (f_x, f_y, c_x, c_y)$.

For optimization, we need the Jacobians with respect to $G_i$, $G_j$, and $d$. We use the local parameterization $\xi_i G_i$ and $\xi_j G_j$ and treat $d$ as a vector in $\mathbb{R}^1$. The Jacobians of the projection and inverse projection functions are given as

$$ \frac{\partial \Pi_c(X)}{\partial X} = \begin{pmatrix} f_x \frac{1}{Z} & 0 & -f_x \frac{X}{Z^2} & 0 \\ 0 & f_y \frac{1}{Z} & -f_y \frac{Y}{Z^2} & 0 \end{pmatrix} \quad \frac{\partial \Pi^{-1}_c(p, d)}{\partial d} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}. $$

(3)

Using the local parameterization, we compute the Jacobian of the 3D point transformation

$$ X' = \text{Exp}(\xi_j) \cdot G_j \cdot (\text{Exp}(\xi_i) \cdot G_i)^{-1} \cdot X = \text{Exp}(\xi_j) \cdot G_j \cdot G_i^{-1} \cdot \text{Exp}(-\xi_i) \cdot X $$

(4)

using the adjoint operator to move the $\xi_i$ term to the front of the expression

$$ X' = \text{Exp}(\xi_j) \cdot \text{Exp}(-\text{Adj}_{G_j G_i^{-1}} \xi_i) \cdot G_j \cdot G_i^{-1} \cdot X $$

(5)

allowing us to compute the Jacobians using the generators

$$ \frac{\partial X'}{\partial \xi_j} = \begin{pmatrix} W' & 0 & 0 & 0 & Z' & -Y' \\ 0 & W' & 0 & -Z' & 0 & X' \\ 0 & 0 & W' & Y' & -X' & 0 \end{pmatrix} \quad \frac{\partial X'}{\partial \xi_i} = -\begin{pmatrix} W' & 0 & 0 & 0 & Z' & -Y' \\ 0 & W' & 0 & -Z' & 0 & X' \\ 0 & 0 & W' & Y' & -X' & 0 \end{pmatrix} \cdot \text{Adj}_{G_j G_i^{-1}} $$

(6)

(7)
Using the chain rule, we can compute the full Jacobians with respect to the variables

\[
\frac{\partial p'}{\partial \xi_j} = \frac{\partial \Pi_c(X')}{\partial X'} \frac{\partial X'}{\partial \xi_j}, \quad \frac{\partial p'}{\partial \xi_i} = \frac{\partial \Pi_c(X')}{\partial X'} \frac{\partial X'}{\partial \xi_i}
\]  

(8)

\[
\frac{\partial p'}{\partial d} = \frac{\partial \Pi_c(X')}{\partial X'} \frac{\partial X'}{\partial d} = \frac{\partial \Pi_c(X')}{\partial X'} = \frac{\partial \Pi_c(X')}{\partial X'} \begin{pmatrix} t_x \\ t_y \\ t_z \\ 1 \end{pmatrix}
\]  

(9)

where \((t_x, t_y, t_z)\) is the translation vector of \(G_j \circ G_i^{-1}\).

D Network Architecture

Figure 4: Architecture of the feature and context encoders. Both extract features at 1/8 the input image resolution using a set of 6 basic residual blocks. Instance normalization is used in the feature encoder; no normalization is used in the context encoder. The feature encoder outputs features with dimension \(D=128\) which the context encoder outputs features with dimension \(D=256\).

Figure 5: Architecture of the update operator. During each iteration, context, correlation, and flow features get injected into the GRU. The revision \(r\) and confidence weights \(w\) are predicted from the updated hidden state.

References


