Supplementary Material

DirectShape: Photometric Alignment of Shape Priors for Visual Vehicle Pose and Shape Estimation

Rui Wang\textsuperscript{1}, Nan Yang\textsuperscript{1}, Jörg Stückler\textsuperscript{2}, Daniel Cremers\textsuperscript{1}

Abstract—In this supplementary document, we first show the shape variations of the adopted PCA model in Sec. I. In Sec. II, the full derivations of the analytical Jacobians of all the residuals defined in the main paper are presented. Lastly in Sec. III, we show more results on the KITTI dataset, which qualitatively demonstrate the ability of our method to recover the 3D poses and shapes of cars in challenging real-world environments. Apart from this document, a video showing how our method works on some selected stereo frames can be found on the project page https://vision.in.tum.de/research/vslam/direct-shape.

I. SHAPE VARIATIONS OF PCA MODEL

To demonstrate that our PCA model can deform and fit properly to a variety of car shapes, we first fit it to 12 selected vehicles from the CAD samples which we used to extract the PCA model. The shapes together with the color coded signed distance function (SDF) are shown in Fig. 1. An animation showing the different car shapes by modifying the shape coefficients \(z\) can be found on the project page. We further show some real-world examples in Fig. 2, where the PCA model is optimized using our approach to fit the corresponding cars in the second row. We claim that although the adopted PCA shape embedding is a simple linear model, it works nicely for object categories like cars.

II. FULL DERIVATIONS OF JACOBIANS

A. Jacobian of Silhouette Alignment Residual

As the relative transformation between the left and right cameras are considered to be fixed in this work, the Jacobians of \(r_{\text{silh}}\) and \(r_{\text{silh}}\) are the same and we will omit the superscript in the following. As shown in Eq. 2 in the main paper, the silhouette alignment residual of pixel \(p\) is defined as

\[
r_{\text{silh}}(p) = -\log \left( \frac{\pi(\Phi, p) p_{fg}(p) + (1 - \pi(\Phi, p)) p_{bg}(p)}{\pi(\Phi, p)} \right),
\]

\[
:= A(\pi)
\]

(1)

thus using chain rule its Jacobian with respect to the pose and shape parameters \([\xi; z]\) can be factorized to

\[
J_{\text{silh}} = \frac{\partial r_{\text{silh}}(p)}{\partial [\xi; z]} = \frac{\partial \log(A(\pi))}{\partial A(\pi)} \frac{\partial A(\pi)}{\partial \pi} \frac{\partial \pi(\Phi, p)}{\partial \Phi} \frac{\partial \Phi}{\partial [\xi; z]},
\]

(3)

where

\[
\frac{\partial \log(A(\pi))}{\partial A(\pi)} = \frac{1}{A(\pi)}
\]

(4)

\[
\frac{\partial A(\pi)}{\partial \pi} = p_{fg}(p) - p_{bg}(p).
\]

(5)

Recall that the shape embedding projection function \(\pi(\Phi, p)\) is defined as

\[
\pi(\Phi, p) = 1 - \prod_{X_v} \frac{1}{\exp(\Phi(X_v)\kappa) + 1},
\]

(6)

to make it easier to derive its Jacobian, we convert the multiplications in \(\pi(\Phi, p)\) to summations by reformulating it to

\[
\pi(\Phi, p) = 1 - \exp \left( \sum_{X_v} \log \frac{1}{\exp(\Phi(X_v)\kappa + 1)} \right) = \exp(-B(\Phi)).
\]

(7)

Therefore,

\[
\frac{\partial \pi(\Phi, p)}{\partial \Phi} = -\exp(B(\Phi)) \sum_{X_v} \frac{1}{C(\Phi)} \frac{\partial C(\Phi)}{\partial \Phi},
\]

(8)

where

\[
\frac{\partial C(\Phi)}{\partial \Phi} = \frac{\partial \left( \exp(-X_v \kappa + 1) \right)}{\partial \Phi} = \exp(-X_v \kappa) \left( \exp(-X_v \kappa + 1) \right)^{-1} \frac{\partial X_v \kappa}{\partial \Phi} = \exp(-X_v \kappa) \left( \exp(-X_v \kappa + 1) \right)^{-1} \cdot (-1) \frac{X_v}{\exp(-X_v \kappa + 1)^2}.
\]

(9)

(10)

(11)

The remaining part to derive is \(\frac{\partial \Phi(X_v)}{\partial \kappa}\). As \(\Phi(X_v) = V(X_v)z + \Phi_{\text{mean}} = \sum_{k=1}^{K} v_k(X_v)z_k + \Phi_{\text{mean}}\), we have

\[
\frac{\partial \Phi(X_v)}{\partial z} = [v_1(X_v), v_2(X_v), ... , v_K(X_v)].
\]

(12)
To derive $\frac{\partial \Phi(X_o)}{\partial \xi^o}$, we first compute the coordinate for $X_o$ in the camera coordinate system as $X_c$, so we have $X_o = \exp(\hat{\xi}^o) X_c$, where $\exp(\cdot)$ is the exponential map that maps the twist coordinate to $\text{SE}(3)$. The remaining part of the Jacobian is then computed as

$$\frac{\partial \Phi(X_o)}{\partial \xi^o} = \nabla \Phi \Big|_{X_o} \frac{\partial X_o}{\partial \xi^o},$$

where $\nabla \Phi$ is the spatial gradient of $\Phi$, $\delta \xi$ is a small increment in $\text{se}(3)$ and is applied with the exponential map to the left hand side of the pose estimate. The closed form solution for $\frac{\partial \exp(\delta \xi)}{\partial (\delta \xi)}$ near $\delta \xi = 0$ can be obtained using the infinitesimal generators of $\text{SE}(3)$ (please refer Eq. 44 and 45).

Depending on the derivations of the specific derivatives above, the full Jacobian of the silhouette alignment residual can be computed by combining Eq. 4, 5, 8, 11 and Eq. 12, 13, 14.

### B. Jacobian of the Photometric Consistency Residual

As defined in the main paper, the photometric consistency residual of pixel $p$ is

$$r_{\text{photo}}(p) = I_r(\Pi_c\Pi^{-1}_r(p, d^p) + t^r) - I_l(p),$$

where the pose and shape parameters $[\xi^o_c; z]$ only appear in $d^p$. Using chain rule the Jacobian with respect to the pose and shape parameters can be factorized to

$$J_{\text{photo}} = \frac{\partial r_{\text{photo}}(p)}{\partial [\xi^o_c; z]} \bigg|_{\xi^o_c} = \nabla I_r(warp(p, d^p)) \frac{\partial warp(p, d^p)}{\partial p} \frac{\partial d^p}{\partial [\xi^o_c; z]},$$

where $warp(p, d^p) = \Pi_c(R_t^r\Pi^{-1}_r(p, d^p) + t^r)$ is the pixel warping function from the left image to the right image, $\nabla I_r(warp(p, d^p))$ is the image gradient of the right image at the warped pixel location $warp(p, d^p)$. In the following we derive $\frac{\partial warp(p, d^p)}{\partial p}$ and $\frac{\partial d^p}{\partial [\xi^o_c; z]}$ successively.

Denoting the 3D coordinates of $p$ in the left and the right
camera coordinate systems as \(X_l\) and \(X_r\), we have

\[
warp(p, dp) = \Pi_c \left( \mathbf{R}_l^T \mathbf{P}_c^{-1} (p, dp) + t_o^l \right)
\]

\[
X_l = d_p K^{-1} [p(u), p(v), 1]^T,
\]

\[
X_r = \mathbf{R}_r^T X_l + t_o^r,
\]

\[
= \frac{d_p \mathbf{R}_r^T K^{-1} [p(u), p(v), 1]^T + t_o^r}{v = [v(x), v(y), v(z)]^T}
\]

\[
d_p v + t_o^r,
\]

\[
\Pi_c (X_r) = \begin{bmatrix} f_u & 0 & c_u \cr f_v & 0 & c_v \cr 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_r(x) \\
X_r(y) \\
1 \end{bmatrix}
\]

\[
= \begin{bmatrix} f_u (X_r(x) + c_u) \\
f_v (X_r(y) + c_v) \end{bmatrix},
\]

where \(K = [f_u, 0, c_u; 0, f_v, c_v; 0, 0, 1]\) is the camera intrinsic matrix. \(\partial \text{warp}(p, dp) / \partial dp\) therefore can be computed as

\[
\frac{\partial \text{warp}(p, dp)}{\partial dp} = \begin{bmatrix} \frac{\partial X_r(x)}{\partial dp} \\
\frac{\partial X_r(y)}{\partial dp} \end{bmatrix}
\]

\[
= \begin{bmatrix} f_u \frac{\partial X_r(x)}{\partial dp} \\
\frac{\partial X_r(y)}{\partial dp} \end{bmatrix}
\]

\[
= \begin{bmatrix} X_r(x) \\
X_r(y) \end{bmatrix}
\]

\[
= \begin{bmatrix} f_u (X_r(x) + c_u) \\
f_v (X_r(y) + c_v) \end{bmatrix},
\]

To compute \(\partial dp / \partial \xi_c^z\), we first compute the 3D coordinate of the intersecting point of the ray and the zero-level surface based on \(d_p\) obtained by ray-casting, then transform it from the camera coordinate system to the object coordinate system and denote it as \(X^d_0\). The Jacobian with respect to the shape encoding vector is then computed as

\[
\frac{\partial dp}{\partial \Phi} = \frac{\partial dp}{\partial z} |_{\Phi(X^d_0)} = \frac{\partial \Phi}{\partial z} |_{\Phi(X^d_0)}
\]

where \(\partial \Phi / \partial z\) can be computed similarly as in Eq. 12, the derivation of \(\partial dp / \partial \Phi\) is illustrated in Fig. 3. At the intersecting point \(X^d_0\), the change of the depth along the ray \(\partial d\) is approximately proportional to the change of the SDF value \(\partial \Phi\) by a factor of \(1/\cos(\theta)\) where \(\theta\) is the angle between the ray and the surface normal. Taking the sign into account we have

\[
\frac{\partial dp}{\partial \Phi} = \frac{\partial dp}{\partial \Phi} |_{\Phi(X^d_0)} = -\frac{1}{\cos(\theta)}.
\]

The Jacobian with respect to \(\xi_c^z\) can be factorized to

\[
\frac{\partial dp}{\partial \xi_c^z} = \frac{\partial dp}{\partial \Phi} |_{\Phi(X^d_0)} \nabla \Phi |_{X^d_0} \frac{\partial X^d_0}{\partial \xi_c^z},
\]

which can be computed according to Eq. 29 and 14.
are the infinitesimal generators of SE(3). Assuming the first three elements in the twist coordinate correspond to the translation part and the last three correspond to the rotation part, the infinitesimal generators are defined as:

\[
G_0 = \begin{bmatrix}
0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0
\end{bmatrix},
\]

(46)

\[
G_1 = \begin{bmatrix}
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0
\end{bmatrix},
\]

(47)

\[
G_2 = \begin{bmatrix}
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0
\end{bmatrix},
\]

(48)

\[
G_3 = \begin{bmatrix}
0 & 0 & -1 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0
\end{bmatrix},
\]

(49)

\[
G_4 = \begin{bmatrix}
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 \\
-1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0
\end{bmatrix},
\]

(50)

\[
G_5 = \begin{bmatrix}
0 & -1 & 0 & 0 \\
1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0
\end{bmatrix},
\]

(51)

Lastly, the Jacobian with respect to \(z\) is

\[
\frac{\partial r_{trans}}{\partial z} = 0.
\]

(52)

3) Jacobian of Rotation Prior Residuals.: The rotation prior residual can be reformulated to

\[
r_{rot} = 1 - (R_0^c[0, -1, 0]^	op n_g
\]

(53)

\[
= 1 - [0, -1, 0] R_0^c \top n_g
\]

(54)

\[
= 1 + [0, 1, 0] R_0^c n_g
\]

(55)

\[
= 1 + r_2 n_g,
\]

(56)

where \(r_1\) is the second row of \(R_0^c\). Therefore, the Jacobian with respect to \(\xi_c^o\) is

\[
\frac{\partial r_{rot}}{\partial \xi_c^o} = \frac{\partial r_1}{\partial \xi_c^o} n_g
\]

(57)

\[
= \frac{\partial T_c^o (1:0:2)}{\partial \xi_c^o} n_g
\]

(58)

\[
= \left(\frac{\partial \text{exp}(\delta \xi)}{\partial \delta \xi}\right) T_c^o (1:0:2) n_g
\]

(59)

\[
= \left(\begin{bmatrix} G_0^o & \cdots & G_5^o \end{bmatrix} (1:0:2) n_g
\]

(60)

where (1:0:2) denotes the operation of getting the part corresponding to the second row of the rotation matrix.

Lastly, we have

\[
\frac{\partial r_{rot}}{\partial z} = 0.
\]

(61)
Fig. 4: Qualitative results on 3D pose and shape refinements. Each two-row block shows the results on one stereo frame from the KITTI Stereo 2015 benchmark. Top row: The initial pose and shape (left) and our results (right) projected to the left image. Bottom row: The initial poses estimated by 3DOP and the poses refined by our method shown together with the ground truth 3D point cloud (1st and 2nd); The following three images show the 3D points estimated by ELAS (middle), our method (4th) and from the ground truth (last). Note that the 3D point clouds are not used in our optimization. (better viewed electronically)
Fig. 5: Qualitative results on 3D pose and shape estimation.
Fig. 6: Qualitative results on 3D pose and shape estimation (cont.).