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

Rui Wang[1], Nan Yang[1], Jörg Stückler[2], Daniel Cremers[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 $\mathbf{z}$ 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^l_{silh} and r^r_{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 $\mathbf{p}$ is defined as

$$r_{silh}(\mathbf{p}) = -\log\big(\underbrace{\pi(\mathbf{\Phi},\mathbf{p})p_{fg}(\mathbf{p}) + (1-\pi(\mathbf{\Phi},\mathbf{p}))p_{bg}(\mathbf{p})}_{:=A(\pi)}\big), \tag{1}$$

thus using chain rule its Jacobian with respect to the pose and shape parameters $[\boldsymbol{\xi}^o_c;\mathbf{z}]$ can be factorized to

$$\mathbf{J}_{silh} = \frac{\partial r_{silh}(\mathbf{p})}{\partial[\boldsymbol{\xi}^o_c;\mathbf{z}]} \tag{2}$$

$$= -\frac{\partial log(A(\pi))}{\partial A(\pi)}\frac{\partial A(\pi)}{\partial \pi}\frac{\partial \pi(\mathbf{\Phi},\mathbf{p})}{\partial \mathbf{\Phi}}\frac{\partial \mathbf{\Phi}}{\partial[\boldsymbol{\xi}^o_c;\mathbf{z}]}, \tag{3}$$

where

$$\frac{\partial log(A(\pi))}{\partial A(\pi)} = \frac{1}{A(\pi)}, \tag{4}$$

$$\frac{\partial A(\pi)}{\partial \pi} = p_{fg}(\mathbf{p}) - p_{bg}(\mathbf{p}). \tag{5}$$

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

$$\pi(\mathbf{\Phi},\mathbf{p}) = 1 - \prod_{\mathbf{X}_o}\frac{1}{e^{\mathbf{\Phi}(\mathbf{X}_o)\zeta}+1}, \tag{6}$$

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

$$\pi(\mathbf{\Phi},\mathbf{p}) = 1 - \exp\Bigg(\underbrace{\sum_{\mathbf{X}_o}\log\Bigg(\overbrace{\frac{1}{e^{\mathbf{\Phi}(\mathbf{X}_o)\zeta}+1}}^{:=C(\mathbf{\Phi})}\Bigg)}_{:=B(\mathbf{\Phi})}\Bigg). \tag{7}$$

Therefore,

$$\frac{\partial \pi(\mathbf{\Phi},\mathbf{p})}{\partial \mathbf{\Phi}} = -\exp(B(\mathbf{\Phi}))\sum_{\mathbf{X}_o}\frac{1}{C(\mathbf{\Phi})}\frac{\partial C(\mathbf{\Phi})}{\partial \mathbf{\Phi}}, \tag{8}$$

where

$$\frac{\partial C(\mathbf{\Phi})}{\partial \mathbf{\Phi}} = \frac{\partial(\frac{1}{e^{\mathbf{\Phi}(\mathbf{X}_o)\zeta}+1})}{\partial \mathbf{\Phi}} \tag{9}$$

$$= (-1)\frac{e^{\mathbf{\Phi}(\mathbf{X}_o)\zeta}}{(e^{\mathbf{\Phi}(\mathbf{X}_o)\zeta}+1)^2}\zeta \tag{10}$$

$$= -\frac{\zeta e^{\mathbf{\Phi}(\mathbf{X}_o)\zeta}}{(e^{\mathbf{\Phi}(\mathbf{X}_o)\zeta}+1)^2}. \tag{11}$$

The remaining part to derive is $\partial\mathbf{\Phi}(\mathbf{X}_o)/\partial[\boldsymbol{\xi}^o_c;\mathbf{z}]$. As $\mathbf{\Phi}(\mathbf{X}_o) = \mathbf{V}(\mathbf{X_o})\mathbf{z} + \mathbf{\Phi}_{mean} = \sum_{k=1}^K \mathbf{v}_k(\mathbf{X}_o)z_k + \mathbf{\Phi}_{mean}$, we have

$$\frac{\partial \mathbf{\Phi}(\mathbf{X}_o)}{\partial \mathbf{z}} = [\mathbf{v}_1(\mathbf{X}_o), \mathbf{v}_2(\mathbf{X}_o), ..., \mathbf{v}_K(\mathbf{X}_o)]. \tag{12}$$

[1]R. Wang, N. Yang and D. Cremers are with the Department of Computer Science, Technical University of Munich, Garching bei München, 85748, Germany and Artisense Corporation, 350 Cambridge Avenue 250, Palo Alto, CA 94306, USA. {wangr, yangn, cremers}@in.tum.de

[2]J. Stückler is with Max Planck Institute for Intelligent Systems Tübingen, Tübingen, 72076, Germany. joerg.stueckler@tuebingen.mpg.de

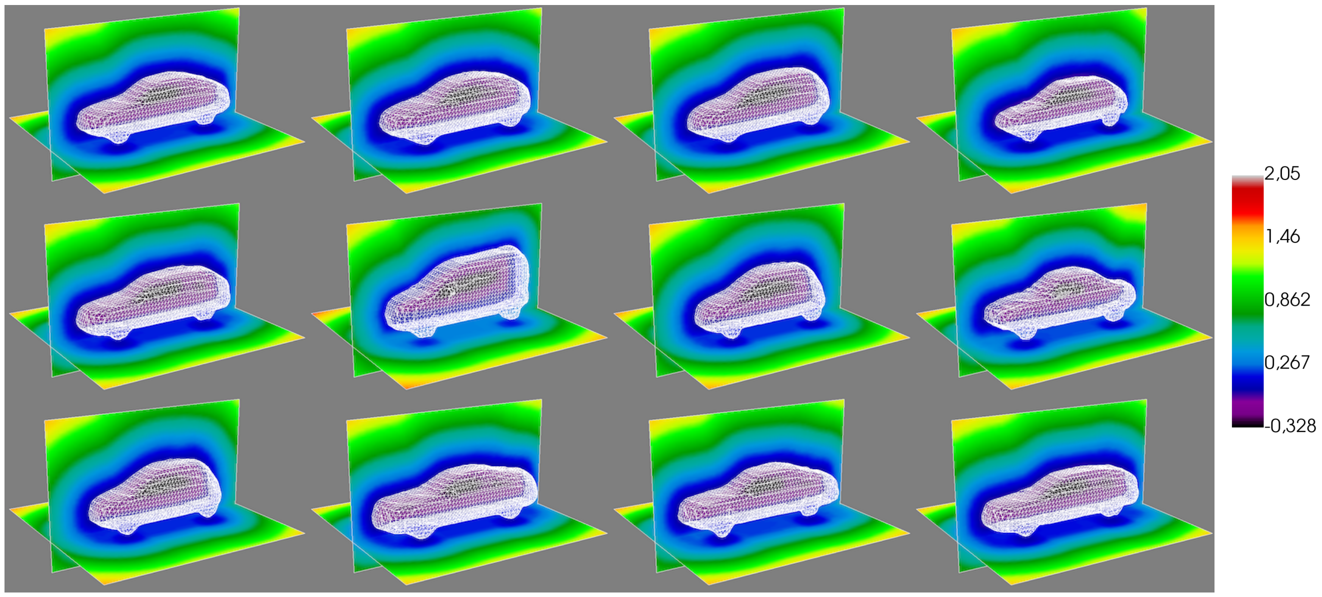

Fig. 1: Shape variations of PCA model with color coded SDF.

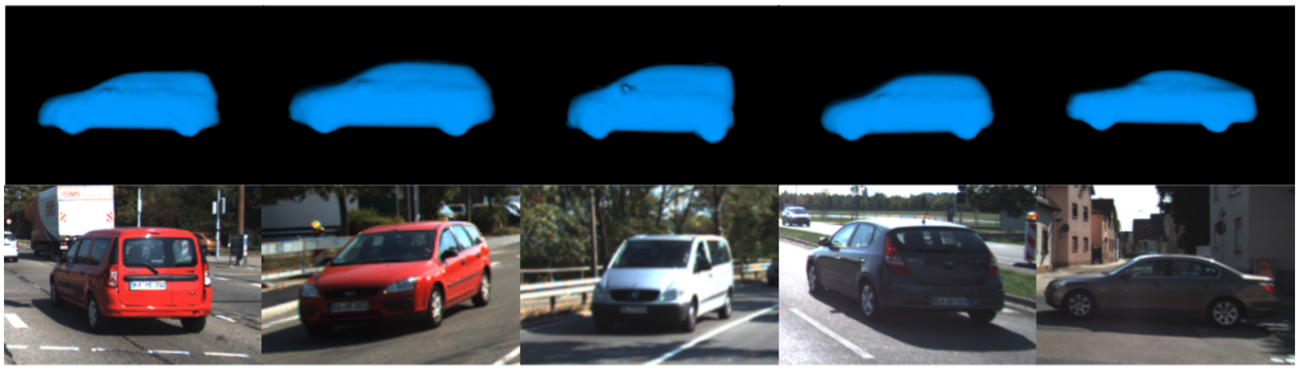

Fig. 2: Qualitative results on the shape estimation.

To derive $\partial\mathbf{\Phi}(\mathbf{X}_o)/\partial\boldsymbol{\xi}_c^o$, we first compute the coordinate for $\mathbf{X}_o$ in the camera coordinate system as $\mathbf{X}_c$, so we have $\mathbf{X}_o = \exp\left(\hat{\boldsymbol{\xi}_c^o}\right)\mathbf{X}_c$, where $\exp(\hat{\cdot})$ is the exponential map that maps the twist coordinate to SE(3). The remaining part of the Jacobian is then computed as

$$\frac{\partial\mathbf{\Phi}(\mathbf{X}_o)}{\partial\boldsymbol{\xi}_c^o} = \nabla\mathbf{\Phi}\Big|_{\mathbf{X}_o}\frac{\partial\mathbf{X}_o}{\partial\boldsymbol{\xi}_c^o}, \tag{13}$$

$$\frac{\partial\mathbf{X}_o}{\partial\boldsymbol{\xi}_c^o} = \frac{\partial\exp\left(\hat{\boldsymbol{\xi}_c^o}\right)}{\partial\boldsymbol{\xi}_c^o}\Bigg|_{\boldsymbol{\xi}_c^o}\mathbf{X}_c = \frac{\partial\exp(\hat{\delta\boldsymbol{\xi}})}{\partial(\delta\boldsymbol{\xi})}\Bigg|_{\mathbf{0}}\exp(\hat{\boldsymbol{\xi}_c^o})\mathbf{X}_c, \tag{14}$$

where $\nabla\mathbf{\Phi}$ is the spatial gradient of $\mathbf{\Phi}$, $\delta\boldsymbol{\xi}$ is a small increment in $\mathfrak{se}(3)$ and is applied with the exponential map to the left hand side of the pose estimate. The closed form solution for $\partial\exp(\hat{\delta\boldsymbol{\xi}})/\partial(\delta\boldsymbol{\xi})$ near $\delta\boldsymbol{\xi} = \mathbf{0}$ can be obtained using the infinitesimal generators of 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 $\mathbf{p}$ is

$$r_{photo}(\mathbf{p}) = \mathbf{I}_r\Big(\underbrace{\Pi_c(\mathbf{R}_l^r\Pi_c^{-1}(\mathbf{p}, d_\mathbf{p}) + \mathbf{t}_l^r)}_{:=warp(\mathbf{p},d_\mathbf{p})}\Big) - \mathbf{I}_l(\mathbf{p}), \tag{15}$$

where the pose and shape parameters $[\boldsymbol{\xi}_c^o; \mathbf{z}]$ only appear in $d_\mathbf{p}$. Using chain rule the Jacobian with respect to the pose and shape parameters can be factorized to

$$\mathbf{J}_{photo} = \frac{\partial r_{photo}(\mathbf{p})}{\partial[\boldsymbol{\xi}_c^o; \mathbf{z}]} \tag{16}$$

$$= \nabla\mathbf{I}_r(warp(\mathbf{p}, d_\mathbf{p}))\frac{\partial warp(\mathbf{p}, d_\mathbf{p})}{\partial d_\mathbf{p}}\frac{\partial d_\mathbf{p}}{\partial[\boldsymbol{\xi}_c^o; \mathbf{z}]}, \tag{17}$$

where $warp(\mathbf{p}, d_\mathbf{p}) = \Pi_c(\mathbf{R}_l^r\Pi_c^{-1}(\mathbf{p}, d_\mathbf{p}) + \mathbf{t}_l^r)$ is the pixel warping function from the left image to the right image, $\nabla\mathbf{I}_r(warp(\mathbf{p}, d_\mathbf{p}))$ is the image gradient of the right image at the warped pixel location $warp(\mathbf{p}, d_\mathbf{p})$. In the following we derive $\partial warp(\mathbf{p}, d_\mathbf{p})/\partial d_\mathbf{p}$ and $\partial d_\mathbf{p}/\partial[\boldsymbol{\xi}_c^o; \mathbf{z}]$ successively.

Denoting the 3D coordinates of $\mathbf{p}$ in the left and the right camera coordinate systems as $\mathbf{X}_l$ and $\mathbf{X}_r$, we have

$$warp(\mathbf{p}, d_\mathbf{p}) = \Pi_c(\underbrace{\mathbf{R}_l^r \overbrace{\Pi_c^{-1}(\mathbf{p}, d_\mathbf{p})}^{\mathbf{X}_l} + \mathbf{t}_l^r}_{\mathbf{X}_r}), \tag{18}$$

$$\mathbf{X}_l = d_\mathbf{p} \mathbf{K}^{-1}[\mathbf{p}(u), \mathbf{p}(v), 1]^\top, \tag{19}$$

$$\begin{aligned} \mathbf{X}_r &= \mathbf{R}_l^r \mathbf{X}_l + \mathbf{t}_l^r, && (20)\\ &= d_\mathbf{p} \underbrace{\mathbf{R}_l^r \mathbf{K}^{-1}[\mathbf{p}(u), \mathbf{p}(v), 1]^\top}_{:=\mathbf{v}=[\mathbf{v}(x),\mathbf{v}(y),\mathbf{v}(z)]^\top} + \mathbf{t}_l^r, && (21)\\ &= d_\mathbf{p} \mathbf{v} + \mathbf{t}_l^r, && (22) \end{aligned}$$

$$\begin{aligned} \Pi_c(\mathbf{X}_r) &= \begin{bmatrix} f_u & 0 & c_u \\ 0 & f_v & c_v \end{bmatrix} \begin{bmatrix} \frac{\mathbf{X}_r(x)}{\mathbf{X}_r(z)} \\ \frac{\mathbf{X}_r(y)}{\mathbf{X}_r(z)} \\ 1 \end{bmatrix} && (23)\\ &= \begin{bmatrix} f_u \frac{\mathbf{X}_r(x)}{\mathbf{X}_r(z)} + c_u \\ f_v \frac{\mathbf{X}_r(y)}{\mathbf{X}_r(z)} + c_v, \end{bmatrix}, && (24) \end{aligned}$$

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

$$\begin{aligned} \frac{\partial warp(\mathbf{p}, d_\mathbf{p})}{\partial d_\mathbf{p}} &= \begin{bmatrix} f_u \frac{\partial \frac{\mathbf{X}_r(x)}{\mathbf{X}_r(z)}}{d_\mathbf{p}} \\ f_v \frac{\partial \frac{\mathbf{X}_r(y)}{\mathbf{X}_r(z)}}{d_\mathbf{p}} \end{bmatrix} && (25)\\ &= \begin{bmatrix} f_u \frac{\frac{\partial \mathbf{X}_r(x)}{\partial d_\mathbf{p}} \mathbf{X}_r(z) - \mathbf{X}_r(x) \frac{\partial \mathbf{X}_r(z)}{\partial d_\mathbf{p}}}{\mathbf{X}_r^2(z)} \\ f_v \frac{\frac{\partial \mathbf{X}_r(y)}{\partial d_\mathbf{p}} \mathbf{X}_r(z) - \mathbf{X}_r(y) \frac{\partial \mathbf{X}_r(z)}{\partial d_\mathbf{p}}}{\mathbf{X}_r^2(z)} \end{bmatrix} && (26)\\ &= \begin{bmatrix} f_u \frac{\mathbf{v}(x)\mathbf{X}_r(z) - \mathbf{X}_r(x)\mathbf{v}(z)}{\mathbf{X}_r^2(z)} \\ f_v \frac{\mathbf{v}(y)\mathbf{X}_r(z) - \mathbf{X}_r(y)\mathbf{v}(z)}{\mathbf{X}_r^2(z)} \end{bmatrix}. && (27) \end{aligned}$$

To compute $\partial d_\mathbf{p}/\partial[\boldsymbol{\xi}_c^o; \mathbf{z}]$, we first compute the 3D coordinate of the intersecting point of the ray and the zero-level surface based on $d_\mathbf{p}$ obtained by ray-casting, then transform it from the camera coordinate system to the object coordinate system and denote it as $\mathbf{X}_o^d$. The Jacobian with respect to the shape encoding vector is then computed as

$$\frac{\partial d_\mathbf{p}}{\partial \mathbf{z}} = \frac{\partial d_\mathbf{p}}{\partial \boldsymbol{\Phi}}\Big|_{\boldsymbol{\Phi}(\mathbf{X}_o^d)} \frac{\partial \boldsymbol{\Phi}}{\partial \mathbf{z}}\Big|_{\mathbf{z}}, \tag{28}$$

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

$$\frac{\partial d_\mathbf{p}}{\partial \boldsymbol{\Phi}}\Big|_{\boldsymbol{\Phi}(\mathbf{X}_o^d)} = -\frac{1}{\cos(\theta)}. \tag{29}$$

The Jacobian with respect to $\boldsymbol{\xi}_c^o$ can be factorized to

$$\frac{\partial d_\mathbf{p}}{\partial \boldsymbol{\xi}_c^o} = \frac{\partial d_\mathbf{p}}{\partial \boldsymbol{\Phi}}\Big|_{\boldsymbol{\Phi}(\mathbf{X}_o^d)} \nabla \boldsymbol{\Phi}\Big|_{\mathbf{X}_o^d} \frac{\partial \mathbf{X}_o^d}{\partial \boldsymbol{\xi}_c^o}\Big|_{\boldsymbol{\xi}_c^o}, \tag{30}$$

which can be computed according to Eq. 29 and 14.

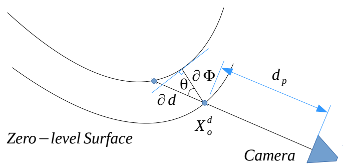

Fig. 3: Deriving the Jacobian of the depth wrt. the SDF value.

C. Jacobian of Prior Residuals

Based on the energy terms defined in the Eq. 7-9 in the main paper, we define the residuals of the priors on the shape and pose parameters as

$$\begin{aligned} r_{shape}^i &= \frac{z_i}{\sigma_i}, \quad i = 1, 2, ..., K && (31)\\ r_{trans} &= \mathbf{t}_o^c(y) - g(\mathbf{t}_o^c(x, z))(y), && (32)\\ r_{rot} &= 1 - (\mathbf{R}_o^c[0, -1, 0]^\top)^\top \mathbf{n}_g. && (33) \end{aligned}$$

1) Jacobian of Shape Prior Residuals.: Based on Eq. 31 we have

$$\begin{aligned} \frac{\partial r_{shape}^i}{\partial \boldsymbol{\xi}_c^o} &= \mathbf{0}, && (34)\\ \frac{\partial r_{shape}^i}{\partial \mathbf{z}} &= [0, ..., 0, \frac{1}{\sigma_i}, 0, ..., 0]. && (35) \end{aligned}$$

2) Jacobian of Translation Prior Residuals.: Denoting the equation for the ground plane as $\mathbf{n}_g(x)x + \mathbf{n}_g(y)y + \mathbf{n}_g(z)z + d = 0$ with $\mathbf{n}_g$ the plane normal vector and d a constant, the height of the ground plane at $\mathbf{t}_o^c(x, z)$ is

$$g(\mathbf{t}_o^c(x, z))(y) = -\frac{\mathbf{n}_g(x)\mathbf{t}_o^c(x) + \mathbf{n}_g(z)\mathbf{t}_o^c(z) + d}{\mathbf{n}_g(y)}, \tag{36}$$

thus

$$r_{trans} = \mathbf{t}_o^c(y) + \frac{\mathbf{n}_g(x)\mathbf{t}_o^c(x) + \mathbf{n}_g(z)\mathbf{t}_o^c(z) + d}{\mathbf{n}_g(y)}. \tag{37}$$

Its Jacobian with respect to $\boldsymbol{\xi}_c^o$ then can be computed as

$$\begin{aligned} \frac{\partial r_{trans}}{\partial \boldsymbol{\xi}_c^o} &= \frac{\partial r_{trans}}{\partial \mathbf{t}_o^c} \frac{\partial \mathbf{t}_o^c}{\partial \boldsymbol{\xi}_c^o} && (38)\\ &= [\frac{\mathbf{n}_g(x)}{\mathbf{n}_g(y)}, 1, \frac{\mathbf{n}_g(z)}{\mathbf{n}_g(y)}] \frac{\partial \mathbf{t}_o^c}{\partial \boldsymbol{\xi}_c^o}, && (39) \end{aligned}$$

where the last term can be computed as

$$\begin{aligned} \frac{\partial \mathbf{t}_o^c}{\partial \boldsymbol{\xi}_c^o} &= \frac{\partial \mathbf{T}_o^c(0:2, 3)}{\partial \boldsymbol{\xi}_c^o} && (40)\\ &= \frac{\partial \mathbf{T}_c^{o-1}(0:2, 3)}{\partial \boldsymbol{\xi}_c^o} && (41)\\ &= \frac{\partial \big((exp(\hat{\delta \boldsymbol{\xi}})\mathbf{T}_c^o)^{-1}(0:2, 3)\big)}{\partial (\delta \boldsymbol{\xi})} && (42)\\ &= \frac{\partial \big((\mathbf{T}_o^c exp(-\hat{\delta \boldsymbol{\xi}}))(0:2, 3)\big)}{\partial (\delta \boldsymbol{\xi})} && (43)\\ &= \big(\mathbf{T}_o^c(-\frac{\partial exp(\hat{\delta \boldsymbol{\xi}})}{\partial (\delta \boldsymbol{\xi})})\big)(0:2, 3) && (44)\\ &= -[(\mathbf{T}_o^c \mathbf{G}_0)(0:2, 3), (\mathbf{T}_o^c \mathbf{G}_1)(0:2, 3), ..., (\mathbf{T}_o^c \mathbf{G}_5)(0:2, 3)], && (45) \end{aligned}$$

where we use $(0:2, 3)$ to denote the operation of getting the translation part from the corresponding matrix; $\mathbf{G}_0, ..., \mathbf{G}_5$

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

$$G_0 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \tag{46}$$

$$G_1 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \tag{47}$$

$$G_2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \tag{48}$$

$$G_3 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \tag{49}$$

$$G_4 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \tag{50}$$

$$G_5 = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \tag{51}$$

Lastly, the Jacobian with respect to $\mathbf{z}$ is

$$\frac{\partial r_{trans}}{\partial \mathbf{z}} = \mathbf{0}. \tag{52}$$

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

$$\begin{aligned} r_{rot} &= 1 - (\mathbf{R}_o^c[0,-1,0]^\top)^\top \mathbf{n}_g && (53) \\ &= 1 - [0,-1,0]\mathbf{R}_o^{c\top}\mathbf{n}_g && (54) \\ &= 1 + [0,1,0]\mathbf{R}_c^o\mathbf{n}_g && (55) \\ &= 1 + \mathbf{r}_2\mathbf{n}_g, && (56) \end{aligned}$$

where $\mathbf{r}_1$ is the second row of $\mathbf{R}_c^o$. Therefore, the Jacobian with respect to $\boldsymbol{\xi}_c^o$ is

$$\begin{aligned} \frac{\partial r_{rot}}{\partial \boldsymbol{\xi}_c^o} &= \frac{\partial \mathbf{r}_1}{\partial \boldsymbol{\xi}_c^o}\mathbf{n}_g && (57) \\ &= \frac{\partial \mathbf{T}_c^o(1,0:2)}{\partial \boldsymbol{\xi}_c^o}\mathbf{n}_g && (58) \\ &= (\frac{\partial exp(\hat{\delta\boldsymbol{\xi}})}{\partial(\delta\boldsymbol{\xi})}\mathbf{T}_c^o)(1,0:2)\mathbf{n}_g && (59) \\ &= [(\mathbf{G}_0\mathbf{T}_c^o)(1,0:2), ..., (\mathbf{G}_5\mathbf{T}_c^o)(1,0:2)]\mathbf{n}_g, && (60) \end{aligned}$$

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

$$\frac{\partial r_{rot}}{\partial \mathbf{z}} = \mathbf{0}. \tag{61}$$

III. More Qualitative Results

In Fig. 4 we qualitatively show the refinements on 3D pose and shape delivered by our method. The results on each stereo image pair are shown in each two-row block. In the first row we show the initial pose and shape estimates and our results projected onto the left image. In the second row, the initial pose (3DOP) and the estimated pose by our method are shown in the first two images, together with the ground truth 3D point cloud. In the following three images, the 3D point cloud estimated by ELAS and by our method, as well as the ground truth are shown, respectively. As shown in the results, dense stereo matching results become extremely noisy on the strong non-lambertian reflective car surfaces. Our results avoid using such results for recovering the 3D poses and shapes of cars, instead it works directly on images by performing joint silhouette and photometric alignment. While it drastically improves the 3D shape reconstruction, it can also effectively recover the 3D poses of the objects.

More qualitative results are listed in Fig. 5 and 6, where the input images are shown on the left and our results overlaid to the input images are shown on the right.

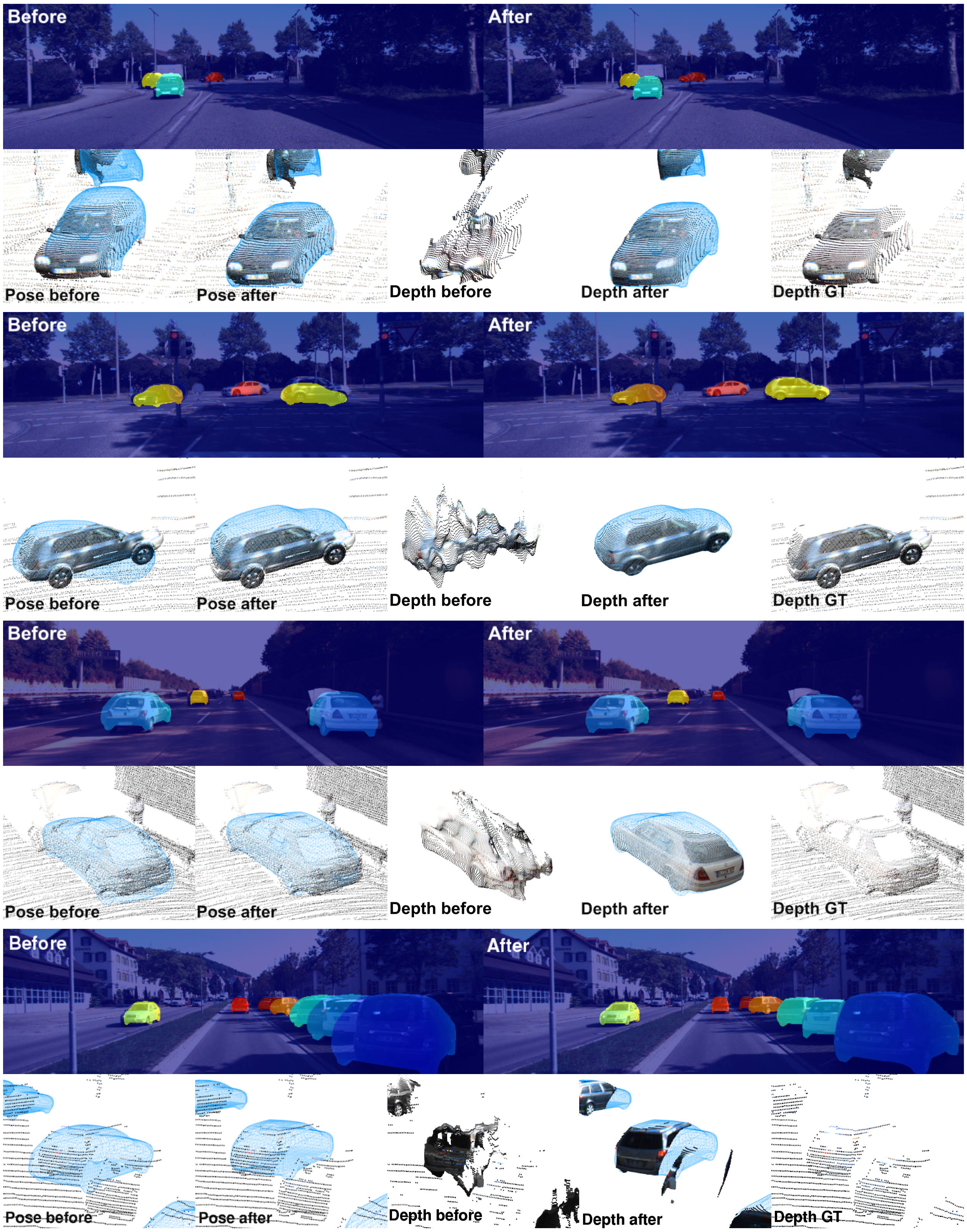

Fig. 4: Qualitative results on 3D pose and shape refinements. Each two-row block shows the results on one stereo frame from the KITTI Stereo 2015 benchmark. Top row: The initial pose and shape (left) and our results (right) projected to the left image. Bottom row: The initial poses estimated by 3DOP 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)

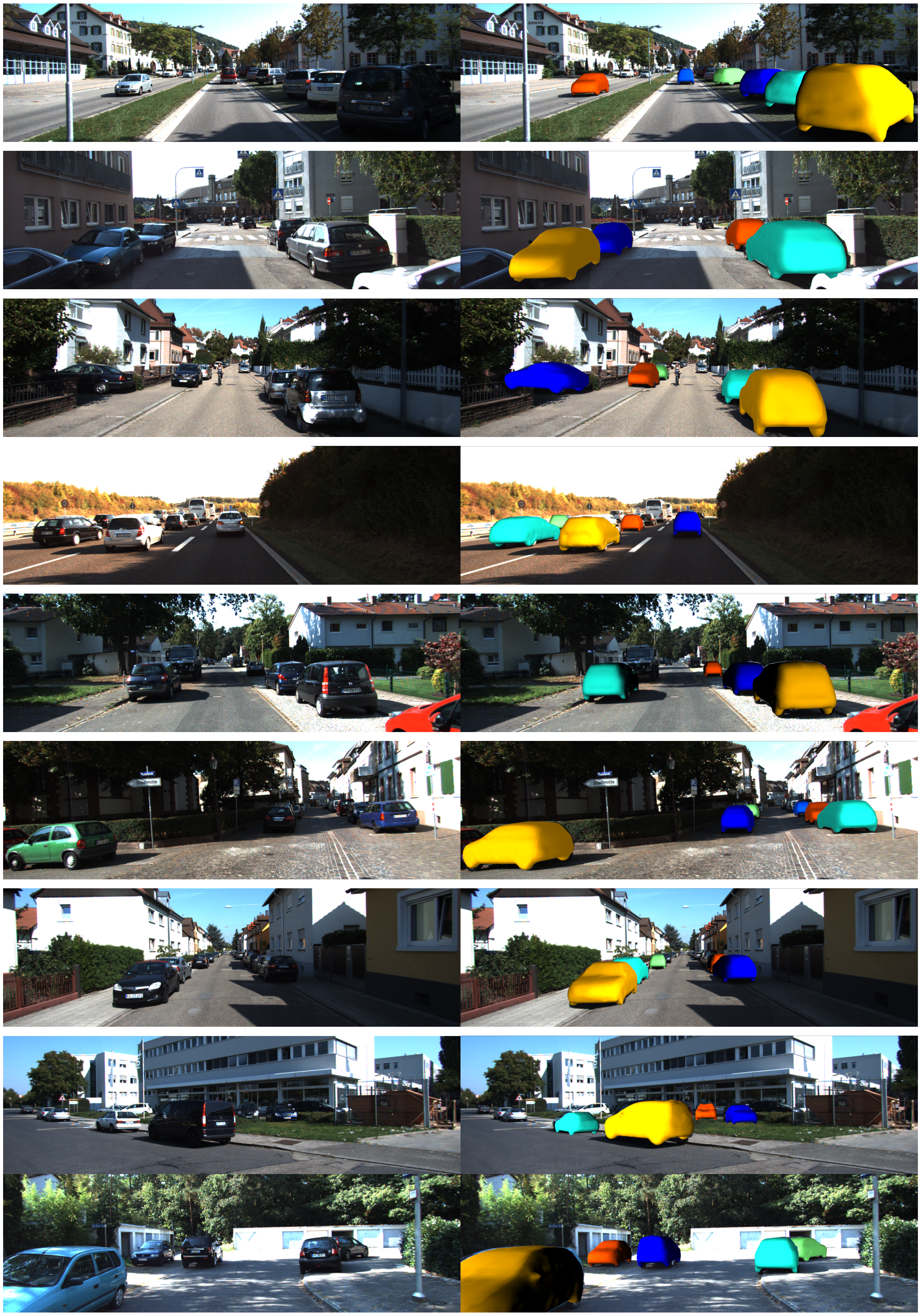

Fig. 5: Qualitative results on 3D pose and shape estimation.

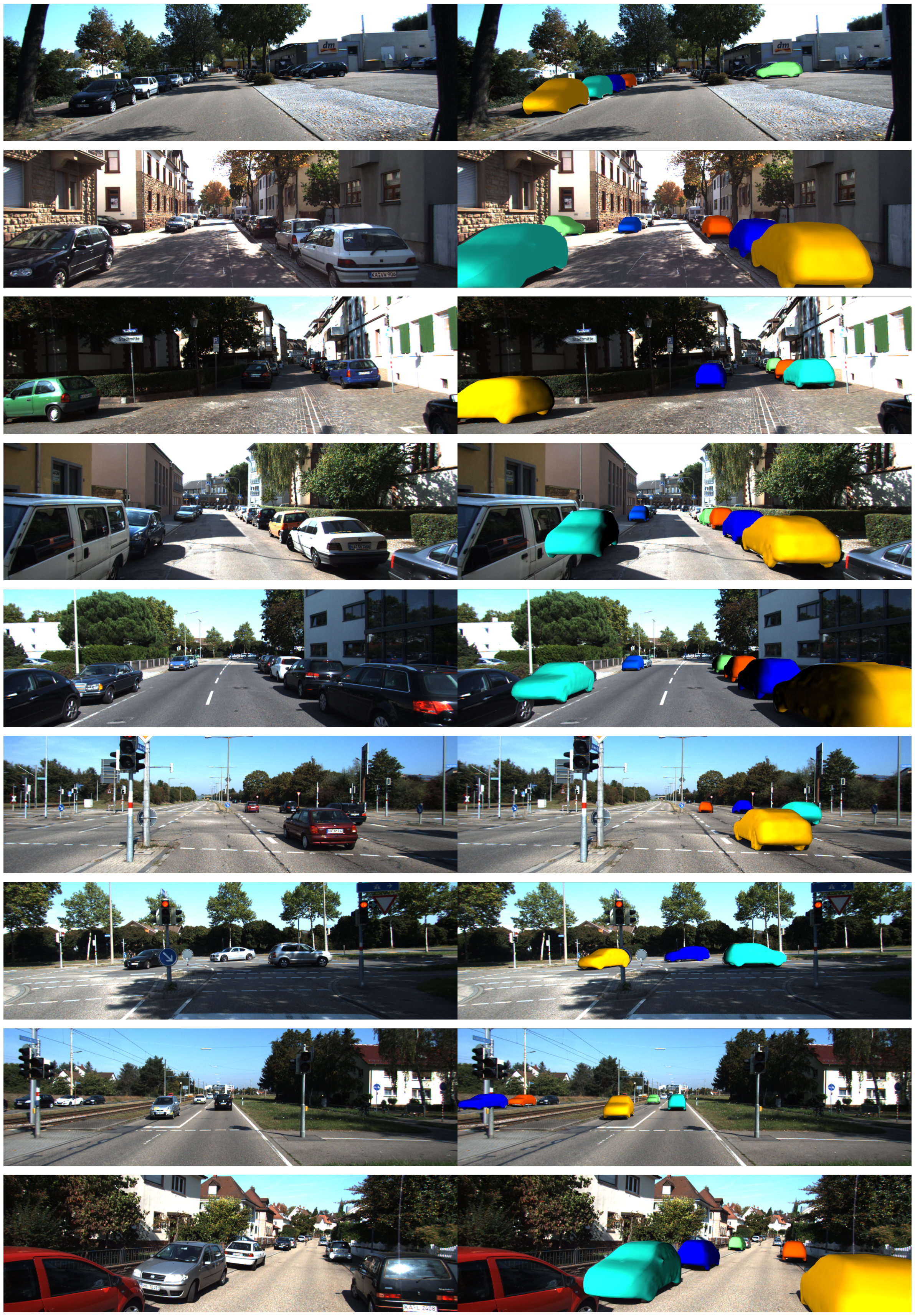
Fig. 6: Qualitative results on 3D pose and shape estimation (cont.).