Multimodal Colored Point Cloud to Image Alignment-Supplementary Material

1. Clipping and Differentiability

The color transformation in section (3.3) can turn color values into values that exceed $[0, 1]$. This could introduce a bias to the comparison with c, which is within bounds. We use a simple clip operation for each color value V,

$$V = \min(\max(V, 0), 1).$$

However, the use of this clipping method results in the clipped values no longer depending on θ_i and therefore, undesirably not being included in the optimization process. To fix this, the original gradient before clipping is used for the optimization process. In this way, we keep the differentiability and gradients while truncating the problematic color values. The clipping operation is then performed on the transformed colors.

2. Transition Proof

A proof of the last transition of Equation (9),

$$\begin{aligned}
f_x^B(x) &= (1-\delta) \cdot h_a(x_j) + \delta \cdot h_a(x_{j+1}) \\
&= \overbrace{\frac{\Delta h_j}{2} - \frac{\Delta h_j}{2}}^{0} + (1-\delta) \cdot h_a(x_j) + \delta \cdot h_a(x_j + 1) \\
&= \frac{\Delta h_j}{2} - \frac{h(x_{j+1}) - h(x_j)}{2} + (1-\delta) \cdot \frac{h(x_{j+1}) - h(x_{j-1})}{2} + \delta \cdot \frac{h(x_{j+2}) - h(x_j)}{2} \\
&= \frac{\Delta h_j}{2} + (1-\delta) \cdot \frac{h(x_j) - h(x_{j-1})}{2} + \delta \cdot \frac{h(x_{j+2}) - h(x_{j+1})}{2} \\
&= \frac{(1-\delta) \cdot \Delta h_{j-1} + \Delta h_j + \delta \cdot \Delta h_{j+1}}{2} \\
&= f_x^A * w(x).
\end{aligned}$$

The last transition can be easily derived from the definition of convolution.

3. Extension to 2D and Bilinear Interpolation

Let us analyze strategies A and B using the 2-D image and bilinear interpolation. First, we study the partial derivative by u with strategy A, we denote $u = u_j + \delta_u$, $v = v_k + \delta_v$,

$$\begin{aligned}
I_u^A(u,v) &= BL(J)_u(u,v)\\
&= \frac{d}{du}\left(\begin{bmatrix}1-\delta_u & \delta_u\end{bmatrix}\begin{bmatrix}J(u_j,v_k) & J(u_j,v_{k+1})\\ J(u_{j+1},v_k) & J(u_{j+1},v_{k+1})\end{bmatrix}\begin{bmatrix}1-\delta_v\\ \delta_v\end{bmatrix}\right)\\
&= \begin{bmatrix}\frac{d}{du}(1-\delta_u) & \frac{d}{du}(\delta_u)\end{bmatrix}\begin{bmatrix}J(u_j,v_k) & J(u_j,v_{k+1})\\ J(u_{j+1},v_k) & J(u_{j+1},v_{k+1})\end{bmatrix}\begin{bmatrix}1-\delta_v\\ \delta_v\end{bmatrix}\\
&= \begin{bmatrix}-1 & 1\end{bmatrix}\begin{bmatrix}J(u_j,v_k) & J(u_j,v_{k+1})\\ J(u_{j+1},v_k) & J(u_{j+1},v_{k+1})\end{bmatrix}\begin{bmatrix}1-\delta_v\\ \delta_v\end{bmatrix}\\
&= \begin{bmatrix}J(u_{j+1},v_k)-J(u_j,v_k) & J(u_{j+1},v_{k+1})-J(u_j,v_{k+1})\end{bmatrix}\begin{bmatrix}1-\delta_v\\ \delta_v\end{bmatrix}\\
&\triangleq \begin{bmatrix}\Delta_a J_{j,k} & \Delta_a J_{j,k+1}\end{bmatrix}\begin{bmatrix}1-\delta_v\\ \delta_v\end{bmatrix}\\
&= (1-\delta_v)\cdot\Delta_a J_{j,k} + \delta_v\cdot\Delta_a J_{j,k+1}
\end{aligned}$$

Using strategy B,

$$\begin{aligned}
I_u^B(u,v) &= BL(J_a)(u,v)\\
&= \begin{bmatrix}1-\delta_u & \delta_u\end{bmatrix}\begin{bmatrix}J_a(u_j,v_k) & J_a(u_j,v_{k+1})\\ J_a(u_{j+1},v_k) & J_a(u_{j+1},v_{k+1})\end{bmatrix}\begin{bmatrix}1-\delta_v\\ \delta_v\end{bmatrix}\\
&= \begin{bmatrix}\overbrace{(1-\delta_u)\cdot J_a(u_j,v_k)+\delta_u\cdot J_a(u_{j+1},v_k)}^{E_1} & \overbrace{(1-\delta_u)\cdot J_a(u_j,v_{k+1})+\delta_u\cdot J_a(u_{j+1},v_{k+1})}^{E_2}\end{bmatrix}\begin{bmatrix}1-\delta_v\\ \delta_v.\end{bmatrix}
\end{aligned}$$

Without the loss of generality, we analyze E_1, Since E_1 is a 1-D function of δ_u, we can use the proof 2,

$$E_1 = (1-\delta_u)\cdot J_a(u_j,v_k)+\delta_u\cdot J_a(u_{j+1},v_k) = \frac{(1-\delta_u)\cdot\Delta J_{j-1,k}+\Delta J_{j,k}+\delta_u\cdot\Delta J_{j+1,k}}{2} = \Delta_a J_{j,k} * w_u$$

Where w_u is a rectangular window function,

$$w_u(x,y) = \begin{cases}0.5, & -1\le x\le 1, y=0\\ 0, & \text{otherwise.}\end{cases}$$

It is easy to show that similarly $E_2 = \Delta J_{j,k+1} * w_u$. Thus, from linearity,

$$I_u^B(u,v) = I_u^A(u,v) * w_u$$

And similarly for v,

$$I_v^B(u,v) = I_v^A(u,v) * w_v$$

Where w_v is a rectangular window function,

$$w_v(x,y) \quad = \quad \begin{cases}0.5, & x=0, -1\le y\le 1\\ 0, & \text{otherwise.}\end{cases}$$

4. Synthetic Color Transformation

To simulate multimodality in the synthetic experiment, we apply a series of color effects to the RGB images in the ICL-NUIM dataset. The point clouds with the original colors are then aligned with the images whose colors have been modified. The set of effects chosen was,

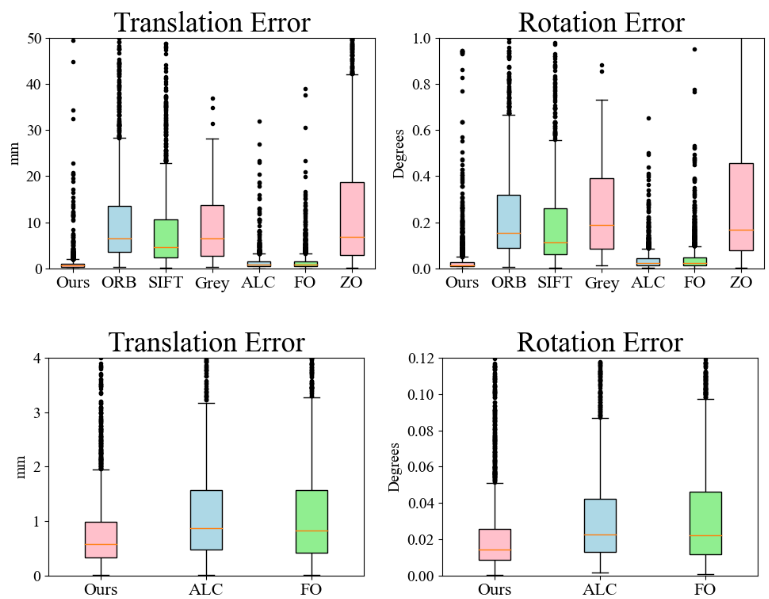
1. Apply random color transformation of brightness, contrast, saturation, and hue using Pytorch ColorJitter with a random range of $[0, 0.4]^3$ and $[0, 0.06]$, respectively.

2. Apply gamma correction with a randomly selected gamma from $[0.5, 1]$ or $[1, 2]$.
3. Simulate different point spread functions and sensor properties by applying a Gaussian blur to the image using a random $[0, 0.75]$.

The next figure shows an example of an original image of the ICL-NUIM dataset (left) and the same image after applying such a random color transformation (right),

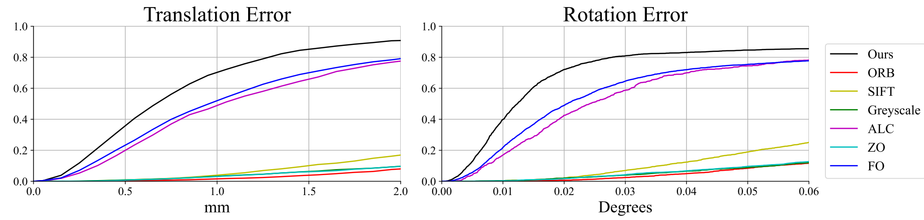
5. Synthetic Box Plot Results

Section 4.1 presents cumulative normalized histograms of translation and rotation errors for synthetic data experiments. Another informative way to present the results and effectiveness of our method is to present them in a boxplot. The top row contains all comparisons from Section 4.1 , while the bottom row contains an enlarged version of the best performing methods. The proposed method with second-order polynomial color alignment outperforms the other methods,



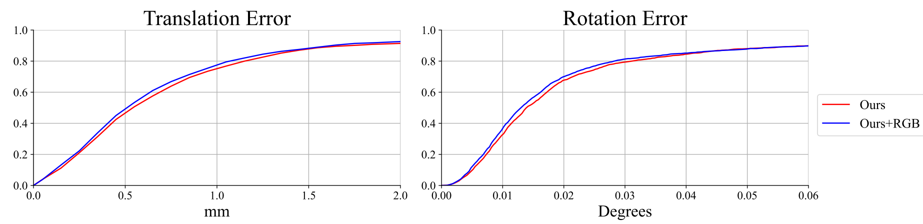
6. Robustness to Initialization

To test the robustness of our method to initializations, the same experiment as in Section 4.1 is performed with larger initialization values. Instead of aligning each image in the ICL-NUIM datasetwith the subsequent image $I^{(i+1)}$ it is aligned with the non-consecutive image $I^{(i+3)}$. As can be seen, this scenario maintains our favorable results,



7. Third Degree Term

As explained, in the study of Hong et al. for camera colorimetric characterization the third-order RGB term is added as an additional dimension of the second-order polynomial kernel. To test the advantage of adding this term, we perform the same experiment as in Section 4.1. The methods tested are the method proposed in the paper and the same method with the additional term. As can be seen in the next figure, adding such a term has no significant effect on the results,



8. Real Data Results

In the next figure, we present more comparisons between edges from camera images and edges extracted from rendered point cloud images, as in Figure 5,