OmniZoomer: Learning to Move and Zoom in on Sphere at High-Resolution

Zidong Cao[1] Hao Ai[1,2*] Yan-Pei Cao[2] Ying Shan[2] Xiaohu Qie[2] Lin Wang[1,3†]
[1] AI Thrust, HKUST(GZ) [2]ARC Lab, Tencent PCG [3]Dept. of CSE, HKUST
caozidong1996@gmail.com, hai033@connect.hkust-gz.edu.cn, caoyanpei@gmail.com
yingsshan@tencent.com, tigerqie@tencent.com, linwang@ust.hk

Abstract

Due to the lack of space in the main paper, we provide more details of the dataset construction and spherical resampling module in the supplementary material. Specifically, in Sec. 1, we provide more details of the parameters of Möbius transformations and dataset construction. Sec. 2 explains more details of the angle calculation in the spherical resampling module. Sec. 3 presents more experimental results and discussion. Lastly, Sec. 4 presents additional visual comparisons between OmniZoomer and other methods on different datasets.

1. Details of the ODIM dataset

We synthesize the proposed ODI-Möbius (ODIM) dataset with various Möbius transformations. As we aim to move and zoom in on ODIs, the parameters of Möbius transformations in Eq. 3 in the main paper are determined by rotation angles α, and zoom level s, respectively. Given a specific rotation angle α along the rotation axis $L = (l, m, n)$, the parameters of Möbius transformations can be described as follows:

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} \cos(\frac{\alpha}{2}) + jn\sin(\frac{\alpha}{2}) & (m - jl)\sin(\frac{\alpha}{2}) \\ (-m - jl)\sin(\frac{\alpha}{2}) & \cos(\frac{\alpha}{2}) - jn\sin(\frac{\alpha}{2}) \end{pmatrix}. \tag{1}$$

Specifically, when performing horizontal rotation with angle β, the rotation axis can be represented with $L = (0, 0, 1)$, and the parameters of Möbius transformations can be simplified as follows:

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} \cos(\beta) + j\sin(\beta) & 0 \\ 0 & 1 \end{pmatrix}. \tag{2}$$

Similarly, when performing vertical rotation with angle γ, the rotation axis can be represented with $L = (0, 1, 0)$, and the parameters of Möbius transformations can be simplified as follows:

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} \cos(\frac{\gamma}{2}) & \sin(\frac{\gamma}{2}) \\ -\sin(\frac{\gamma}{2}) & \cos(\frac{\gamma}{2}) \end{pmatrix}. \tag{3}$$

An arbitrary rotation can be divided into horizontal rotation and vertical rotation. In addition, Möbius transformations can be composed to give a new Möbius transformation. Therefore, we can achieve arbitrary rotation or movement on ODIs with horizontal rotation angle β and vertical rotation angle γ.

For zoom with level s, if the pole is North pole $(0, 0, 1)$, the parameters of Möbius transformations can be described as follows:

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} s & 0 \\ 0 & 1 \end{pmatrix}. \tag{4}$$

Furthermore, if the pole is an arbitrary point (x_0, y_0, z_0), we can rotate the sphere so that the arbitrary pole coincides with the north pole $(0, 0, 1)$, and then apply the standard stereographic projection (STP) and its inversion in Eq. 2,4 in the main

*Intern at ARC Lab, Tencent PCG.
†Corresponding author (e-mail: linwang@ust.hk)

paper. To rotate the sphere, we can employ quaternions. The quaternion of $(0, 0, 1)$ can be written as: $v = [0, i0, j0, k1]$, and the quaternion of (x_0, y_0, z_0) can be written as: $v' = [0, ix_0, jy_0, kz_0]$. To estimate the rotation between two quaternions, we can utilize the equation: $v' = qvq^{-1}$, where $q = [\cos(\frac{1}{2}\theta), \sin(\frac{1}{2}\theta)u]$, u is a unit axis for rotation, and θ is the rotation angle along the unit axis u. To estimate the axis u, we can compute the cross product of v and v': $v \times v'$. The rotation axis u can be expressed with $\frac{v \times v'}{||v \times v'||}$. Moreover, considering that v and v' are unit vectors, $||v \times v'|| = \sin(\theta)$. Meanwhile, we can compute $\cos(\theta) = v \cdot v'$, and utilize that $\cos(\theta) = 2\cos^2(\frac{\theta}{2}) - 1$ to simplify q as follows:

$$q = [\sqrt{\frac{v \cdot v' + 1}{2}}, \frac{1}{\sqrt{2(v \cdot v' + 1)}} v \times v']. \tag{5}$$

Finally, we introduce the parameters of the random distributions we use in Tab. 1. We uniformly sample horizontal rotation angle β and vertical rotation angle γ, and zoom level s. Note that the ODIs vary greatly when rotated vertically with large angles. Empirically, to make the training process stable, we constrain γ into $(-\frac{\pi}{4}, \frac{\pi}{4})$. Similarly, with too large or small (nearly zero) zoom levels, some specific regions are enlarged too much and make the other regions almost disappear from the ODIs. In this case, we also constrain s into $(0.5, 5)$. We do not randomly sample the pole because there are overlapping functions between rotation and changing pole. During validation and testing, we assign a fixed Möbius transformation matrix for each ODI. We also test on SUN360 dataset with the same Möbius transformation matrices in the testing set of ODIM dataset.

Variable	Sampling distribution	Note
β	$\mathcal{U}(0, 2\pi)$	Horizontal rotation
γ	$\mathcal{U}(-\frac{\pi}{4}, \frac{\pi}{4})$	Vertical rotation
s	$\mathcal{U}(0.5, 5)$	Zoom

Table 1: Random variable sampling distributions for the Möbius transformation matrix.

2. Spherical Resampling Module

We first resample p_0, p_1 to p_{01}. The angle α_{01} subtended by p_0 and p_1 is calculated as follows:

$$\alpha_{01} = \arccos(p_0 \cdot p_1), \tag{6}$$

where $(p_0 \cdot p_1)$ denotes dot production between p_0 and p_1. To make the longitude of $p_{01} = (\theta_{01}, \phi_{01})$ same with the query pixel $q = (\theta, \phi)$, *i.e.*, $\theta_{01} = \theta$, the resampling weight t_{01} in Eq. 6 in the main paper needs to be selected specifically. According to Eq. 1,6 in the main paper, we can get the following equation:

$$\frac{\sin(1 - t_{01})\alpha_{01}}{\sin \alpha_{01}} \begin{pmatrix} x_0 \\ y_0 \\ z_0 \end{pmatrix} + \frac{\sin t_{01}\alpha_{01}}{\sin \alpha_{01}} \begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix} = \begin{pmatrix} \cos(\phi_{01})\cos(\theta) \\ \cos(\phi_{01})\sin(\theta) \\ \sin(\phi_{01}) \end{pmatrix}. \tag{7}$$

The unknown parameters in Eq. 7 are t_{01} and ϕ_{01}. Also, Eq. 7 contains two equations because p_{01} is located on the unit sphere. Therefore, t_{01} can be obtained from Eq. 7. The complicated formula is mainly due to p_0, p_1, and p_{01} do not have the same longitude or latitude. If we assume that p_0, p_1, and p_{01} have the same latitude, the calculation of t_{01} can be simplified into:

$$t_{01} = \frac{\theta - \theta_0}{\theta_1 - \theta_0}. \tag{8}$$

The calculation of t_{23} is similar with that of t_{01}. Furthermore, as p_{01}, p_{23}, and q are located on the same longitude, the resampling weight t_q is convenient to calculate as follows:

$$t_q = \frac{\phi - \phi_{01}}{\phi_{23} - \phi_{01}}. \tag{9}$$

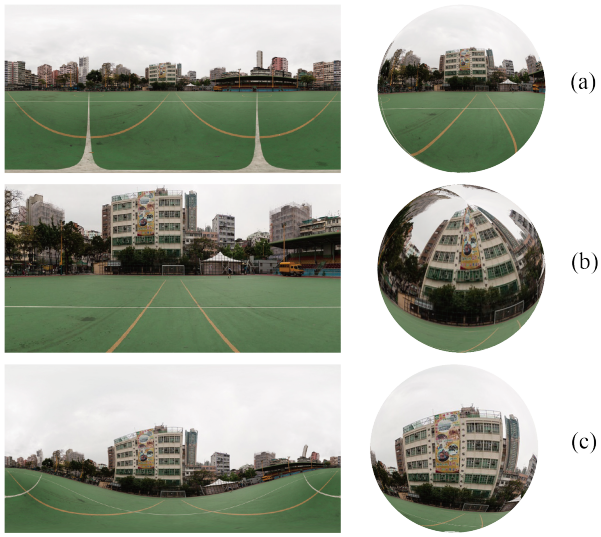

Figure 1: Visual comparison of zooming in with level 3. (a) The original ODI (b) Uniformly zooming in on the ODI. (c) Applying Möbius transformation on the ODI. As shown in the ERP images (left column) and spherical images (right column), uniformly zooming in causes incorrect spherical distortion, while the result from Möbius transformation is conformal and reasonable.

3. More Discussion

About zooming in uniformly. From Fig. 1(b), we can see that zooming in on ERP images uniformly not only loses some pixels from the original ERP images, but also leads to shape distortion when projected onto the sphere surface. This is because uniformly zooming in on ERP images is not conformal. In contrast, as shown in Fig. 1(c), Möbius transformation preserves the angles on the sphere surface effectively, *e.g.*, the rectangular shape of the advertising board. Möbius transformation is the only conformal and bijective transformation on the sphere.

About computation costs and time consuming of each module. In Tab. 2, firstly, the feature extractors and upsampling modules are directly from the backbones with no change. Then, our *core index generation and resampling modules* have minor computational costs compared with backbones. For decoder, we add three ResBlocks, which obtain 0.02 WS-PSNR improvement, but with relatively heavy computational costs.

	Ours-EDSR-baseline		Ours-RCAN	
	Computation	Time	Computation	Time
Feature Extractor	39.9	0.006	496.9	0.079
Up Sampling	101.5	0.010	101.5	0.010
Index Generation	<0.1	0.050	<0.1	0.050
Spherical Resamp.	0.6	0.029	0.6	0.029
Decoder-0	3.6	0.002	3.6	0.002
Decoder-3	467.5	0.066	467.5	0.066
Sum-0	145.6	0.097	602.6	0.170
Sum-3	609.5	0.161	1066.5	0.234

Table 2: MACs/G and time (s) of each module.

4. More Visual results

From Fig. 2,3,4,5,6,7, we can find that OmniZoomer reconstructs clearer textural and structural details, *e.g.*, the branches in Fig. 2, the nearly ninety degree's corner of the object in Fig. 3, the thin handrails in Fig. 4, the windows in Fig. 5. Also, in Fig. 6 and Fig. 7, we can see that other methods' predictions have jagging and blurring artifacts. Inversely, our OmniZoomer predicts more continuous edges of the tables and windows, showing the effectiveness of the HR feature representation and spherical resampling.

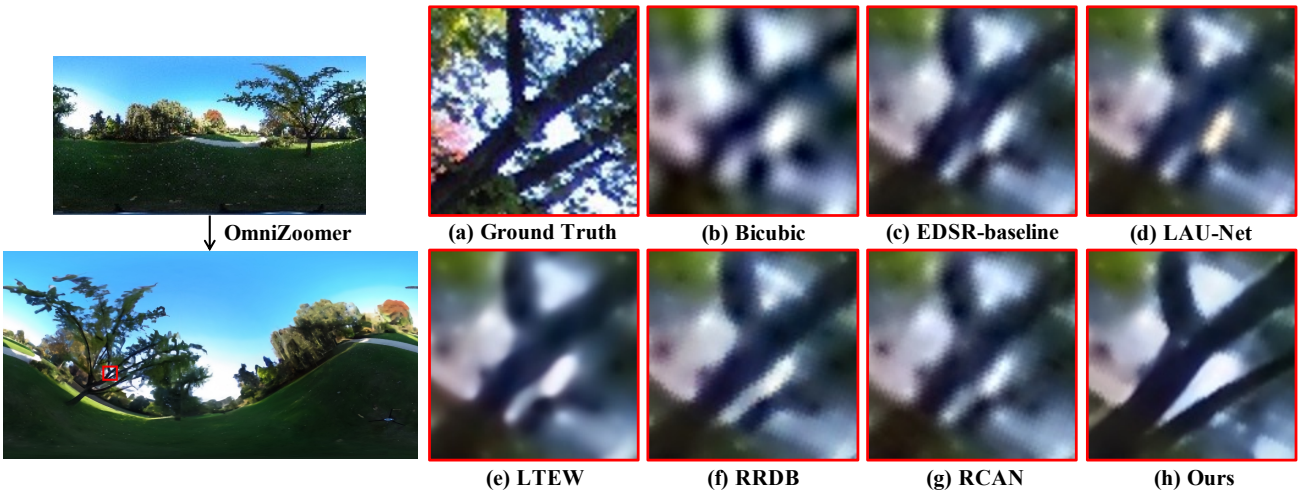

Figure 2: Visual comparisons of different methods for Möbius transformation with $\times 8$ up-sampling factor on ODI-SR dataset.

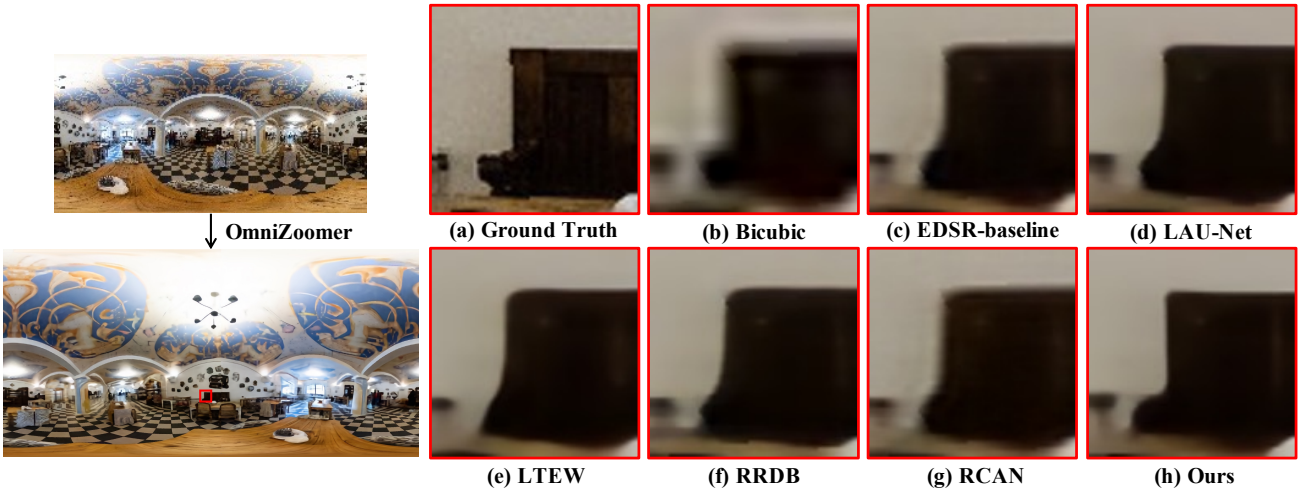

Figure 3: Visual comparisons of different methods for Möbius transformation with $\times 8$ up-sampling factor on ODI-SR dataset.

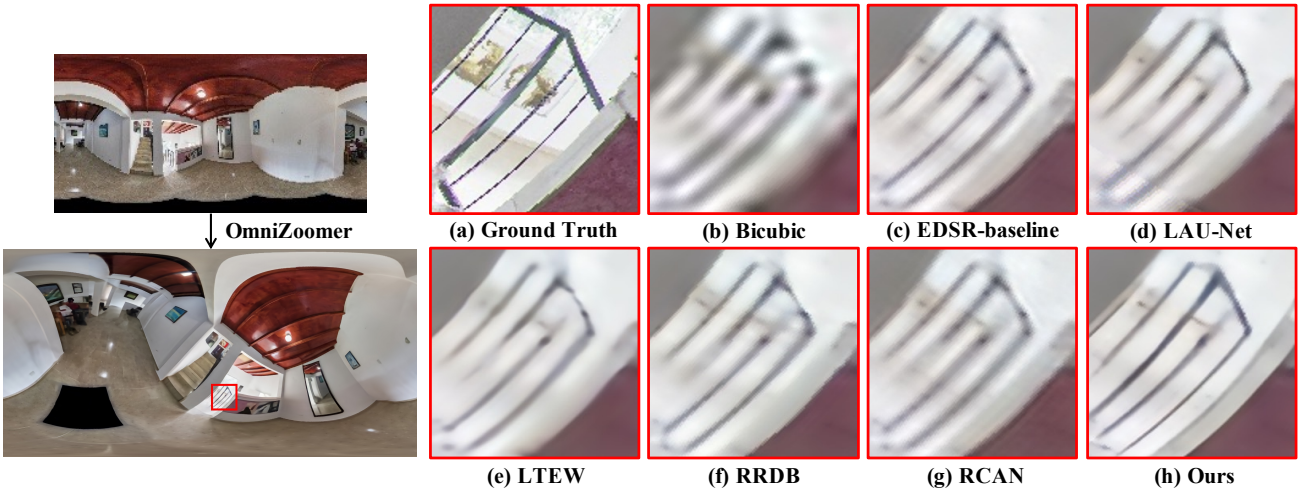

Figure 4: Visual comparisons of different methods for Möbius transformation with $\times 8$ up-sampling factor on ODI-SR dataset.

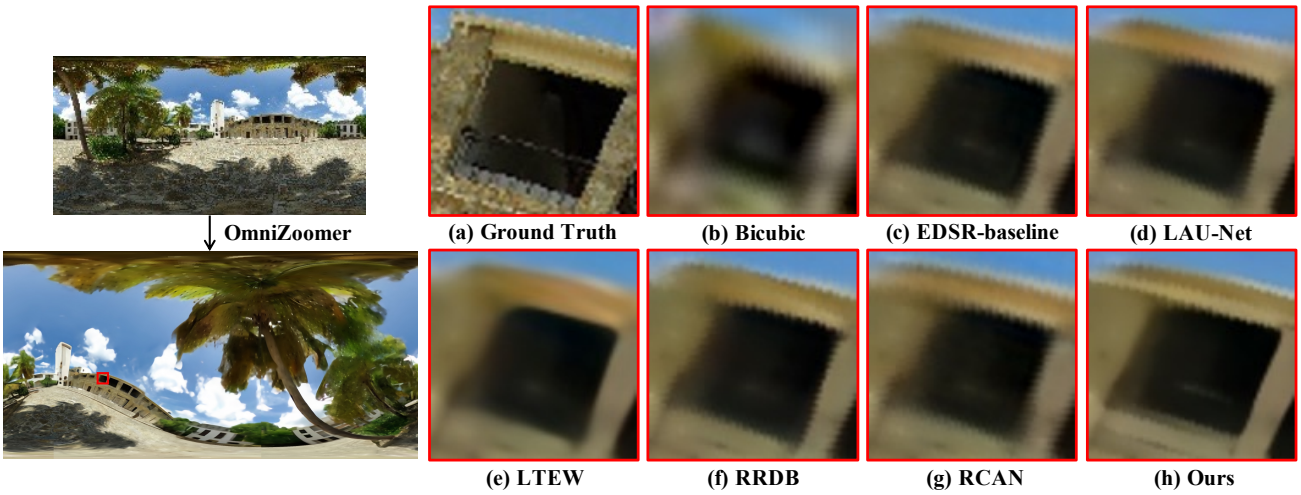

Figure 5: Visual comparisons of different methods for Möbius transformation with $\times 8$ up-sampling factor on ODI-SR dataset.

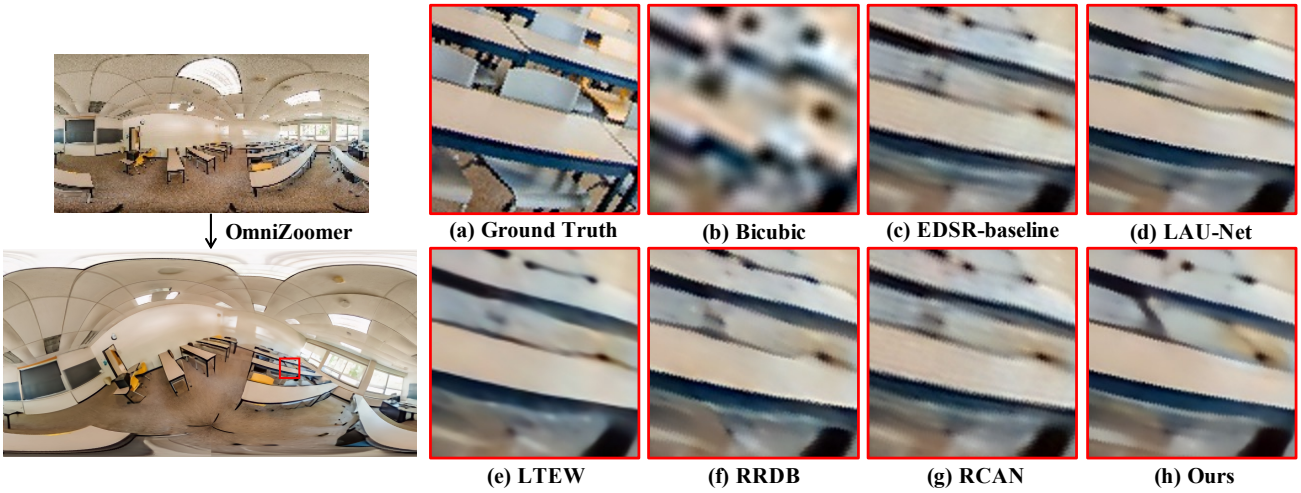

Figure 6: Visual comparisons of different methods for Möbius transformation with $\times 8$ up-sampling factor on SUN360 dataset.

Figure 7: Visual comparisons of different methods for Möbius transformation with $\times 8$ up-sampling factor on SUN360 dataset.