Distortion Reduction for Off-Center Perspective Projection of Panoramas

Chi-Han Peng
College of AI
National Yang Ming Chiao Tung University
pengchihan@nycu.edu.tw

Jiayao Chang
ByteDance Game Department
ByteDance Ltd.
jiayao.zhang@bytedance.com

Chia-Chia Chen
College of AI
National Yang Ming Chiao Tung University
s1073736@gmail.com

Yun-Wei Lin
College of AI
National Yang Ming Chiao Tung University
b2011526@gmail.com

***Abstract*—A key assumption of perspective projection is that linear features in 3D shall remain linear after being projected to the 2D screen. This assumption is preserved when we draw a spherical panorama perspectively in arbitrary viewing directions and field-of-views as long as the camera position is fixed at the center. However, when the camera moves away from the center, barrel-like distortions appear and such assumption breaks. To address this issue, we propose modifications to the equirectangular-to-perspective (E2P) projection that significantly reduced distortion of linear features when the camera position is away from the center. We compared with other common methods that aim to augment panoramas with 3D information including: 1) building a point cloud augmented with per-pixel depth and 2) building a 3D room model according to room layouts, and found that our method produced rendering results with less linearity distortion measured quantitatively and qualitatively.**

***Index Terms*—Virtual Reality, Equirectangular Projection, Perspective Projection**

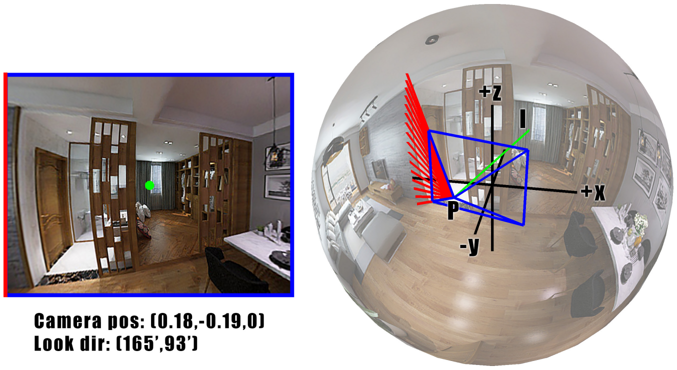

Fig. 1. An off-center E2P projection. Point P is the camera center and point I is the intersection of the looking ray (green) and the unit sphere. We use a right-handed system with +z direction at azimuth=0° and zenith=0° and +x direction at azimuth=0° and zenith=90°. The viewing pyramid is drawn in blue. Red lines show the left edge of the image plane that is projected to a curved geometry in the 3D scene due to distortions.

I. Introduction

A spherical panorama stores incoming ray intensities toward a fixed camera position in all possible directions. Commonly, the directions are sampled on a sphere (centered at the camera position) in a 2D equirectangular format. Using just a single panorama, one can render accurate perspective projections from the camera point in arbitrary viewing directions and field-of-views through sampling strategies commonly known as the equirectangular-to-perspective (E2P) projection. In VR and virtual tour applications, this is known as the 3DoF (i.e., rotations along three axes) of a perspective camera in a virtual environment, albeit at a fixed position.

It is actually straightforward to render perspective projections from an *off-center* position, although the results would not be correct anymore. This is done by mapping the panorama onto a spherical mesh centered at the camera position through projective texture-mapping and then draw perspective views using standard perspective camera models at the off-center positions. An analytical form of this off-center E2P projection is described in Section III-A. The resulting rendering (see the accompanying video for examples) would be still ”intuitive” in general, i.e., objects would become bigger when the camera moves closer, and move to the right when the camera moves to the left, and vice versa. However, the rendering would contain two major sources of errors. First, barrel-like/spherical distortions. Second, the lack of true 3D effects such as parallax, occlusion, and disocclusion (i.e., re-appearing of previously occluded objects).

There already exist numerous previous work that aimed to synthesize parallax effects and occlusion/disocclusion for perspective images or videos [1]–[4]. However, to our best knowledge, no prior published methods exist to specifically address the barrel-like distortion issue. Toward this goal, we reinterpret the issue as the breaching of the ”3D-to-2D linearity preservation” assumption of perspective projections. That is, *linear lines in 3D shall remain linear after being perspectively projected to a 2D image plane*. We propose two modifications to the standard E2P projection scheme that effectively reduce the distortion of visual linearity, as follows:

First, in order to preserve the linearity of vertically straight lines in 3D in the 2D view, we re-project the panorama to a *cylinder* so that all vertically straight lines in 3D are now guaranteed to be projected back to straight lines in 2D. While cylindrical projection is a well known concept, we provide an analytical proof to back up the claim. Second, we propose novel *computational dolly-zoom* effects to find a set of alternative camera position and field-of-view for the same viewable region that would minimize linearity distortions. In summary, the first measure completely eliminates all distortions of vertical lines while the second measure can reduce distortions in both horizontal and vertical directions.

To quantitatively and qualitatively measure the performance of linearity distortions of our approaches, we used HAWP [5] to detect straight line segments in the rendered perspective views. We compared our results with the vanilla off-center E2P projection scheme and two common ways to estimate rough 3D models from panoramas: 1) building a point cloud-like mesh with per-pixel depths (ground-truth or predicted by neural networks), and 2) building a 3D room model using room layouts estimated by deep learning methods such as LED2-Net [6] textured by the panorama. We find that our method produced renderings with significantly lower linearity distortions measured by quantitatively counting the straight line segments after E2P projections and qualitative observations.

Our contributions are summarized as follows:

- We provide a theoretical proof to support the claim that cylindrical projections eliminate all linearity distortions of vertically straight features in 3D in off-center perspective views.
- We propose a novel computational dolly-zoom effect scheme to generate perspective views of moving cameras in which linearity distortions are significantly reduced.
- We propose a novel way to to quantitatively measure the linearity distortions of off-center perspective projections of panoramas based on HAWP [5], a robust neural network-based straight line segment detection method.

II. Related Work

A. Panoramic 3D modeling and datasets

3D modeling and reconstruction of real-world scenes based on spherical panoramic (also called ”360°” in commercial settings) image inputs is a popular and fast advancing research field in recent years. Key tasks are now mainly solved by deep learning, including depth estimation [8], [8]–[14], room layout estimation [15]–[19], object detection and segmentation [20], [21], and structure-from-motion (SfM) tasks such as the registration of multiple panoramas [22], [23]. A number of panoramic image datasets have been produced to aid the research. Matterport3D [24] and Standford2D3D [25] are real-world large-scale RGB-D datasets of diverse indoor scenes. Structure3D [7] and SunCG [26] dataset are synthetic datasets providing realistically rendered indoor RGB-D images and annotations of 3D structures. Gibson [21] is a real-world dataset that includes high-quality RGB panoramas, global camera poses, and 3D meshes. Finally, Zillow Indoor Dataset [27] is a massive dataset of panoramas taken in real-world, largely unfurnished houses that also come with room layouts, camera poses, and bounding boxes of windows, doors and openings.

B. Novel View Synthesis with Panoramas

The task of rendering perspective views of panoramas from their original camera positions is a special case of *novel view synthesis* problems. We limit the scope of discussions to methods rely on panoramic image inputs. Layered Depth Images (LDI) [28] and Multiplane Images (MPI) [29] are used as image-based representations for novel view synthesis, but for large translations, they might lack sufficient information to render correctly. Multi Depth Panoramas (MDPs) [30] and PerspectiveNet [31] comprise of multi-RGBDα images for high-quality and efficient novel view generations. Hedman et al. [32] input burst of aligned color-and-depth photos to generate 3D panorama, and their 3D effects could also interact with the scene. Jin et al. [33] and Zeng et al. [34] leverage the geometric structures of a 360° indoor image for depth estimation. Attal et al. [35] simultaneously learn depth and occlusions via a multi-sphere image representation, which could handle occluded regions in dynamic scenes. With 46 input light field videos, Broxton et al. [36] present a system that is able to reproduce view-dependent reflections, semi-transparent surfaces, and near-field objects. Tobias et al. introduce OmniPhotos [37] for quickly and casually capturing 360° VR panoramas, and improve the visual rendering quality by alleviating distortion using a novel deformable proxy geometry. Serrano et al. [38] present a device which enable head motion parallax in 360°video. Other works extend these approach to point clouds, Aliev et al. [39] present a point-based approach to generate novel views of the scene. Voxel grid-based methods such as DeepVoxels [40] encodes the view-dependent appearance of a 3D scene as 3D voxels. Implicit function-based methods such as Sitzmann et al [41] propose a continuous, 3D structure-aware scene representation that encodes both geometry and appearance.

Overall, most existing methods reply on input data that consists of multiple panoramas (taken at different camera positions), often as panoramic videos shoot either from a single moving 360° camera or an array of fixed or moving cameras (capturing rigs). One exception is [42], in which they rely on deep learning (DL)-based methods to directly synthesize novel views from just a single panorama. In comparison, our method takes an extremely lightweight approach to the problem (no DL training/inference nor video-sequence inputs are needed, and is very computationally light), with a focus on tackling linearity distortions.

III. Method

In this section, we first describe the analytical form of the vanilla off-center E2P projection in Section III-A. We then describe our two measures to reduce barrel distortions in Section III-B and III-C.

A. Off-center E2P projection

We first review annotations. We assume the panorama is mapped to a unit sphere centered at the origin. We assume the perspective camera's position is $P = (p_x, p_y, p_z)$, looking direction is $Dir = (dir_x, dir_y, dir_z)$, and up direction is $Up =$

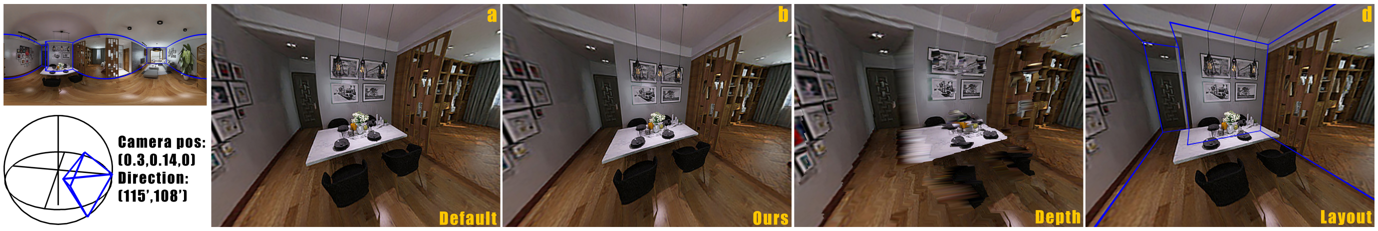

Fig. 2. (a) Off-center perspective projections of a panorama contain barrel distortions and break the ”linearity” assumption of perspective projection that straight lines in 3D remain straight in 2D projections. (b) With our distortion-reduction measures, linearity distortions are significantly reduced. In comparisons, existing 3D-modeling methods such as augmenting the panorama with (c) ground-truth depth or (d) predicted room layouts (using LED2-Net [6]) still do not adequately preserve the linearity assumption. For examples, observe the broken lamp wires and doorway in (d). We show the panorama and room layout on upper-left and the camera position and looking direction on lower-left. The panorama and the ground-truth depth are from the Structure3D dataset [7].

(up_x, up_y, up_z). Up is orthogonal to Dir. The shape of the rectangular image plane in 3D can then be described by two angles $fovx$ and $fovy$, i.e., the horizontal and vertical field-of-view angles of the pyramid formed by the camera center and the image plane. Therefore, the width and height of the image plane are $width = near \cdot tan(fovx/2) \cdot 2$ and $height = near \cdot tan(fovy/2) \cdot 2$, respectively. $near$ is the distance between the camera position and the image plane. We denote the ”left” direction as $Left = Up \times Dir$, × denotes right-handed cross product. See Figure 1 for an example.

To draw a perspective projection, we compute the colors of every 2D positions on the image plane (e.g., pixels). For 2D position (X, Y), $0 \leq X, Y \leq 1$, X is left to right direction and Y is top to bottom direction, we first derive the corresponding 3D ray direction, $Ray = (ray_x, ray_y, ray_z)$, as:

$$|Dir \cdot near + Left \cdot (0.5 - X) \cdot width + Up \cdot (0.5 - Y) \cdot height|,$$

which can be shortened as:

$$\begin{aligned} |Dir + Left \cdot (1 - 2\dot{X}) \cdot tan(fovx/2) + \\ Up \cdot (1 - 2\dot{Y}) \cdot tan(fovy/2)|. \end{aligned} \quad (1)$$

Next, we find the spherical coordinate, (θ, ϕ), of the point I on the unit sphere that intersects with Ray from the camera position. Recall that θ is the zenith (angle from the +z axis), ϕ is the azimuth (angle of counterclockwise rotation along +z-axis from the +x axis). [...] Finally, the corresponding 2D coordinate in equirectangular projection can be trivially derived as $(\frac{\phi}{2\pi}, \frac{\theta}{\pi})$, which is used to sample a color in the panorama.

B. Cylindrical projection

We can make the assumption that vertical features in the scene, such as columns and wall intersections, are nearly always mapped to vertical lines in the panorama. There are two reasons to support this assumption. First, modern 360° cameras have strong image stabilization features to align the shoot panoramas to be in nearly upright position, i.e., the -z direction in 3D is aligned to the direction of gravity. Second, there exist post-processing algorithms to further transform a panorama to make it in the upright position (e.g., the ”camera rotation pose alignment” in [17]). However, the assumption doesn’t mean that vertical features in the scene are mapped to vertical lines in 3D through the equirectangular projection. Instead, they are mapped to *meridians on the unit sphere*, which are in fact curves in 3D. As a result, they would be perspectively drawn as curved lines in 2D when the camera position is off the origin.

We point to a simple solution to ensure that vertical features in the scene are always perspectively drawn as straight lines in 2D even when the camera is not at the origin: *projecting the panorama to a cylinder*. We assume the cylinder has a radius of 1, is centered at the origin, is in the ”upright” position (i.e., axis is aligned to the +z axis), and has an infinite height. We review general *equirectangular-to-cylinder (E2C)* projection as follows.

Equation 1 again describes the 3D ray direction, Ray, of a 2D position (X, Y) on the image plane of the camera. Its intersection point with the cylinder, I^c, can be calculated by solving t' in:

$$I^c = P + t' \cdot Ray, |I^c_x + I^c_y| = 1,$$

which can be solved by having:

$$\begin{aligned} a' &= Ray_x^2 + Ray_y^2, \\ b' &= 2 \cdot (P_x \cdot Ray_x + P_y \cdot Ray_y), \\ c &= P_x^2 + P_y^2 - 1. \end{aligned}$$

where P denotes the camera position. t' equals $(-b' + \sqrt{b'^2 - 4a'c'})/2a'$. Next, convert $|I^c|$ to spherical coordinate (θ, ϕ), and use it to sample a color in the panorama.

To realize the above projection model in the OpenGL rendering pipeline, render a scene with a 3D cylinder. The cylinder is textured using the panorama as a projective texture.

We now propose the following lemma to support the claim that cylindrical projections necessarily preserve linearity of vertical straight lines in 3D in the 2D perspective projections:

Lemma III.1. *A vertical feature in 3D in the scene is perspectively drawn as a 2D straight line under a cylindrical projection.*

Proof. Recall that a vertical feature in 3D is necessarily drawn as a vertical line in the 2D panorama under the default equirectangular projection, which is then mapped to a subset

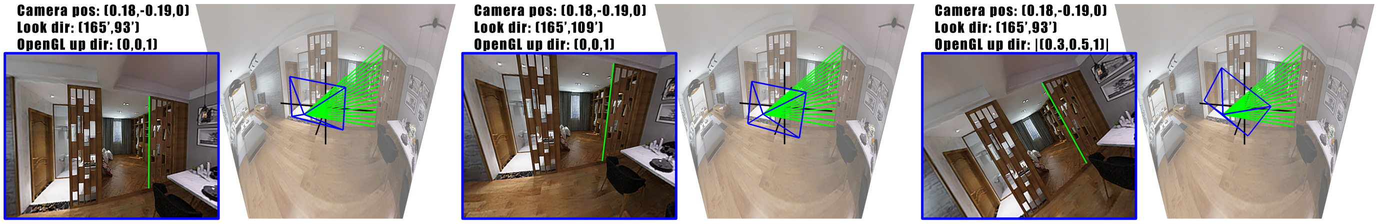
Camera pos: (0.18,-0.19,0)
Look dir: (165',93')
OpenGL up dir: (0,0,1)
Camera pos: (0.18,-0.19,0)
Look dir: (165',109')
OpenGL up dir: (0,0,1)
Camera pos: (0.18,-0.19,0)
Look dir: (165',93')
OpenGL up dir: |(0.3,0.5,1)|

Fig. 3. Off-center cylindrical projections. (a): A rendering using the same camera position and looking direction as in Figure 1. All vertical features in the scene are now drawn as straight lines. (b): After the camera is pitched downward. The vertical features are still drawn as 2D straight lines, but not vertical. (c): After the camera is rolled clockwise. Again, vertical features are drawn as 2D straight lines.

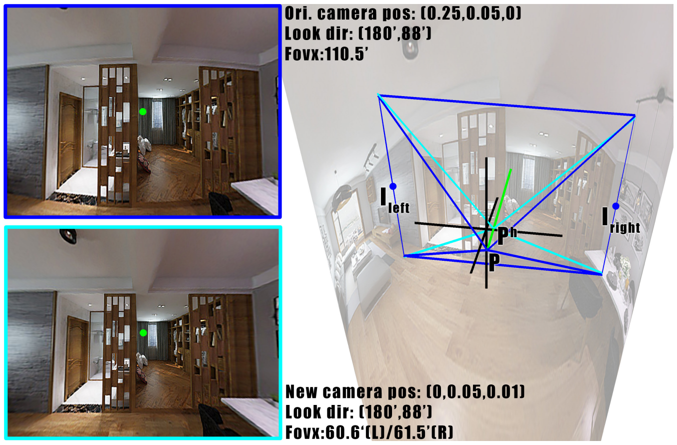

Fig. 4. Inspired by the dolly-zoom effects in cinematography, we find an alternative set of camera position and FOV angles that with which the perspective camera draws roughly the same subset of the panorama but with less distortions. The original camera is shown in blue and the adjusted camera is shown in cyan. Observe that the two drawn images have roughly the same boundary and center. Just that the images are stretched in different ways.

of a meridian on the sphere. A cylindrical projection maps every subset of a meridian on the sphere to a vertical straight line on the cylinder. This is because we can always find a plane that intersects the +z axis and the subset of the meridian, and the plane would necessarily intersects a vertical straight line on the cylinder. Finally, recall that perspective projection preserves the linearity of any straight lines in 3D in the 2D perspective views. □

Note that Lemma III.1 applies for cameras with not only arbitrary positions but also arbitrary orientations (rolls, pitches, and yaws), that is, not just ”upright” cameras with a left direction perpendicular to the z axis. See Figure 3 for examples.

In summary, Lemma III.1 states that any vertical features in the 3D scene are guaranteed to be drawn as straight lines in any 2D perspective views. However, the inverse is not necessarily true - *everything* depicted on a column in the panorama would remain straight in 2D perspective views, no matter they are actually vertically aligned in 3D or not.

C. Computational dolly-zoom effect

In cinematography, a ”dolly-zoom” effect refers to the technique that the field-of-view (FOV) angle is continuously narrowed while the camera is moving away from an object in the scene, or vice versa, in a calibrated manner such that the object would appear at the same size on the 2D frame during the movements. The goal is to create so called ”perspective distortions” in which the relative sizes of other objects in the scene change w.r.t. to the size of the particular object. We refer to the excellent introductory video made by Fandor [43] for a visual explanation of dolly-zoom effects. In Computer Graphics and Computer Vision, dolly-zoom techniques were leveraged to perform post-capture image composition [44] and as a guideline to do single-shot view synthesis [45].

We qualitatively observed that, in perspective drawing of panoramas, the amount of barrel distortions is proportional to the distance between the camera position and the origin (becomes zero when they collide). Therefore, our main idea is to leverage dolly-zoom principles to draw roughly the same subset of the panorama in the image plane but using an alternative set of camera position and field-of-view (FOV) angles such that the camera position is as close to the origin as possible. We call the new camera position, P^h, and the new FOV angles, $fovx^h$ and $fovy^h$, the heuristic solution to the computational dolly-zoom problem. To elaborate, P^h is the solution to the following optimization problem:

$$\begin{aligned} \underset{t}{\text{argmin}} \quad & |P^h| \\ \text{subject to} \quad & P^h = P + t \cdot Dir, \end{aligned} \tag{2}$$

where P is the original camera position and Dir is the looking direction. Given the new camera position P^h, we solve the new FOV angles, $fovx^h$ and $fovy^h$, as follows. First, we denote the ”left-middle” and ”right-middle” viewing rays, Ray_{left} and Ray_{right}, as the rays from P toward the left-middle and right-middle points of the original image plane. Next, we find the intersections of Ray_{left} and Ray_{right} to the sphere or cylinder (depending on which projection scheme is used), denoted as I_{left} and I_{right}, respectively. See Figure 4 for an example. Our goal is that in the new viewing pyramid, the new left-middle and right-middle viewing rays should intersect the sphere/cylinder at the same positions. Therefore, the new left-middle viewing ray, Ray^{new}_{left}, is the ray from P^h toward I_{left}. The new right-middle viewing ray, Ray^{new}_{right}, is the ray from P^h toward I_{right}. Finally, we calculate the new ”left-horizontal” and ”right-horizontal” FOV angles, $fovx_{left}$ and $fovx_{right}$, as the angles between Dir and Ray^{new}_{left} and

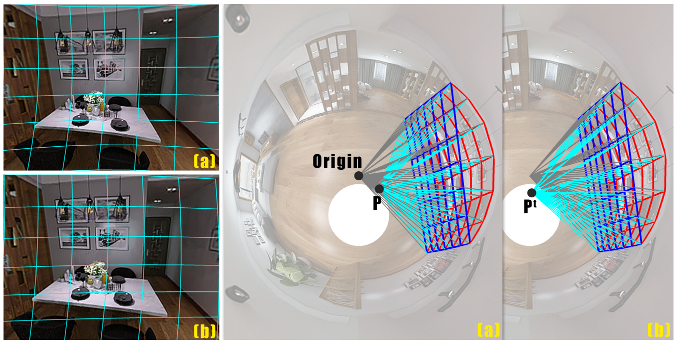

Fig. 5. The viewing pyramids of perspective views of a original camera (a) and the dolly-zoom adjusted version (b). To assist qualitative comparisons, we show how a regular grid on the camera's image plane (blue) are first projected to the cylinder (red) and then projected back to the respective image planes (cyan). We can see that the new perspective view exhibited lower distortions of the image and the projected regular grid.

between Dir and Ray_{left}^{new}, respectively. The new vertical FOV angle is calculated as:

$$atan(\frac{(tan(fovx_{left}) + tan(fovx_{right}))}{aspect}),$$

$aspect$ is the aspect ratio of the original image plane. Note that the new viewing pyramid could be skewed as $fovx_{left}$ and $fovx_{right}$ are not necessarily the same. A visual comparison of two different viewing pyramids that covers nearly the same viewing plane is shown in Figure 5.

IV. Results and comparisons

We tested on a laptop computer with 6-core 2.6GHZ CPU, 16GB ram, and NVidia GTX 1650 Ti graphics card. We use Google Ceres-Solver to solve the computational dolly-zoom effect optimization problem, with which the problems are solved within a few milliseconds. We tested on panoramas from the Structure3D dataset [7] (synthetic), the Gibson dataset [21] (real-world), and several panoramas shoot by ourselves (we used a RICOH THETA Z1 360° camera). The depth information are either ground truth (available in the Structure3D dataset only) or predicted by a neural network-based depth prediction method (BiFuse [10]). The layouts are either ground truth (available in the Structure3D dataset only) or predicted by LED2-Net [6].

A. Qualitative Evaluations

In Figure 8, we qualitatively compare off-center perspective projection results of our method with the vanilla E2P projection and two common approaches to augment panoramas with 3D information (namely per-pixel depths and room layouts). To assist visual comparison, we highlight straight features in 3D in novel off-center perspective views. To do so, we first run HAWP on the original perspective views in cubemap format rendered at the original camera position. Note that straight features in 3D shall be drawn as straight 2D lines in the perspective views so they are easily detectable by HAWP. The detected straight line segments are then projected back to the equirectangular domain using the inverse of E2P projection and baked into the panorama. These baked straight lines are visible when we draw novel perspective views from off-center camera positions.

In summary, our results have significantly lower linear distortions then the vanilla E2P projections. Comparing to novel view synthesis with 3D models built by augmenting panoramas with per-pixel depths or room layouts, their results often have visible artifacts, such as blurs (happen when viewing "stretched" pixels by depths from sideways) and broken images features (happen when room layouts mismatch 3D objects, such as furniture, in the scene). In contrast, our results have no such artifacts. See the accompanying video for animated versions of the results.

B. Quantitative Evaluations

In Figure 7, we quantitatively compared the same video sequences produced by our method and other off-center perspective projection methods by counting the numbers and the sums of 2D Euclidean lengths of straight line segments detected by HAWP [5]. The higher the number and sum of lengths imply that the projection scheme better preserves linearity of 3D straight features in 2D projections. Example detection results are shown in Figure 6. The lengths are normalized w.r.t. the height of the image. We tested on video sequences produced in four different real-world scenes: office, kitchen, dining room, and living room. In all four scenes, our method gets highest scores on both numbers of detected lines and sums of lengths. The quantitative results agree with our qualitative evaluations that the vanilla off-center perspective method would make straight features in 3D to become curved lines in 2D views, which are more difficult for HAWP to detect. Meanwhile, depth and room layout-based 3D models may deform the whole scene and make the 3D straight features blurred or broken.

V. Conclusion

We proposed two lightweight and easy-to-implement solutions to reduce linearity distortions in novel off-center perspective views of panoramas. The effectiveness of our methods are confirmed by qualitative and quantitative evaluations. In terms of limitations, our method does not create true 3D effects such as parallax and occlusion/disocclusion. Therefore, applications of our method are limited to casual viewing of panoramas, not full 6DoF (3D rotations and 3D translations) viewing with VR headsets. Still, we find that the illusion of a 3D scene remains when the camera movement is small. Our hypothesis is that human brains can still deduct 3D depths by the important image features including the linear lines. However, realism of the rendering does begin to degrade when the movements of the camera become larger. For future work, we would like to leverage readily-available 3D information of a panorama such as depths and room layouts predicted by neural networks. Another goal is to evaluate how much linearity of straight features contributes to human perceptions of a 3D world.

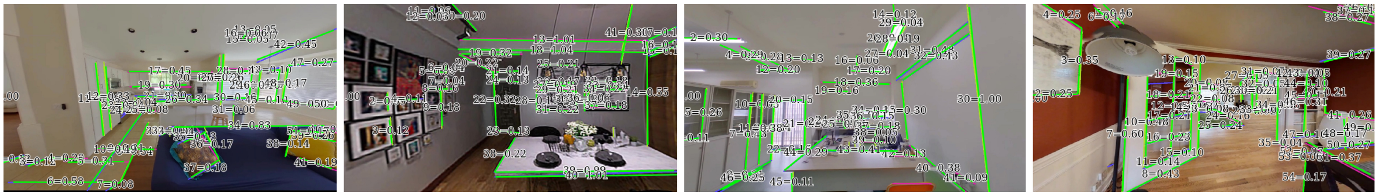

Fig. 6. 2D straight line segments successfully detected by HAWP [5] in novel off-center perspective views by the vanilla E2P projection scheme in four different scenes. HAWP also calculate the length of each detected line as 2D Euclidean distance normalized w.r.t. the height of the image.

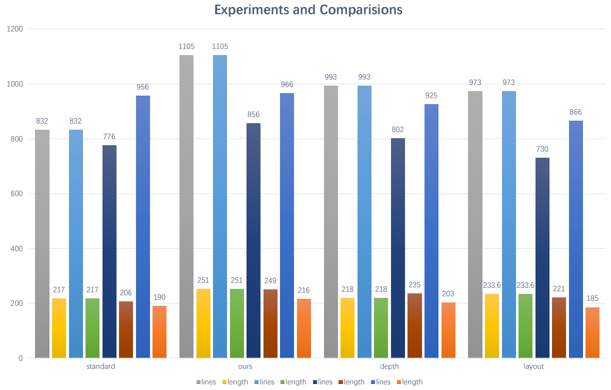

Fig. 7. Quantitatively results for four different Off-center perspective projection scenes. From left to right columns: 1) by vanilla E2P projection, 2) our method, panoramas augmented with 3) depths, and 4) room layouts. For each method, we calculate the numbers of lines and sums of lengths of straight lines detected by HAWP for four scenes. We can see that Our method preserved the most straight lines.

VI. Acknowledgements

This work is funded by the National Science and Technology Council of Taiwan (project number 111R10286C).

References

[1] S. Niklaus, L. Mai, J. Yang, and F. Liu, "3d ken burns effect from a single image," *ACM Trans. Graph.*, vol. 38, no. 6, nov 2019. [Online]. Available: https://doi.org/10.1145/3355089.3356528

[2] M.-L. Shih, S.-Y. Su, J. Kopf, and J.-B. Huang, "3d photography using context-aware layered depth inpainting," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[3] R. Hu, N. Ravi, A. Berg, and D. Pathak, "Worldsheet: Wrapping the world in a 3d sheet for view synthesis from a single image," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2021.

[4] C. Rockwell, D. F. Fouhey, and J. Johnson, "Pixelsynth: Generating a 3d-consistent experience from a single image," in *ICCV*, 2021.

[5] N. Xue, T. Wu, S. Bai, F.-D. Wang, G.-S. Xia, L. Zhang, and P. H. Torr, "Holistically-attracted wireframe parsing," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[6] F.-E. Wang, Y.-H. Yeh, M. Sun, W.-C. Chiu, and Y.-H. Tsai, "Led2-net: Monocular 360deg layout estimation via differentiable depth rendering," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 12 956–12 965.

[7] J. Zheng, J. Zhang, J. Li, R. Tang, S. Gao, and Z. Zhou, "Structured3d: A large photo-realistic dataset for structured 3d modeling," in *Proceedings of The European Conference on Computer Vision (ECCV)*, 2020.

[8] G. Pintore, M. Agus, E. Almansa, J. Schneider, and E. Gobbetti, "SliceNet: deep dense depth estimation from a single indoor panorama using a slice-based representation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 11 536–11 545.

[9] G. Albanis, N. Zioulis, P. Drakoulis, V. Gkitsas, V. Sterzentsenko, F. Alvarez, D. Zarpalas, and P. Daras, "Pano3d: A holistic benchmark and a solid baseline for 360deg depth estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 3727–3737.

[10] F.-E. Wang, Y.-H. Yeh, M. Sun, W.-C. Chiu, and Y.-H. Tsai, "Bifuse: Monocular 360 depth estimation via bi-projection fusion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 462–471.

[11] H. Jiang, Z. Sheng, S. Zhu, Z. Dong, and R. Huang, "Unifuse: Unidirectional fusion for 360° panorama depth estimation," *IEEE Robotics and Automation Letters*, 2021.

[12] Z. Shen, C. Lin, K. Liao, L. Nie, Z. Zheng, and Y. Zhao, "Panoformer: Panorama transformer for indoor 360-degree depth estimation," in *European Conference on Computer Vision*. Springer, 2022, pp. 195–211.

[13] M. Rey-Area, M. Yuan, and C. Richardt, "360MonoDepth: High-resolution 360-degree monocular depth estimation," in *CVPR*, 2022.

[14] C.-H. Peng and J. Zhang, "High-resolution depth estimation for 360deg panoramas through perspective and panoramic depth images registration," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, January 2023, pp. 3116–3125.

[15] C. Zou, A. Colburn, Q. Shan, and D. Hoiem, "Layoutnet: Reconstructing the 3d room layout from a single rgb image," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2051–2059.

[16] S.-T. Yang, F.-E. Wang, C.-H. Peng, P. Wonka, M. Sun, and H.-K. Chu, "Dula-net: A dual-projection network for estimating room layouts from a single rgb panorama," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3363–3372.

[17] C. Sun, C. Hsiao, M. Sun, and H. Chen, "Horizonnet: Learning room layout with 1d representation and pano stretch data augmentation," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, 2019, pp. 1047–1056.

[18] F.-E. Wang, Y.-H. Yeh, M. Sun, W.-C. Chiu, and Y.-H. Tsai, "Led2-net: Monocular 360deg layout estimation via differentiable depth rendering," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 12 956–12 965.

[19] Z. Jiang, Z. Xiang, J. Xu, and M. Zhao, "Lgt-net: Indoor panoramic room layout estimation with geometry-aware transformer network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[20] C. Sun, M. Sun, and H.-T. Chen, "Hohonet: 360 indoor holistic understanding with latent horizontal features," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 2573–2582.

[21] F. Xia, A. R. Zamir, Z. He, A. Sax, J. Malik, and S. Savarese, "Gibson env: Real-world perception for embodied agents," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 9068–9079.

[22] H. Wang, W. Hutchcroft, Y. Li, Z. Wan, I. Boyadzhiev, Y. Tian, and S. B. Kang, "Psmnet: Position-aware stereo merging network for room layout estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 8616–8625.

[23] W. Hutchcroft, Y. Li, I. Boyadzhiev, Z. Wan, H. Wang, and S. B. Kang, "Covispose: Co-visibility pose transformer for wide-baseline relative pose estimation in 360-degree indoor panoramas," in *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXII*. Berlin, Heidelberg: Springer-Verlag, 2022, p. 615–633.

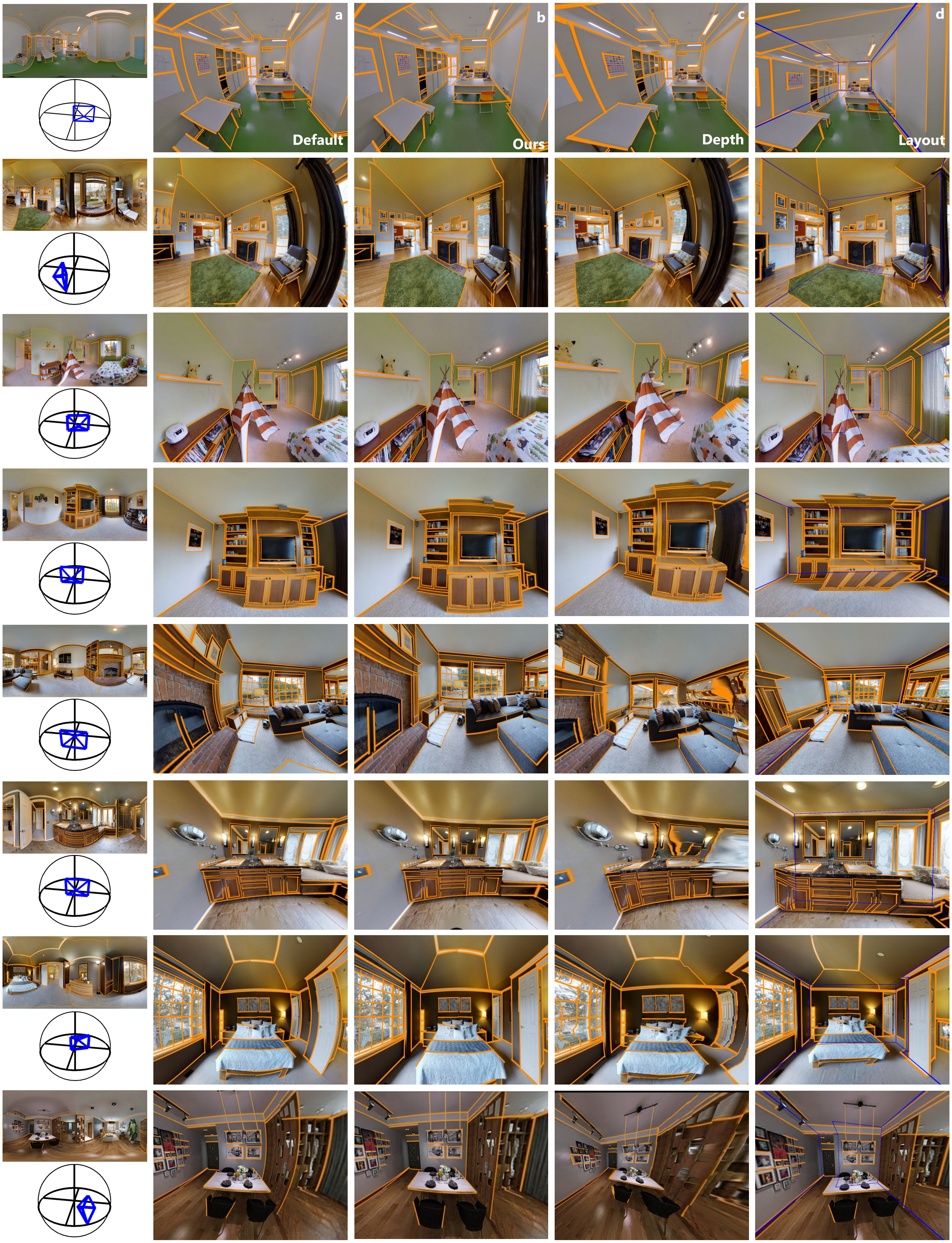

Fig. 8. Off-center perspective projection results. From second left to right columns: 1) by vanilla E2P projection, 2) our method, panoramas augmented with 3) depths and 4) room layouts. The first two rows are based on real-world indoor panoramas shoot by ourselves with predicted depths (BiFuse [10]) and predicted room layouts (LED2-Net [6]). The third row is based on a panorama from the Structure3D dataset (synthetic) with ground truth depths and room layout. The last three rows are based on Gibson dataset (real-world) with predicted depths and room layouts. We show the panoramas annotated with detected straight line segments and visualizations of the off-center cameras in the left column.

[24] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Niessner, M. Savva, S. Song, A. Zeng, and Y. Zhang, "Matterport3d: Learning from rgb-d data in indoor environments," *arXiv preprint arXiv:1709.06158*, 2017.

[25] I. Armeni, S. Sax, A. R. Zamir, and S. Savarese, "Joint 2d-3d-semantic data for indoor scene understanding," *arXiv preprint arXiv:1702.01105*, 2017.

[26] N. Zioulis, A. Karakottas, D. Zarpalas, and P. Daras, "Omnidepth: Dense depth estimation for indoors spherical panoramas," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 448–465.

[27] S. Cruz, W. Hutchcroft, Y. Li, N. Khosravan, I. Boyadzhiev, and S. B. Kang, "Zillow indoor dataset: Annotated floor plans with 360º panoramas and 3d room layouts," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 2133–2143.

[28] J. Shade, S. Gortler, L.-w. He, and R. Szeliski, "Layered depth images," in *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, 1998, pp. 231–242.

[29] T. Zhou, R. Tucker, J. Flynn, G. Fyffe, and N. Snavely, "Stereo magnification: Learning view synthesis using multiplane images," *ACM Trans. Graph.*, vol. 37, no. 4, jul 2018.

[30] K.-E. Lin, Z. Xu, B. Mildenhall, P. P. Srinivasan, Y. Hold-Geoffroy, S. DiVerdi, Q. Sun, K. Sunkavalli, and R. Ramamoorthi, "Deep multi depth panoramas for view synthesis," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIII 16*. Springer, 2020, pp. 328–344.

[31] D. Novotny, B. Graham, and J. Reizenstein, "Perspectivenet: A scene-consistent image generator for new view synthesis in real indoor environments," *Advances in Neural Information Processing Systems*, vol. 32, pp. 7601–7612, 2019.

[32] P. Hedman and J. Kopf, "Instant 3d photography," *ACM Transactions on Graphics (TOG)*, vol. 37, no. 4, pp. 1–12, 2018.

[33] L. Jin, Y. Xu, J. Zheng, J. Zhang, R. Tang, S. Xu, J. Yu, and S. Gao, "Geometric structure based and regularized depth estimation from 360 indoor imagery," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 889–898.

[34] W. Zeng, S. Karaoglu, and T. Gevers, "Joint 3d layout and depth prediction from a single indoor panorama image," in *European Conference on Computer Vision*. Springer, 2020, pp. 666–682.

[35] B. Attal, S. Ling, A. Gokaslan, C. Richardt, and J. Tompkin, "Matryodshka: Real-time 6dof video view synthesis using multi-sphere images," in *European Conference on Computer Vision*. Springer, 2020, pp. 441–459.

[36] M. Broxton, J. Flynn, R. Overbeck, D. Erickson, P. Hedman, M. Duvall, J. Dourgarian, J. Busch, M. Whalen, and P. Debevec, "Immersive light field video with a layered mesh representation," *ACM Transactions on Graphics (TOG)*, vol. 39, no. 4, pp. 86–1, 2020.

[37] T. Bertel, M. Yuan, R. Lindroos, and C. Richardt, "Omniphotos: casual 360° vr photography," *ACM Transactions on Graphics (TOG)*, vol. 39, no. 6, pp. 1–12, 2020.

[38] A. Serrano, I. Kim, Z. Chen, S. DiVerdi, D. Gutierrez, A. Hertzmann, and B. Masia, "Motion parallax for 360 rgbd video," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 5, pp. 1817–1827, 2019.

[39] K.-A. Aliev, A. Sevastopolsky, M. Kolos, D. Ulyanov, and V. Lempitsky, "Neural point-based graphics," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXII 16*. Springer, 2020, pp. 696–712.

[40] V. Sitzmann, J. Thies, F. Heide, M. Nießner, G. Wetzstein, and M. Zollhöfer, "Deepvoxels: Learning persistent 3d feature embeddings," in *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2019.

[41] V. Sitzmann, M. Zollhöfer, and G. Wetzstein, "Scene representation networks: Continuous 3d-structure-aware neural scene representations," *arXiv preprint arXiv:1906.01618*, 2019.

[42] J. Xu, J. Zheng, Y. Xu, R. Tang, and S. Gao, "Layout-guided novel view synthesis from a single indoor panorama," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 16 438–16 447.

[43] Fandor, "This Is How A Dolly Zoom Works," 09 2017. [Online]. Available: https://youtu.be/MeyjyZ6UZII

[44] A. Badki, O. Gallo, J. Kautz, and P. Sen, "Computational zoom: A framework for post-capture image composition," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 46:1–46:14, Jul. 2017.

[45] Y. Liang, R. Ranade, S. Wang, D. Bai, and J. Lee, "The "vertigo effect" on your smartphone: Dolly zoom via single shot view synthesis," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2020, pp. 1407–1415.