Gravity Alignment for Single Panorama Depth Inference

Matheus A. Bergmann Paulo G. L. Pinto Thiago L. T. da Silveira Cláudio R. Jung
Institute of Informatics - Federal University of Rio Grande do Sul

***Abstract*—Monocular depth inference methods based on 360° images allow 3D reconstruction of entire rooms with a single capture. However, most state-of-the-art approaches assume gravity-aligned images and are highly sensitive to camera rotations. Such limitations result in poor depth estimates, which may jeopardize further 3D-based applications. Here, we present a pipeline for spherical single-image depth inference supplied by a novel rotation correction module. We show that our gravity alignment module can improve existing single-image depth estimation methods, being also useful for aligning color and depth to the horizon, which is highly desirable in many applications.**

I. Introduction

Spherical (360°, panoramic or omnidirectional) images are intrinsically defined on the sphere surface and present a full field of view (FoV) coverage of the environment [1], [2]. Recent releases of consumer-grade devices for acquiring and visualizing such images/videos are becoming cheaper and popular, paving the way for many novel applications. In fact, exploring depth from 360° media boosts not only traditional 3D-based applications but also enables fully immersive navigation in augmented, mixed, and virtual reality (AR/MR/VR) [3].

A promising "all-in-one" full 3D scene capturing system is to take a single 360° image and then *infer* its depth map based on the captured panorama. Such a solution enables several applications, like 3D reconstruction/recognition or full six degrees of freedom (DoF) navigation in virtual environments (i.e., allowing the user/virtual camera to freely perform translational and rotational movements). Despite the recent advances on this ill-posed problem when considering *perspective* imagery [4], [5], only few – and more recent – works try to solve it under the omnidirectional optics [6], [7], [8], [9], [10], [11].

Projecting intensities from a spherical image to the plane – which may allow using traditional visual computing methods – induces severe non-affine distortion regardless of the mapping function [12]. The *de facto* planar representation of 360° images is called the equirectangular projection [13], [14]. It maps the spherical image to a rectangular domain and allows using popular Convolutional Neural Networks (CNNs). However, such representation presents a strongly non-uniform sampling (particularly closer to the poles). As such, applying a 3D rotation to an image, although not changing its information, results in very different visual appearances, as shown in Fig. 1.

This study was partially funded by Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001 - and Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq).

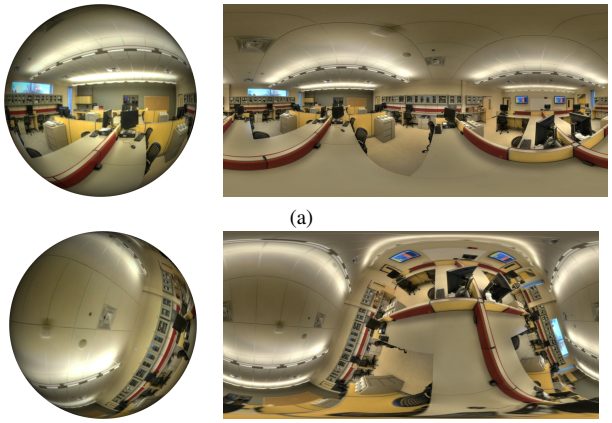


Fig. 1. Two panoramas (on the sphere and in equirectangular format) captured at the same position but different rotations. The image in (a) is roughly aligned to the horizon/ground plane, whereas the image (b) has an arbitrary rotation.

Many existing single-panorama depth estimation methods present a considerable decrease in accuracy when the input images are not gravity-aligned[1]. This happens mainly because most existing datasets for training such methods contain roughly horizon-aligned images, and some authors [6], [15], [16] mention that the input images should be upright corrected – the degradation when pitch and roll rotations are present are explicitly shown in [16]. However, we are not aware of any single-panorama depth estimation approach that actually applies an upright correction mechanism as a pre-processing step. In fact, gravity-alignment approaches for panoramas that work in a variety of scenarios (indoor or outdoor) were introduced recently and popular approaches [17], [18], [19] provide bad estimates when the input panoramas are already roughly aligned. Others approaches [20], [21] report more robust results at the cost of more pre- and post-processing.

This paper presents an approach for "robustifying" existing single-panorama depth estimation methods w.r.t. rotations. We first present a robust gravity-alignment module that produces a roughly homogeneous error distribution regardless of the input rotation of the panorama (hence, correcting the problem of nearly-aligned panoramas of [17], [18], [19]). We then compute the depth map of the gravity-aligned image and perform an inverse rotation to register it with the input

[1]Here, we consider the terms "upright correction", "gravity alignment" or "horizon alignment" as synonyms to refer to the process of aligning the equator of the panorama to the ground plane of the scene (as in Fig. 1(a)).

(possibly rotated) panorama. As we will show in the experimental results, existing depth estimation approaches are indeed sensitive to the capture setup. The proposed strategy improves both quantitative and qualitative depth inference results.

II. Related Work

A. Depth Estimation

Despite the additional problems when processing omnidirectional media, some approaches for single-panorama depth estimation have been proposed in the past few years. To mitigate distortions, Silveira et al. [6] deal with multiple overlapping narrow-FoV tangent projections of the image. The perspective views are individually fed to a *planar* depth estimation method, and the resulting depth maps are further combined by minimizing depth discrepancies and alpha-blending. Yang et al. [7] also support the idea of representing spherical images as a set of tangent planes. By following the Manhattan world assumption, the authors extract and combine geometric and semantic cues from the views for ground plane detection and occlusions reasoning. Finer estimates of typical indoor scene objects, such as furniture, are obtained via depth propagation. The multiplane representation of 360° images was recently formalized by Eder and colleagues [12], who exposed many other potential applications.

Other studies try to work directly in the equirectangular domain, in some cases introducing adaptations to CNNs for dealing with the distortions of panoramas. Zioulis et al. [9] present a fully convolutional encoder-decoder network architecture that uses dilated convolutions, which adapt the receptive fields depending on the kernel position (latitude). Eder and colleagues [8] introduce a plane-aware encoder-decoder network that jointly estimates depth and normal maps from indoor scenarios. Besides the color image, the authors also supply the network with a latitude-longitude geodesic map that, as they claim, helps to account for the irregular sampling of the equirectangular images. Tateno et al. [10] introduce a deformable convolutional filtering approach applicable to monocular depth inference. Their contribution allows cross-domain transfer learning, where perspective image sets can be used to train models applicable to problems based on 360° imagery. Wang and colleagues [11] propose a two-branch architecture that relies not only on the equirectangular projection of the sphere but also its cube-map projection – a particular case of tangent planes. They claim that exploring the benefits of both representations helps to improve the depth estimates. UniFuse [22] builds on top of BiFuse, presenting two encoders fed with equirectangular and cube-map representations and just one decoder. Unlike BiFuse, however, UniFuse unidirectionally feeds the cube-map features to the equirectangular features at the beginning of the decoder, arguing that it improves the final depth estimates in equirectangular format.

The HoHoNet architecture [16] explored the idea of encoding depth information along a column-wise vector. The latent representation is extracted with a feature pyramid as backbone and combined by an attention layer [23], and then unwrapped to the 2D spatial domain using the inverse discrete cosine transform (IDCT). They emphasize the importance of having gravity-aligned inputs, and show that depth estimates degrade as the input panorama deviates from gravity-alignment.

Despite the advances achieved by these approaches, a common drawback is the lack of performance when rotations are introduced in the input panorama, as shown in Fig. 1. In fact, most approaches for monocular depth inference [9], [7], [8], [10] and 3D layout recovery [24], [15] only show results for roughly aligned images with the ground plane.Some works such as [25] evaluate the impact of rotation in the predicted depth values, but within a very limited range (misalignments up to $5°$).

B. Gravity Alignment

Altough gravity alignment could be performed using an Inertial Measurement Unit (IMU) [26], several high-end cameras do not feature this sensor. To overcome this limitation, several approaches for gravity alignment have been proposed in the past, initially for narrow-FoV perspective images and more recently for panoramas. Earlier approaches were based on geometrical cues, such as lines and vanishing points (VP), possibly exploring additional constraints on the input image such as Manhattan [27], [28], [29], [30] or Atlanta [31], [32] worlds. In general, VP-based methods work well in indoor and urban scenes, but tend to fail in natural images due to the lack of structural information. Other approaches [33], [34] try to estimate the horizon line instead, which is orthogonal to the gravity vector. These methods usually explore sky/ground photometric separation and are more suited to natural views (particularly when there is a clear distinction between the sky and the ground), prone to errors in urban and indoor scenarios.

More recently, some methods for upright vector estimation in a generic capture scenario (indoor, outdoor, urban and natural scenes) based on deep learning have been developed. Jeon et al. [18] propose a CNN that predicts the rotation in common perspective images. To predict the upright vector in spherical images, it generates several crops from a panorama and then aggregates all the results. Such an approach has considerably more pre- and post-processing than a direct regression using the full panorama as input. Shan and Li [20] propose two classification models to perform first a coarse alignment (classifying within bins of 10°) and after that a fine alignment (classifying within bins of 1°). Besides using two models, this method cannot fully describe the domain of the problem, generating error even when all the classifications are correct. The Deep360Up model [17] uses a DenseNet backbone with a fully connected layer to directly regress the two relevant angles (pitch and roll) that align the input image to the horizon. This approach is simple and typically produces good results, but the errors grow larger as the input panorama is already roughly aligned. Davidson and colleagues [21] use VPs to help a CNN segment pixels near the vertical axis, which is used to find the vertical axis and to estimate the upright direction. Segmentation models usually have more layers and are slower than direct regression methods because, after the feature extraction, they have several up-scaling layers instead

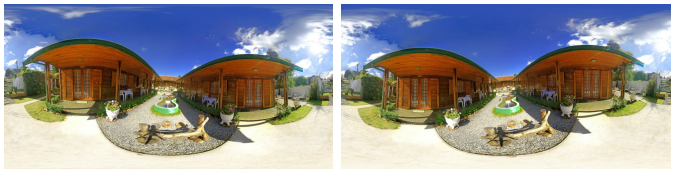


Fig. 2. Example of the "circularity problem". Despite being perceptually similar, the roll angle of two images are numerically very different: 5° and 355°, respectively.

of a fully connected one. Jung et al. [19] use a CNN to extract features and a graph convolutional network (GCN) to find a spherical representation of the input. However, graph-based networks are slower than planar ones, and their method does not address the errors in roughly aligned images.

One crucial issue related to the regression of angular information is the "circularity problem", which might occur since the angles θ and $\theta + 2\pi$ encode the same rotation. Hence, very different angle values in roughly aligned images ($\theta \approx 0$ ou $\theta \approx 2\pi$) induce similar rotations, as illustrated in Fig. 2. Direct regression of the angles, as adopted in [17], [18], [19] is prone to the circularity problem, which can slow down (or compromise) the network optimization. In fact, larger errors were reported for nearly gravity-aligned images in [17]. Note that this poses a relevant issue when supplying a monocular depth estimation method with this rotation awareness module: if the input panorama is already roughly aligned, the baseline depth estimator tends to perform well; on the other hand, a noisy estimate for the gravity vector might misalign the input panorama instead of correcting it, which would decrease the quality of the depth map. Therefore, this paper focuses on developing a gravity alignment approach that yields homogeneous, small errors for all input rotations (i.e., mitigating the circularity problem) and then using it to improve the results of baseline single-panorama depth estimation methods.

III. The Proposed Method

Achieving rotation-invariant depth estimation could be done by training the model with a large range of rotation angles. However, this might slow down or even turn impractical the network optimization because of the high variability of the training set. Here, we propose a two-step approach composed of rotation estimation and upright correction, followed by depth inference (and potential de-rotation). Fig. 3 shows an overview of the proposed method, where the final de-rotation might be omitted depending on the application (such as generating a 3D model aligned with the ground plane).

A. Gravity Alignment

The proposed gravity alignment model is inspired by the Deep360Up architecture [17]. Deep360Up regresses the two relevant rotation angles and corrects the upright vector. It explores a DenseNet backbone [35] (pre-trained on ImageNet [36]), with the classification layer replaced by a regression one (with linear activation). Deep360Up encodes the rotation by the pitch (varying in $[0, 180^\circ)$) and roll (varying

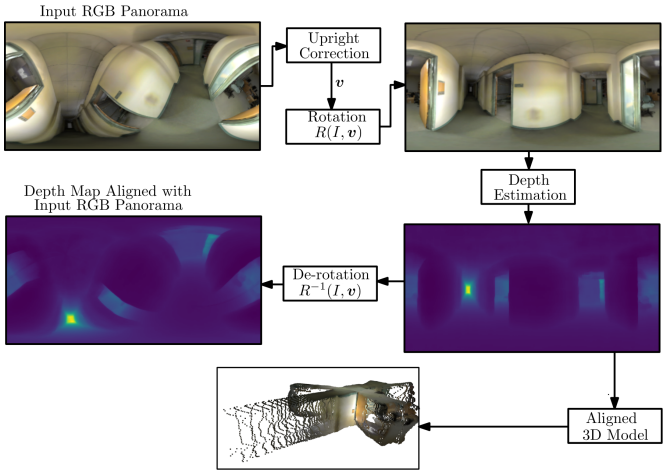


Fig. 3. Overview of our method, where $R(\boldsymbol{I}, \boldsymbol{v})$ denotes the rotation of the spherical panorama $\boldsymbol{I}$ that aligns a unit vector $\boldsymbol{v}$ of $\boldsymbol{I}$ with the vertical axis.

in $[0, 360^\circ)$) angles, and it is trained using high resolution images ($9,104 \times 4,552$) from the SUN360 dataset [37], which are rotated and then downscaled to 442×221 (unfortunately, the full resolution images from the SUN360 dataset are no longer available). Rotating low-resolution equirectangular images produce artifacts due to their non-uniform sampling, which might bias the model as noted in [17].

To generate the poses required to train the proposed model, we synthetically rotate the images of the SUN360 dataset at the maximum available resolution. As far as we know, there are no publicly available datasets with camera viewing angle annotations, and as other works [17], [18], [20], [21], [19], we assume all images in SUN360 are naturally upright-corrected. To generate a balanced dataset in which all possible rotations present approximately the same number of training samples, we generate n upright vectors on the surface of a sphere using a Fibonacci Lattice [38], which yields approximately equiangularly spaced points. Then, each aligned image in the dataset is rotated according to these upright vectors, generating our training set (we used $n = 39,791$, one vector for each image in the train set). According to Jung et al. [17], $10,000$ poses provide enough generalization capabilities, and the performance only increases marginally beyond this value. We use a larger value to generate more variability and compensate for the low-resolution dataset.

In our model, named VectorUp[2], we use the same backbone as [17] for feature extraction, but a different parametrization for the rotation angles: our model outputs the three components of the normalized upright vector. Moreover, due to the unavailability of the full resolution images in SUN360, we use 1024×512 equirectangular images (resized to 442×221 after rotation, the same size used by [17]). By outputting the vector instead of the angles, we fix the circularity problem. Recall that this issue might occur when similar views with very high

[2]Our pre-trained network is available at https://github.com/mabergmann/anglesup

or very low roll angles have different numerical representations (as shown in Fig. 2), which might slow down or compromise the training process.

Formally, our gravity alignment model outputs a 3D vector $\boldsymbol{v} = [x\, y\, z]^\top$, and minimize the squared ℓ_2 error between a normalized version of $\boldsymbol{v}$ and the ground-truth unit vector $\boldsymbol{v}_{gt}$:

$$\mathcal{L}(\boldsymbol{v}) = \left\| \frac{\boldsymbol{v}}{\|\boldsymbol{v}\|} - \boldsymbol{v}_{gt} \right\|^2 = 2\left(1 - \cos\theta\right), \tag{1}$$

where θ is the angle between $\boldsymbol{v}$ and $\boldsymbol{v}_{gt}$.

Also, we noticed that blurring the images alleviates the issues introduced by artificial rotations mentioned by [17], and added Gaussian noise as data augmentation for regularization. These operations are applied at training time with probabilities of 30% and 40%, respectively. The variance of the Gaussian noise is set between 10 and 50 (assuming intensities in the range $[0, 255]$), and the blur is performed with a normalized box filter with a kernel size of 3, 5, or 7 (randomly chosen). Finally, we modified the training protocol to accelerate convergence by using layer-dependent learning rates: 10^{-2} in the randomly initialized layer and 10^{-3} in the pre-trained layers. The use of individualized learning rates aims to focus the training process on the task-specific weights while keeping the feature extraction more stable. To preserve the fine-grained optimization in the late stages of the training process, we used a scheduler that reduces the learning rate by a factor of 0.1 after 10 epochs without improvements in the angular error. These higher learning rates values are feasible only because we replace the unconstrained regression layers with a layer that outputs a normalized vector, reducing numerical instability and vanishing gradients. These changes produced a $4\times$ speed-up improvement (convergence in 200 epochs) in the training stage when compared to [17].

For the sake of comparison, we also trained a model with the exact same architecture as Deep360Up [17], but with the images at available resolution. For training this model, further referred to as AnglesUp for clarity, we reduced the learning rate to 10^{-5} and considered 800 epochs, as in [17] – we noted that convergence using angles was considerably lower than using unit vectors.

B. Depth inference.

Our framework is suited for any single-panorama depth estimation approach. Here, we tested our method using a state-of-the-art monocular method that presents source code and pre-trained weights called OmniDepth [9]. OmniDepth [9] is based on a fully convolutional encoder-decoder that regresses a per-pixel depth map of indoor scenes represented in equirectangular format. For handling the distortions present in these images, the authors employ distortion-aware convolutional filters, which are implemented using dilated convolutions [39] that increase each neuron's receptive field according to its latitude. OmniDepth was originally trained on the 3D60 dataset (also made available in [9]) in a supervised manner with color plus depth upright-aligned 360° images using the ℓ_2 loss and gradient regularization.

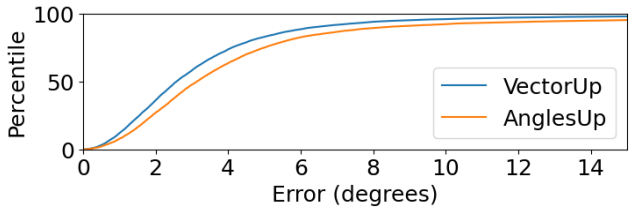


Fig. 4. Percentiles versus the angular error in degrees for VectorUp and AnglesUp.

C. Rectified depth inference and back-rotation.

As shown in Fig. 3, after correcting the orientation of the input image and feeding it to a depth inference method, we de-rotate the resulting depth map, aligning it with the input image. More precisely, let $R(I, \boldsymbol{v})$ denote the rotation of an equirectangular image I that aligns the unit vector point $\boldsymbol{v}$ of I with the vertical axis (i.e., the gravity vector of the rotated image is aligned with $-\boldsymbol{v}$). Also, let $f_u(I)$ denote the upright correction module, which outputs an estimate for $\boldsymbol{v}$, and $f_d(I)$ denote the depth estimation module, which produces a pixel-wise depth estimate for the input image I. The complete pipeline can be summarized as

$$D = R^{-1}\left(f_d\left(R\left(I, f_u(I)\right)\right), f_u(I)\right), \tag{2}$$

where I is the input RGB panorama and D is the output depth map aligned with I. For some applications such as AR, MR, and VR, it might be desirable to generate a 3D model of the captured scene aligned with the horizon. In that case, it is possible to use $R\left(I, f_u(I)\right)$ and $f_d\left(R\left(I, f_u(I)\right)\right)$ in the 3D modeling step, and omit the inverse rotation at the end to obtain a model aligned with the world gravity vector.

IV. Results and Discussion

In this section, we show the results of the upright correction module itself and its integration with monocular panorama depth inference methods.

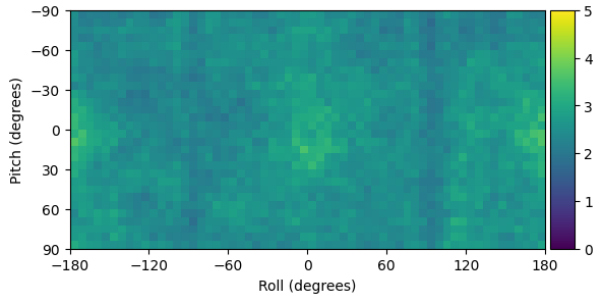
A. Upright correction

For testing the proposed upright correction module, VectorUp, we use the test split of the SUN360 [37] dataset, after synthetically rotating the images, as during the training phase. For assessment, we adopt the angular error, given by

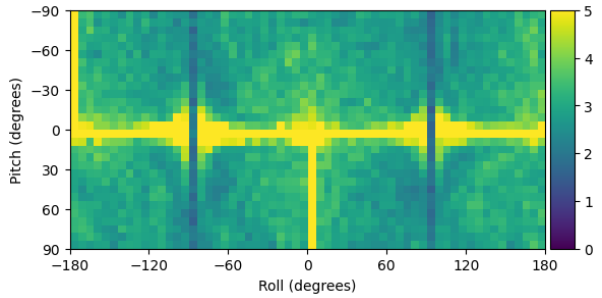
$$e(\boldsymbol{v}_{gt}, \boldsymbol{v}_{out}) = \cos^{-1}\langle \boldsymbol{v}_{gt}, \boldsymbol{v}_{out} \rangle, \tag{3}$$

where $\boldsymbol{v}_{gt}$ and $\boldsymbol{v}_{out}$ represent the ground-truth and estimated upward unit vectors, and $< \cdot, \cdot >$ denotes the inner product.

Fig. 4 shows the percentiles of the error distribution versus the angular error for VectorUp and our implementation of Deep360Up (AnglesUp). The sharp increase of the plot indicates that most of the error distribution presents a small angular error. Note that the curve related to VectorUp is sharper, and present a larger percentile for any angular error, even though AnglesUp was trained for four times more epochs. More precisely, 82.77% of the samples present an angular error smaller than $5°$, which is considered by people as a "very



(a)



(b)

Fig. 5. Median error for each pitch/roll for VectorUp (a) and AnglesUp (b).

satisfactory" orientation, and 97.03% are smaller than $12°$, which is considered a "satisfactory" orientation according to the subjective assessment from [17].

In its original implementation, Deep360Up presents 90.27% and 96.36% of the errors smaller than $5°$ and $12°$, respectively, as reported in [17]. Note that these percentiles are larger than those associated with AnglesUp and larger than those of VectorUp for the "very satisfactory" angular threshold (but smaller in the "satisfactory" range). However, recall that AnglesUp and VectorUp were trained with lower-resolution images, which tend to be affected by prominent artifacts when artificially rotated [17]. Furthermore, we emphasize that our main goal is to mitigate the circularity problem that produces bad estimates for nearly aligned panoramas since they can generate "de-rotated" images that are even less aligned than the original one and compromise the depth map inference instead of improving it.

As shown in Fig. 5a, the angular error of VectorUp is roughly homogeneous for all rotation angles. In fact, it attains a maximum median error of $3.64°$ compared to $5.2°$ reported in [17] (possibly related to nearly aligned images), and $67.14°$ achieved by AnglesUp. The high maximum median error of AnglesUp indicates that using lower-resolution images to train the model strongly impacts the direct angle regression method. In fact, the error distribution of AnglesUp clearly shows a peak for low pitch and roll angles, reproducing the issues reported in [17] originally.

Fig. 6 depicts some results of rectified images in both indoor and outdoor scenarios, along with the corresponding angular errors of the estimated upright vectors using VectorUp and AnglesUp. The images on the top are rotated versions of panoramas from the SUN360 dataset [37] (we assume that original images are gravity-aligned), and the rectified versions (using the regressed vector) for VectorUp and AnglesUp (shown on the middle and bottom, respectively). Note that most of the images are nearly aligned after applying our gravity-alignment process, opposed to AnglesUp. The last image shows a common mistake in complex scenes, in which an "inverted" upright vector is regressed so that the "aligned" image is roughly upside-down.

It is also worth mentioning that we also implemented a modified DenseNet backbone replacing planar convolutions and max-pooling layers with spherical counterparts based on SphereNet [40]. However, this adaptation does not allow using pre-trained weights from ImageNet, and it is considerably slower than the planar versions. When trained from scratch, the resulting model performed worse than the planar one, with a maximum median error of $5.05°$ compared to $3.64°$ with the planar implementation described before (VectorUp). In terms of computational complexity, our model requires 5.65GMac (Multiply and accumulate operations) in the inference phase.

B. Rotation-Aware Depth inference

To evaluate the impact of the alignment procedure on the estimated depth maps, we randomly rotated panoramas (and the depth maps) from the test set of the 3D60 dataset [9], assuming that they are roughly upright aligned. We then compare the depth maps produced by directly applying OmniDepth and when combining it with VectorUp and AnglesUp, as given by Eq. (2). More precisely, we created $10,000$ panoramas with random synthetic rotations (pitch angle between -90° and 90° and roll angle between -180° and 180°). Similarly to [9], we perform median alignment of predicted depth map D_{pred} and the annotated map D_{gt}, scaling the former by a factor s:

$$s = \frac{\text{median}(D_{gt})}{\text{median}(D_{pred})}. \tag{4}$$

We then compute common evaluation metrics, namely the mean absolute relative error (Abs Rel), mean square relative error (Sq Rel), root mean square error (RMS), logarithmic root mean square error (RMSlog), and the accuracy at a given threshold (δ) [6], [9], [7], [8], [10] while ignoring missing values in the D_{gt}.

For the sake of comparison, we also show the result of the proposed pipeline applied directly to the original images of 3D60. In this case, the results with our rotation-aware procedure tend to be worse than the original method since the two rotations (the initial alignment of the RGB panorama and the inverse rotation of the depth map) introduce artifacts. As a final experiment, we tested a hybrid approach in which the panorama is only rotated if the estimated upright vector presents an angle larger than ψ w.r.t. the vertical axis so that panoramas that are roughly aligned (and for which the baseline depth estimator is expected to work well) are not rotated.

The average results, in all three tested scenarios, for OmniDepth without upright correction or using either VectorUp and AnglesUp are shown in Table I. We can observe a

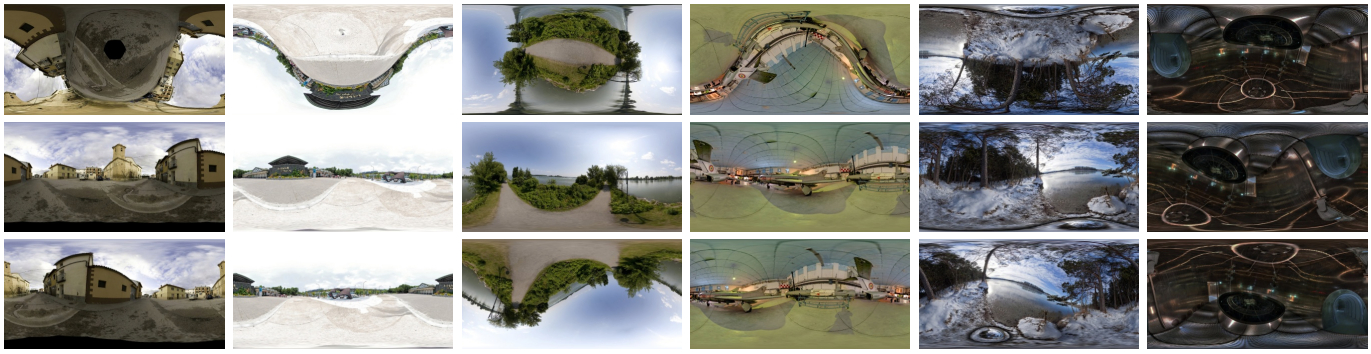


Fig. 6. Example of inputs (top) rectified by VectorUp (middle) and AnglesUp (bottom). Errors for Vector up are, from left to right, 0.8°, 2.18°, 3.26°, 6.53°, 10.7°, 158.74°. Errors for AnglesUp are, from left to right 5.07°, 1.9°, 146.48°, 7.15°, 24.05° and 11.95°.

TABLE I
QUANTITATIVE RESULTS FOR THE BASELINE METHOD UNDER ROTATION SCENARIOS. RESULTS ON THE ORIGINAL DATASET ARE ALSO PROVIDED.

Framework	Orientation	Abs Rel ↓	Sq Rel ↓	RMS ↓	RMSlog ↓	$\delta < 1.25$ ↑	$\delta < 1.25^2$ ↑	$\delta < 1.25^3$ ↑
OmniDepth	rotated	0.1797	0.1236	0.5972	0.2511	0.7126	0.9285	0.9783
OmniDepth + VectorUp	rotated	0.1156	0.0547	0.3735	0.1701	0.8677	0.9741	0.9928
OmniDepth + AnglesUp	rotated	0.1113	0.0534	0.3735	0.1676	0.8753	0.9745	0.9925
OmniDepth	original	0.0641	0.0197	0.2297	0.0993	0.9663	0.9951	0.9984
OmniDepth + VectorUp	original	0.1127	0.0533	0.3656	0.1658	0.8765	0.9763	0.9931
OmniDepth + AnglesUp	original	0.1767	0.1172	0.5660	0.2454	0.7228	0.9319	0.9799
OmniDepth + VectorUp ($\psi = 12°$)	original	0.1029	0.0479	0.3449	0.1585	0.8874	0.9775	0.9936
OmniDepth + AnglesUp ($\psi = 12°$)	original	0.1762	0.1169	0.5653	0.2452	0.7233	0.9319	0.9799
OmniDepth + VectorUp ($\psi = 14°$)	original	0.1015	0.0472	0.3419	0.1572	0.8898	0.9777	0.9936
OmniDepth + AnglesUp ($\psi = 14°$)	original	0.1761	0.1168	0.5648	0.2451	0.7234	0.9319	0.9799

considerable improvement in all error metrics when correcting the image orientation for the rotated experiment using either the proposed method VectorUp or AnglesUp compared to the baseline OmniDepth. We can also note that applying the rotation-aware procedures to already aligned images (middle section of the table) produces worse results than the baseline due to rotation-induced artifacts, as mentioned before, but are consistent with all the error metrics in the first experiment. The use of AnglesUp attained considerably worse results than our method since it tends to produce particularly larger errors for roughly aligned panoramas, as shown in Fig. 5b.

In the last section of Table I (with ψ values indicated), we see the depth estimation results using the rotation-aware scheme only if the upright vector presents an angle larger than ψ w.r.t. the vertical axis. We tested two values for ψ. The first one, $\psi = 12°$, is based on the "satisfactory" orientation according to [17]. The second value for ψ is set experimentally considering the depth error produced by OmniDepth and variants as a function of the rotation angle. Fig. 7 shows at what point (marked in red, representing the second choice for $\psi = 14°$) OmniDepth start producing worst results in rotated images than the rotation-aware approaches. A total of 189 and 199 (out of 325) images were considered aligned using VectorUp with $\psi = 12°$ and $\psi = 14°$, respectively, and all the error metrics improved. For AnglesUp, only 6 and 7 images were considered aligned according the same angular thresholds, and only a marginal improvement w.r.t. the second experiment was achieved.

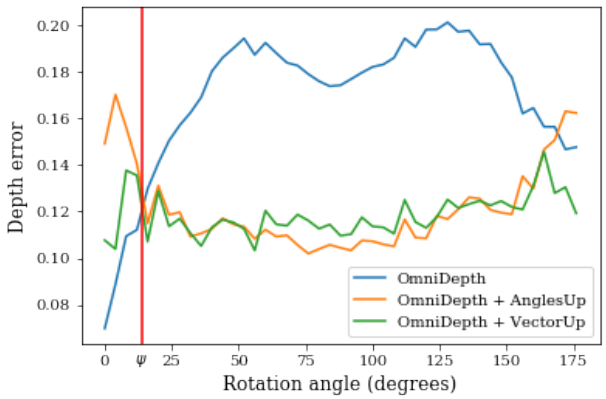


Fig. 7. Mean Absolute Relative error in depth for OmniDepth (blue) and its rotation-aware versions using VectorUp (green) and AnglesUp (orange) as a function of the angular deviation from the gravity vector. The red vertical line at $\psi = 14°$ indicates the angular threshold where rotation-aware versions improve over the baseline.

To better evaluate the effect of using the proposed rectification scheme, Fig. 8 shows the depth maps produced by the two frameworks and the ground-truth depth map for some images that are far from being gravity-aligned. We can observe that the direct application of OmniDepth to rotated images generates artifacts, whereas the proposed method presents results similar to the ground-truth.

We also evaluated the depth inference error as a function of the rotation of the input panorama in Fig. 9, where the

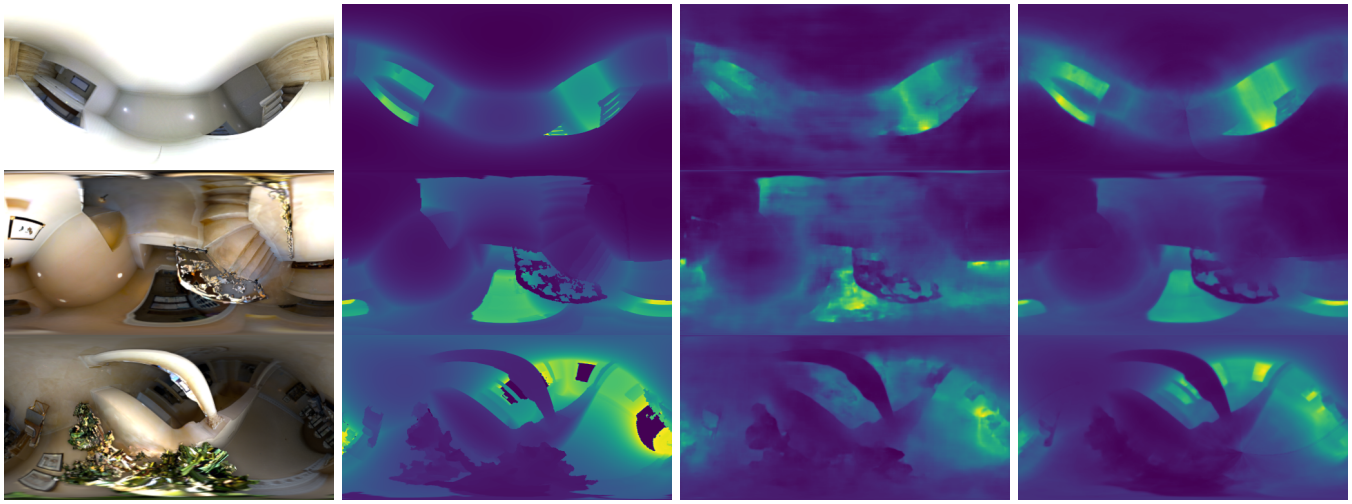


Fig. 8. Depth inference results. First column shows color images. Second to fourth columns present the ground-truth depth maps and the estimates from the baseline OmniDepth and OmniDepth when supplied with VectorUp.

horizontal and vertical axes represent the roll and pitch angles, respectively. The error maps show that OmniDepth, when supplied by our upright correction module, has much more stable estimates regardless of the rotations. Although not very clear, the error map associated with AnglesUp, which resembles the error map shown in Fig. 5b with a much more subtle peak, presents larger values for low rotations (particularly for low pitch values). Fig. 7 also provides this information.

Finally, for the sake of illustration, we also show in Fig. 10 the depth maps produced by the recent depth estimator BiFuse [11] in a rotated panorama and its aligned version by our method. We observe that the prediction of BiFuse with upright adjustment seems sharper and has more valid predictions on areas corresponding to the horizon on the original image, where most of the information is located. A full quantitative analysis of BiFuse cannot be performed because the weights provided by the authors do not work near the 360° images poles, compromising the analysis.

V. Conclusions

This paper presented a pipeline for inferring a dense depth map from a single panorama under a wide range of possible rotations. For that purpose, we proposed an upright correction module that rotates the input image to a canonical (rectified) view. Then we apply a deep monocular depth inference method and align the generated depth map with the original image by de-rotating the depth map.

Our experimental validation shows that the upright rectification improves qualitative and quantitative results of a state-of-the-art baseline depth estimation method [9], which is sensitive to rotations. We also show promising visual results for another depth estimator [11].

As future work, we plan to train the pipeline in an end-to-end manner instead of training two isolated modules aiming to further improve the estimated depth maps. We also intend to evaluate our gravity alignment approach for single-panorama layout estimation. Since several of these methods assume Manhattan worlds, in which the room walls are aligned with a

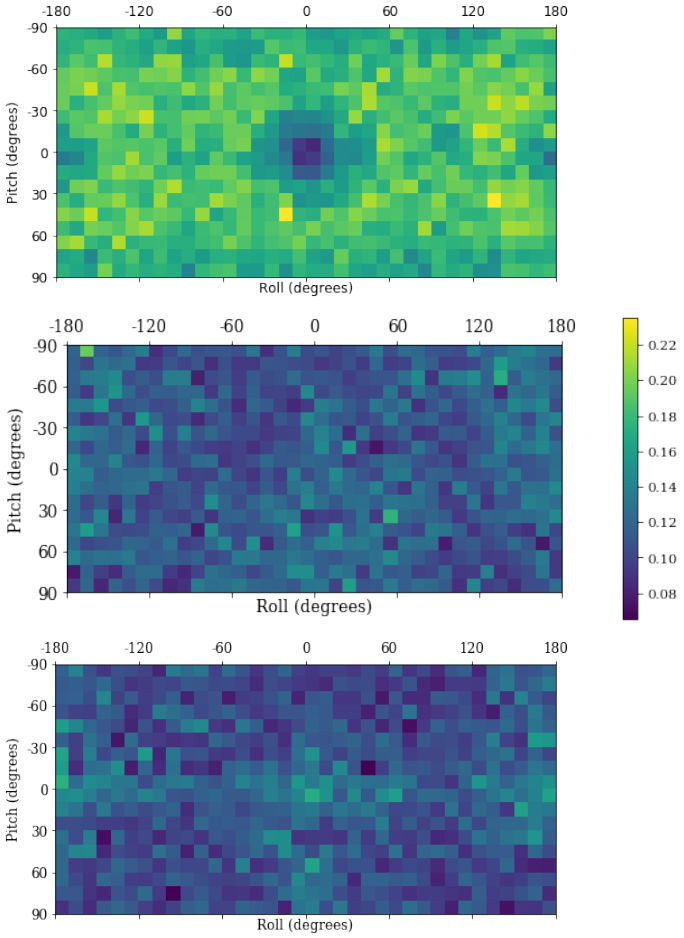


Fig. 9. Relative depth error varying the rotation angles (roll and pitch) without rotation correction (top), with VectorUp (middle) and with AnglesUp (bottom)

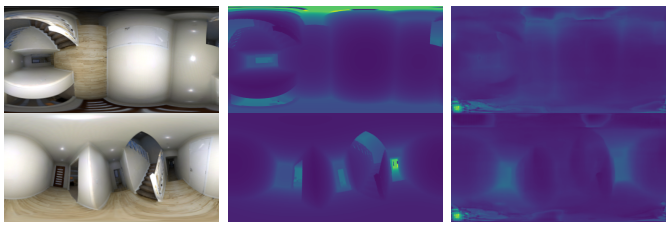


Fig. 10. Depth inference results. In the first row: rotated color and depth maps (ground-truth and results from BiFuse). In the second row: upright corrected color and depth maps (ground-truth and results from BiFuse).

canonical coordinate system [41], gravity-alignment becomes a critical issue.

REFERENCES

[1] J. Fujiki, A. Torii, and S. Akaho, "Epipolar Geometry Via Rectification of Spherical Images," in *MIRAGE*, 2007, pp. 461–471.

[2] T. Akihiko, I. Atsushi, and N. Ohnishi, "Two-and three-view geometry for spherical cameras," *Workshop on Omnidirectional Vision, Camera Networks and Non-classical Cameras*, vol. 105, pp. 29–34, 2005.

[3] J. Huang, Z. Chen, D. Ceylan, and H. Jin, "6-DOF VR videos with a single 360-camera," in *IEEE VR*, 2017, pp. 37–44.

[4] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab, "Deeper depth prediction with fully convolutional residual networks," in *3DV*, 2016.

[5] F. Khan, S. Salahuddin, and H. Javidnia, "Deep learning-based monocular depth estimation methods—a state-of-the-art review," *Sensors*, vol. 20, no. 8, pp. 2272, 2020.

[6] T. L. T. da Silveira, L. P. Dal'aqua, and C. R. Jung, "Indoor Depth Estimation from Single Spherical Images," in *IEEE ICIP*, 2018, pp. 2935–2939.

[7] Y. Yang, R. Liu, and S. B. Kang, "Automatic 3D Indoor Scene Modeling from Single Panorama," in *IEEE/CVF CVPR*, 2018, p. 5430.

[8] M. Eder, R. Moulon, and L. Guan, "Pano Popups: Indoor 3D Reconstruction with a Plane-Aware Network," in *3DV*. 2019, pp. 76–84, IEEE.

[9] N. Zioulis, A. Karakottas, D. Zarpalas, and P. Daras, "OmniDepth: Dense Depth Estimation for Indoors Spherical Panoramas," in *ECCV*, 2018, pp. 453–471.

[10] K. Tateno, N. Navab, and F. Tombari, "Distortion-Aware Convolutional Filters for Dense Prediction in Panoramic Images," in *ECCV*, 2018, pp. 732–750.

[11] F. Wang, Y. Yeh, M. Sun, W. Chiu, and Y. Tsai, "Bifuse: Monocular 360 depth estimation via bi-projection fusion," in *IEEE/CVF CVPR*, 2020, pp. 459–468.

[12] M. Eder, M. Shvets, J. Lim, and J. Frahm, "Tangent Images for Mitigating Spherical Distortion," in *IEEE/CVF CVPR*, 2019, pp. 12426–12434.

[13] B. Coors, A. P. Condurache, and A. Geiger, "SphereNet: Learning spherical representations for detection and classification in omnidirectional images," in *ECCV*, 2018, pp. 525–541.

[14] T. L. T. da Silveira and C. R. Jung, "Dense 3D Scene Reconstruction from Multiple Spherical Images for 3-DoF+ VR Applications," in *IEEE VR*, 2019, pp. 9–18.

[15] C. Sun, C. Hsiao, M. Sun, and H. Chen, "HorizonNet: Learning Room Layout with 1D Representation and Pano Stretch Data Augmentation," in *IEEE/CVF CVPR*, 2019, pp. 1047–1056.

[16] Cheng Sun, Min Sun, and Hwann-Tzong Chen, "Hohonet: 360 indoor holistic understanding with latent horizontal features," in *IEEE/CVF CVPR*, 2021, pp. 2573–2582.

[17] R. Jung, A. S. J. Lee, A. Ashtari, and J. Bazin, "Deep360Up: A Deep Learning-Based Approach for Automatic VR Image Upright Adjustment," in *IEEE VR*, 2019, pp. 1–8.

[18] Junho Jeon, Jinwoong Jung, and Seungyong Lee, "Deep upright adjustment of 360 panoramas using multiple roll estimations," in *ACCV*. Springer, 2018, pp. 199–214.

[19] Raehyuk Jung, Sungmin Cho, and Junseok Kwon, "Upright adjustment with graph convolutional networks," in *IEEE ICIP*. IEEE, 2020, pp. 1058–1062.

[20] Yuhao Shan and Shigang Li, "Discrete spherical image representation for cnn-based inclination estimation," *IEEE Access*, vol. 8, pp. 2008–2022, 2019.

[21] Benjamin Davidson, Mohsan S. Alvi, and João F. Henriques, "360°camera alignment via segmentation," in *Computer Vision – ECCV 2020*, Cham, 2020, pp. 579–595, Springer International Publishing.

[22] Hualie Jiang, Zhe Sheng, Siyu Zhu, Zilong Dong, and Rui Huang, "Unifuse: Unidirectional fusion for 360° panorama depth estimation," *IEEE Robotics and Automation Letters*, 2021.

[23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *NIPS*, vol. 30, pp. 5998–6008, 2017.

[24] C. Fernandez-Labrador, A. Perez-Yus, G. Lopez-Nicolas, and J. J. Guerrero, "Layouts from panoramic images with geometry and deep learning," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3153–3160, 2018.

[25] Giovanni Pintore, Marco Agus, Eva Almansa, Jens Schneider, and Enrico Gobbetti, "Slicenet: Deep dense depth estimation from a single indoor panorama using a slice-based representation," in *IEEE/CVF CVPR*, 2021, pp. 11536–11545.

[26] Pengfei Zhou, Mo Li, and Guobin Shen, "Use it free: Instantly knowing your phone attitude," in *MobiCom*, 2014, pp. 605–616.

[27] Jean-Charles Bazin, Cédric Demonceaux, Pascal Vasseur, and Inso Kweon, "Rotation estimation and vanishing point extraction by omnidirectional vision in urban environment," *The International Journal of Robotics Research*, vol. 31, no. 1, pp. 63–81, 2012.

[28] Jean-Charles Bazin and Marc Pollefeys, "3-line ransac for orthogonal vanishing point detection," in *IEEE/RSJ IROS*. IEEE, 2012, pp. 4282–4287.

[29] Jean-Charles Bazin, Yongduek Seo, and Marc Pollefeys, "Globally optimal consensus set maximization through rotation search," in *ACCV*. Springer, 2012, pp. 539–551.

[30] Lilian Zhang, Huimin Lu, Xiaoping Hu, and Reinhard Koch, "Vanishing point estimation and line classification in a manhattan world with a unifying camera model," *International Journal of Computer Vision*, vol. 117, no. 2, pp. 111–130, 2016.

[31] Kyungdon Joo, Tae-Hyun Oh, In So Kweon, and Jean-Charles Bazin, "Globally optimal inlier set maximization for atlanta frame estimation," in *IEEE/CVF CVPR*, 2018, pp. 5726–5734.

[32] Jinwoong Jung, Beomseok Kim, Joon-Young Lee, Byungmoon Kim, and Seungyong Lee, "Robust upright adjustment of 360 spherical panoramas," *The Visual Computer*, vol. 33, no. 6, pp. 737–747, 2017.

[33] Cédric Demonceaux, Pascal Vasseur, and Claude Pégard, "Omnidirectional vision on UAV for attitude computation," in *ICRA*. IEEE, 2006, pp. 2842–2847.

[34] Cédric Demonceaux, Pascal Vasseur, and Claude Pégard, "Robust attitude estimation with catadioptric vision," in *IEEE/RSJ IROS*. IEEE, 2006, pp. 3448–3453.

[35] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *IEEE/CVF CVPR*, 2017, pp. 4700–4708.

[36] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *IEEE CVPR*, 2009, pp. 248–255.

[37] J. Xiao, K. A Ehinger, A. Oliva, and A. Torralba, "Recognizing scene viewpoint using panoramic place representation," in *IEEE CVPR*, 2012, pp. 2695–2702.

[38] Benjamin Keinert, Matthias Innmann, Michael Sänger, and Marc Stamminger, "Spherical fibonacci mapping," *ACM Transactions on Graphics*, vol. 34, no. 6, pp. 1–7, 2015.

[39] F. Yu, V. Koltun, and T. Funkhouser, "Dilated residual networks," in *IEEE/CVF CVPR*, 2017, pp. 636–644.

[40] Benjamin Coors, Alexandru Paul Condurache, and Andreas Geiger, "Spherenet: Learning spherical representations for detection and classification in omnidirectional images," in *ECCV*, 2018, pp. 518–533.

[41] C. Zou, J.-W. Su, C.-H. Peng, A. Colburn, Q. Shan, P. Wonka, H.-K. Chu, and D. Hoiem, "Manhattan room layout reconstruction from a single 360° image: A comparative study of state-of-the-art methods," *International Journal of Computer Vision*, pp. 1–22, 2021.