Monocular Depth Estimation Using Geometric Perception Fusion

Chunhua Wang[1], Ying Gao[1], and Zhenyu Wang[2]
(Corresponding author: Chunhua Wang)
Suzhou Industrial Park Institute of Service Outsourcing[1]
Suzhou 215123, Jiangsu, China
Qilu University of Technology, Shandong, China[2]
wangch@siso.edu.cn
(Received Apr. 23, 2024; Revised and Accepted Mar. 24, 2025; First Online Apr. 17, 2025)

Abstract

We face a big challenge in applying deep-learning methods to spherical distortion in intensive regression tasks that require structural detail (such as depth estimation). Using the vanilla CNN layer on a distorted 360 image results in undesired information loss. In this paper, we propose ContextFusion, a 360 monocular depth estimation pipeline, to solve the spherical distortion problem. Our pipeline converts 360 images into perspective blocks with less distortion (i.e., tangential images) to obtain piece-by-piece predictions via CNN and then merges the piece-by-piece results for the final output. To address the discrepancy between patch predictions, which is a major issue affecting merge quality, we propose a new framework with the following key components. First, we propose a mechanism of geometric perception feature fusion of D geometric features and 2D image features to compensate for piece-by-piece differences. Second, we use a self-attention-based converter architecture for the global aggregation of patch information, which further improves consistency. Finally, we introduce an iterative depth refinement mechanism to further refine the estimated depth based on more accurate geometric features. Experiments show that our approach greatly alleviates the distortion problem and achieves state-of-the-art performance on several 360 monocular depth estimation benchmark datasets.

Keywords: CNN; Context Fusion; Deep-learning; Geometric Perception Fusion; Spherical Distortion

1 Introduction

360-degree images provide a comprehensive view of a scene with a wide field of view (FoV), which helps to fully understand the scene. However, commonly used 360 image representation formats, such as isometric projection (ERP) images, can introduce geometric distortion. The distortion factor varies vertically and may degrade the performance of regular convolution layers designed for non-distorted perspective images.Many studies have been proposed to address the distortion problem. [4] proposes distortion-sensing convolution or spherical custom cores. However, it is unclear how effective this spherical convolution is, especially at deeper layers [28]. Some spherical cellular neural networks [29] define convolution in the spectral domain, which may bring greater computational overhead. There have also been attempts to solve the ERP distortion problem with other formats with less distortion.

BiFuse [28] and UniFuse [14] have complementary and appropriate connections from ERP and cube map. Some work [3, 8, 22, 27] has repeatedly applied regular CNNs to multiple perspective projections of 360 images. More recently Eder *et al.* [9] proposed using a subdivided set of icosahedral tangential images and demonstrated that using tangential image representation could facilitate network transmission between perspective images and 360 images.

The use of tangential images [9] is advantageous because it has less distortion and makes good use of the large number of pretrained cellular neural networks developed for fluoroscopic imaging. In addition, tangential image representation inherits superior scalability to handle high-resolution inputs compared to the holistic approach. However, the vanilla pipe [9] has some limitations. First, there are serious differences between perspectives because the same object can look different from multiple views. This problem is particularly problematic for deep regression tasks because inconsistent depth scales estimated from a single tangential image can produce unwanted artifacts during merging. Second, unfortunately, the advantage of estimating depth from the overall image is lost because the global scene is broken down into local tangential images. The predictions from the tangential images are independent of each other, and no information is exchanged between the tangential images.

In this paper, we present ContextFusion, a monocu-

lar depth estimation framework with geometry-aware fusion. We proposed the following three key components to solve the aforementioned discrepancy issue and merge the depth results of tangent images seamlessly. First, we use a geometric embedding module to provide additional features to compensate for the discrepancy between 2D features from patch to patch. For each patch, we calculate the 3D points located on the spherical surface that correspond to the patch pixels, encode them and the patch center coordinate through shared Multi-layer Perceptron (MLP), and add the geometric features to the corresponding 2D features. Second, to regain the holistic power in understanding the entire scene, we incorporate a self- attention -based transformer in our pipeline. With the trans- former, patch-wise information is globally aggregated to enhance the estimation of the global scale of depth, and to improve the consistency between patch-wise results. Third, we introduce an iterative refining mechanism, where more accurate 3D information from the predicted depth maps is fed back to the geometric embedding module to further im- prove the depth quality in an iterative manner.

We tested ContextFusion on three benchmark datasets: Stanford2D3D [1], Matterport3D [3], and D [31]. The experimental results show that our method is significantly superior to the prior art methods on all of these data sets. Our contribution can be summarized as follows:

- We propose a monocular depth prediction pipe that solves distortion problems with geometric sensing fusion and achieves state-of-the-art performance.
- We introduced a geometric embedding network to provide 3D geometric features to mitigate differences in block-by-block image features.
- We incorporate a self-focused converter to globally aggregate piece-by-piece information, which enhances the estimation of physical depth scales.
- We propose an iterative mechanism to further improve depth estimation of structural details.

2 Related Work

2.1 Monocular Depth Estimation

Monocular depth estimation, which uses a single RGB image as input to predict per-pixel depth values, has been studied extensively due to its wide range of applications. Earlier work focused on network architecture and hypervision [13]. Recently, researchers have been investigating the use of unsupervised learning on stereo pairs [11] or monocular video streams [12] to extend training data to unlabelled image sequences for wider applications. However, this method is still sensitive to many factors (for example, inherent changes in the camera) and is difficult to generalize to new scenes. To improve robustness and scalability, some methods utilize additional sensor inputs, such as lidar and RGBD cameras [19]. However, there are many real-world scenarios where additional computing or power consumption is undesirable.

2.2 Depth Estimation

Monocular depth estimation from images has been investigated from a variety of perspectives. Zioulis *et al.* [30] explored the spherical stereo geometry and estimated depth from monocular ERP input via view synthesis. PanoDepth [18] leveraged stereo constraints to improve monocular depth performance. Eder et al [7].and Zeng *et al.* explored joint learning from different modalities (e.g. layout, normal, semantics, etc.). HoHoNet proposed to utilize latent horizontal feature representation to encode ERP image features. To handle the irregular distortion of ERP images, several distortion-aware convolutions have been proposed. For example, Fernandez *et al.* [10] introduced EquiConv which applied deformable convolution to accommodate spherical geometry. Tateno *et al.* [25] proposed to apply regular CNN to perspective images during training, and distortion-aware convolution during testing. Instead of directly tackling the distortion of ERP, several approaches proposed to use other representations with less distortion, such as cubemap [27], fusion between ERP and cubemap [16], and multiple perspective projections of images [22]. A recent work by Eder *et al.* [9] proposed to use tangent images, a set of oriented, low-distortion images rendered tangent to faces of the icosahedron, to represent a image. It is advantageous to use tangent images since it has less distortion and can effectively leverage pretrained CNN models developed for perspective imaging. However, discrepancies be- tween tangent images are not addressed in [9], which leads to a downgrade of the final merged result. In this work, we follows the paradigm proposed in [9] of using tangent images, but simplified and adapted it for depth estimation. In addition, we successfully address the discrepancy issue by incorporating geometry-aware fusion and the transformer.

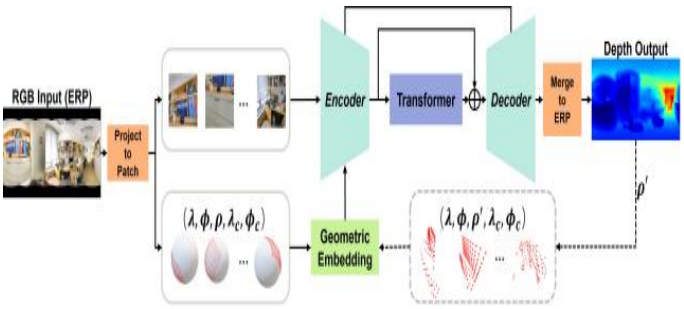

Figure 1: Overview of our proposed ContextFusion.

Our approach takes a monocular ERP RGB as input, projects it onto multiple patches at multiple viewpoints, and processes each distortion-free patch using an encoddecoder network to generate a patch-by-patch depth map (top stream). The piece-by-piece output is incorporated

into the final ERP depth chart. At the same time, corresponding points located on the sphere are sampled and geometric features (underflows) are generated by geometric embedding networks. Geometric features are incorporated into the image encoder to compensate for block by block differences and improve the quality of the merged results. For each sampling point, we use its spherical coordinate (λ, σ, γ) and tangent plane central coordinate (λ_c, σ_c) as input attributes of the geometric embedding network, which provides the necessary information to align 2D features. The converter architecture is integrated for global aggregation of deep chip by chip features, which further improves the consistency of chip by chip output. In addition, we introduced an iterative refinement mechanism (shown as a dashed line) to further improve deep recovery. In particular, the γ value is updated based on the depth estimated in the last iteration.

2.3 Transformer

Transformer architecture was first proposed in natural language processing [26] and has since been widely used for computer vision tasks such as image classification [6], depth estimation [32], object detection [2] and semantic segmentation [21]. The visual converter has a natural fit with monocular depth estimation because the self-attention module can explicitly utilize long-distance text. However, when the converter is applied to an image, distortion reduces the power of the converter to exploit the pairwise correlation between patches. In this work, we provide distortion-free and geometrically aware inputs to the converter so that the converter can focus on global aggregation of piece-by-piece information.

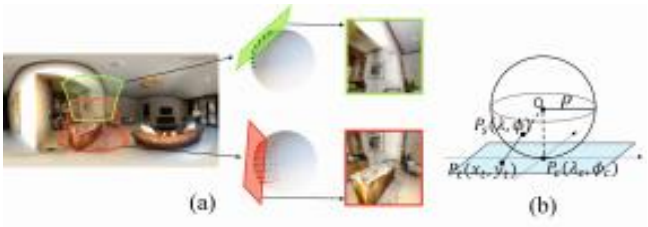

Figure 2: Example of a tangential image projection.

3 Method

Figure 1 shows an overview of the complete flow of the proposed ContextFusion framework. First, the ERP input images are converted into a set of tangential images using a gnomonic projection. The projected distortion-free tangential image is then passed through the encoder-decoder network to produce piece-by-slice depth estimates, which are then fused into the ERP depth output.

Two tan images are projected from two different perspectives. In both ERP and tangential surfaces, the corresponding areas are highlighted in the same color. As shown in the figure 2, there are usually overlapping areas between two adjacent sheets, and the same object may appear in different cases in different sheets. (b) heliospheric projection. The point Ps (λ, σ_c) on the sphere is projected to the point Pt (x_t, y_t) on the plane, which is tangent to Pc (λ_c, σ_c).

To mitigate block-by-block differences, we introduce a new geometric embedding module that encodes the spherical coordinates associated with each tangential image pixel, providing additional geometric features to facilitate the integration of block-by-block image features. In order to further improve the consistency between piece-by-slice predictions and to better estimate the global depth scale, features from the deepest layer of the encoder are globally aggregated by a self-attention-based converter. Finally, the iterative refinement mechanism is used to further improve the depth quality. We iteratively update the spherical coordinates based on more accurate estimates obtained from the previous iteration. We train our network in an end-to-end manner, and the only oversight is the ultimate merge depth compared to the ground reality.

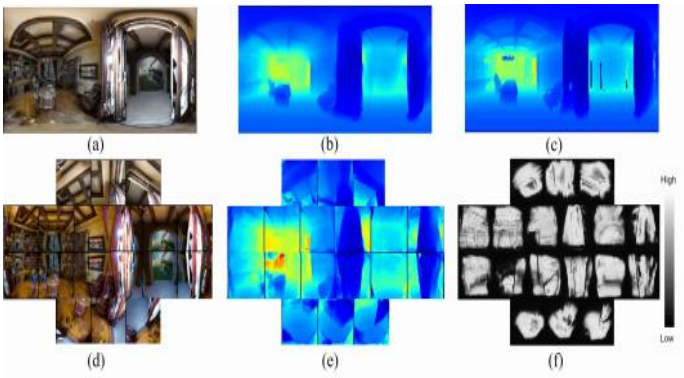

Figure 3: Line 1: (a) An example of an ERP RGB input image, (b) a final merged predictive ERP depth map, and (c) ground truth ERP depth. Line 2: (d) RGB tangential image blocks generated from (a), (e) depth maps estimated piece-by-piece, and (f) piece-by-piece estimated confidence maps used as weights and to facilitate ERP depth merging.

3.1 Depth Estimation from Tangent Images

Our method relies on tangential images with less distortion to solve irregular distortion in the image. Tangential images are generated by projecting the sphere onto a flat rectangular plane. The gnomonic projection is a mapping obtained by projecting the point Ps (λ, σ_c) on the sphere from the sphere center O to the point Pt (x_t, y_t) on the plane tangent to the point Pc (λ_c, σ_c). We use (λ, σ) to indicate longitude and latitude, respectively, and (x_t, y_t) to indicate 2D point position on the tangent image.

In our experiment, we used a set of tangential images with N=18 to balance velocity and performance (see Section 4.4 for relevant ablation studies). All tangential images share the same resolution and FoV. We chose this non-uniform sampling based on the fact that tangential

images of the same resolution can cover different ranges of longitudes when centered at different latitudes. To ensure that the sampling patches near the poles do not overlap to an extreme degree, we take fewer samples to cover the near-pole region in the ERP space. Since the generated tangential images are distortion-free, we can easily apply the traditional encoder-decoder CNN [24] architecture to predict the depth map of each tangential image. In order to achieve better convergence and accuracy, we used a high-performance pre-training network (for example, ResNet [13]) when initializing the encoder. We pass all N tangential images through the encoder at the same time and obtain N feature maps which will later be used as markers in the converter. For the decoder, we use a bunch of upsampled layers, then do a 3×3 convolution and skip the encoder connection.

The baselines presented so far can be considered truncated versions of [9]. We used a different tangential image rendering and network architecture than [9] to make the baseline approach more effective. Note that for our baseline, no converters, geometric fusion, or confidence maps were used, and the output depth was the average of all patches.

3.2 Geometric Perception Feature Fusion

Still, the simplicity of predicting depth maps from tangential images comes at a cost. Since depth estimates are now conducted independently, a globally consistent depth scale is no longer guaranteed. In addition, as shown in Figure 3(d), an object will be projected onto multiple tangential images from different angles and will therefore be encoded differently in different tangential images. Differences between patch depth estimates, especially in overlapping areas, can lead to significant artifacts in the final combined ERP depth map (Figure 4(e)).

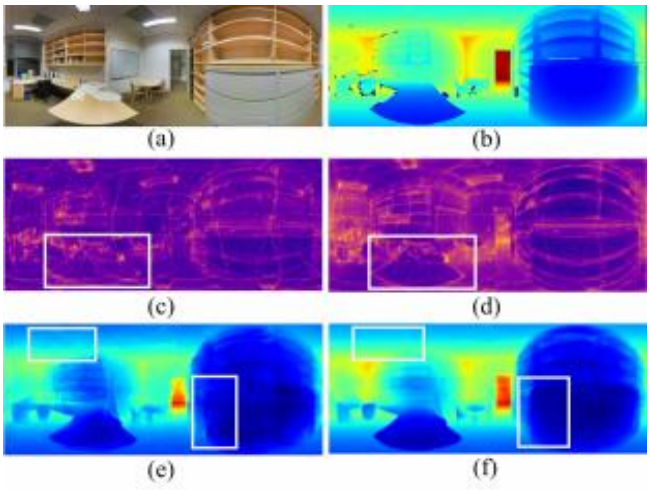

Figure 4: Description of the effectiveness of geometric perception feature fusion.

To compensate for the differences between block-by-block image features, we introduce a geometric embedding network (see Figure 1) to provide additional geometric information. For the pixel Pt (x_t, y_t) on the tangent image, we use its corresponding spherical coordinate Ps (λ, σ, γ) on the unit sphere and the center Pc (λ_c, σ_c) of the tangent image as the input attribute of the geometric embedding network. Ps lets the embed know the global position, for example, to determine whether two image pixels from two patches are related to the same spherical coordinates. However, geometric features other than Ps alone do not align different 2D features. To this end, Pc is used as an additional attribute to enable the embedding to distinguish between different patches, so that the learned geometric features can converge the patch features. The adjusted features result in cleaner merging depths through a combination of tangential image features and geometric features as well as an end-to-end learning network. As shown in Figure 4(d), extracted image features with geometric embeddedness show better consistency in feature graphs merged in ERP space compared to features without geometric embeddedness shown in Figure 4(c). As a result, the final depth map outside ContextFusion shown in Figure 4(f) appears much cleaner than the baseline depth map shown in Figure 4(e).

The geometric embedding network consists of two layers of MLPS, and the 5-channel spherical properties are encoded into 64-channel feature maps. We fuse this geometric em-badding with image features of the same pixel position in the encoder by element-by-element summation. Early fusion was used to preserve more structural detail. Geogrammetric capabilities have been added to Layer 1 of ResNet encoders where we have experimentally achieved optimum performance. Notably, the additional computational costs associated with geometrically embedded modules are minimal compared to the original encod-decoder (Table 2). The geometric features of the first iteration are fixed after learning because they are independent of the image input. Only the second iteration needs to recalculate the geometric features.

The ERP RGB image is shown in (a) and the ground truth depth is shown in (b). The visualization of the feature map and the final depth map from the baseline is shown in (c) and (e), respectively. For comparison, (d) and (f) show the feature map and final depth map of the proposed ContextFusion, where the image feature is fused with the geometric feature. It was observed that our approach produced a more self-consistent feature map and a more structured depth map compared to the baseline, especially in the areas highlighted within the rectangle.

3.3 Use Transformer for Global Polymerizatio

When the ERP is broken down into a series of tan images, we no longer have an overall view of the 3D environment. To compensate for this loss, we utilize the converter architecture to aggregate information from patches in a global manner. Global polymerization is expected to improve the consistency of patch depth estimates and better re-

gression to the global scale of depth from larger FOVs.

Using the feature map extracted from the encoder, we first apply a 1×1 convolution layer to reduce the channel dimension for better efficiency. We then flatten the feature graph into N one-dimensional feature vectors $X_0=[x^1, x^2...x^N]\in R^{N*D}$, which will be used as markers in the converter. A learnable location embedded in $E_{pos}\in R^{N*D}$ is added to feature tags to retain location information in a manner similar to that proposed in [6]. Through a self-attention architecture, the converter learns to globally aggregate information from all patches to adjust the characteristics of each patch, where the aggregate weight takes into account the paired correlation of visual and positional features. The architecture of the multi-head attention converter is shown below [9].

3.4 Deep Fusion and Learnable Confidence Graphs

The above geometric embedding and converter modules significantly reduce the differences between different slice depth estimates. However, deep merging does not achieve seamless fusion at the pixel level. To further improve the merge (Figure 3(b)), in addition to deep regression, we asked the network to predict the confidence graph for each patch. The combination depth was then calculated as a weighted average of all patch depth predictions, and confidence scores were used as weights. In detail, two separate regression layers are attached to the decoder, one for depth regression and the other for confidence score regression. Prior to merging, both the depth chart (Figure 3(e)) and the confidence chart (Figure 2(f)) underwent a reverse gnomonic transformation to map to the ERP domain.

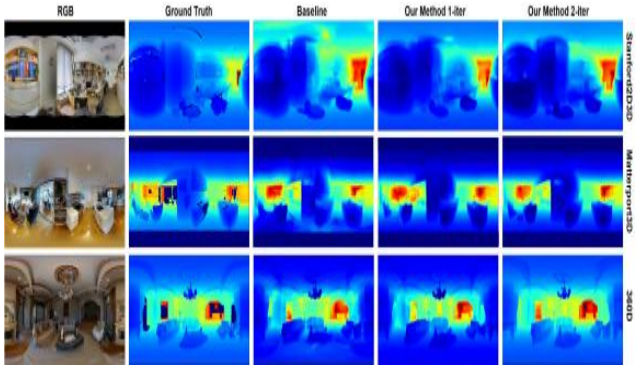

Figure 5: Qualitative results of Stanford2D3D [8], Matterport3D [9] and D [10]. From left to right: ERP RGB input, ground truth depth, depth output from baseline, depth output from our method 1-iter and 2-iter. Our method (1-iter, 2-iter) yields more structural depth than the baseline (as described in Section 3.1), looks sharp along these object boundaries, and is smooth inside the surface.

3.5 Iterative Depth Refinement

Geometric embedding uses spherical coordinates (λ, σ, γ) corresponding to tangential image pixels for geometric perception fusion. γ is initially fixed because no depth information is available. The depth information will be available after one iteration and can be used to update γ and provide more accurate geometry information for the geometric embedding module. Based on this observation, we propose an iterative deep refinement scheme (see Figure 1).

In the first iteration (Section 3.2), the spherical coordinate (λ, σ, γ) of the point on the unit sphere is used for geometric embedding. For subsequent iterations, we update $\gamma \rightarrow \gamma'$, using the new depth value estimated from the previous iteration (the depth of an ERP image is defined as the distance from a point in the real world to the center of the camera). The updated properties with more precise geometry are passed into the geometric embedding network in the next iteration. Section 4 describes an ablation study to demonstrate the effectiveness of more precise geometric embedding.

4 Experiments

4.1 Datasets

ContextFusion was tested on three well-known benchmark datasets: Stanford2D3D [1], Matterport3D [3], D [31]. The Stanford2D3D [1] dataset consists of 1,413 panoramic real-world images from six large indoor areas. We followed the official division of training and testing, using the fifth area for testing and the other areas for training. We used a resolution of 512 × 1024.

Matterport3D [9] contains a total of 10,800 panoramic RGBD images of the interior. We follow the official division, 61 rooms for training and the rest for testing. We used a resolution of 512×1024 in the experiment.

D [31] is the RGBD panoramic reference provided by Zioulis *et al.* [31]. It consists of two other synthetic datasets (SunCG and SceneNet) and two real-world datasets (Stanford2D3D and Matterport3D). D contains 35,977 realistic full-view images of RGBD rendered from the four data sets above. We followed the default train test split and used a resolution of 256×512.

4.2 Implementation Details

We use the same quantitative evaluation measures as used in [31], including absolute relative error (Abs-Rel), root mean square error (RMSE), root mean square error in logarithmic space and accuracy of threshold δ_t, where t∈1.25, 1.252, 1.253.

The arrows next to the metrics indicate the direction of better performance in all tables. We used PyTorch to implement our network and trained on two Nvidia RTX Gpus. We used the default setting of the Adam optimizer [15] with an initial learning rate of 0.0001 and

the cosine annealing learning rate strategy [20]. We trained Stanford2D3D [1] for 80 periods and Matterport3D [3] and D [31] for 60 periods. The default number of patches we used was 18. Our default patch size for Stanford2D3D [1] and matterport [3] is 256×256 and patch FoV is 80°.For D [3], we used 128×128 as the patch size. In these experiments, we used the pre-trained ResNet [13] as the image encoder. The network is trained end-to-end, and all iterations use the same model. For the loss function, after [17,20], we adopted BerHu loss [17] to conduct in-depth supervision. The final loss is the sum of the depth losses of all iterations.

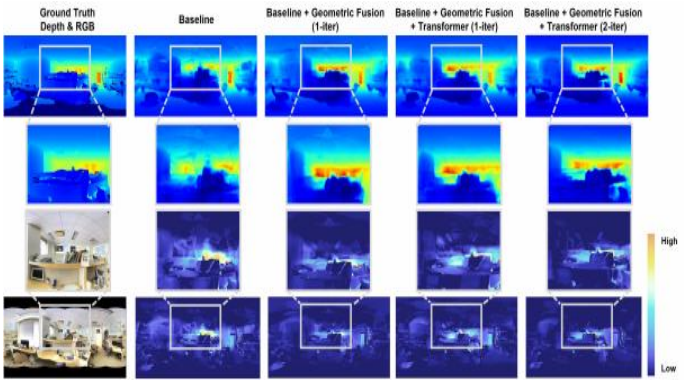

Figure 6: Qualitative results of Stanford2D3D [8], Matterport3D [9] and D [10]. From left to right: ERP RGB input, ground truth depth, depth output from baseline, depth output from our method 1-iter and 2-iter. Our method (1-iter, 2-iter) yields more structural depth than the baseline (as described in Section 3.1), looks sharp along these object boundaries, and is smooth inside the surface.

4.3 Overall Performance

We show the performance of our model and compare it to the existing methods in Table 1. In order to make a fair comparison, we omit the method of using supervision signals beyond depth [7] and the self-supervision method [30]. For all data sets, we show the results of 1 iteration (1-bit iteration) and 2 iteration (2-bit iteration). We demonstrate in Table 1 that our method is able to outperform all existing methods on Matterport3D [3] and achieve performance comparable to the current state of the art on D, even with a 1-bit setup. Using the 2-bit setup, our algorithm is 21.4$% higher than BiFuse [28] on Stanford2D3D (Abs-Rel), 56.1% higher than Bifuse [17] on Matterport 3D (Abs-Rel), and 30% higher than Bifuse [28] on D (Abs-Rel). Compared with UniFuse [14], our method improved 6.3% on Stan-ford2D3D (Abs-Rel) and 15.3% on Matterport3D (Abs-Rel). 7.7% improvement on D (Abs Rel) Note that our approach reduced ABs-rels by 7.9% compared to ODE-CNN [5] with additional sensor input. The qualitative results of our approach can be visualized in Figure 5. As observed, our methods (1-iter and 2-iter) improve the baseline, which is a direct customization [9], significantly reducing erroneous depth maps and restoring clearer boundaries and smoother surfaces.

4.4 Ablation Studies

- Individual component research.
  We examined the effectiveness of our approach by adding one key component at a time (Table 2 and Figure 6). Baseline experiments were performed without transformers or geometric fusion using ResNet34 as the encoder. We ran an experiment on Stanford2D3D using an 18-patch configuration with 256 x 256 patch size and 80∘ FoV company. As can be seen from Table 2, geometric sensing fusion adds less than 2K parameters, which can significantly improve Abs-Rel by 9.7%. Although very light in weight, the geometric fusion section proved to be very beneficial. The addition of transformer increases about 19M parameters and improves the performance by 5.7% (Abs-Rel). Combined with the transformer and geometric fusion, performance improved significantly by 15.4% (ABS-REL) in a 1-liter setting and 16.4% (Abs Rel) in a 2-liter setting. The qualitative results are shown in Figure 6. As observed, as we add more modules to the pipeline, the output depth map seems to show fewer artifacts and more structural detail. At the same time, the visual error graph clearly shows the downward trend of the estimation error.

- Patch size and number of patches.
  The size and number of patches affect the accuracy and efficiency of the method. In this study, our goal was to find the optimal balance between efficiency and performance. In theory, neither a large patch size nor a large number of patches are needed, as both will result in higher computational complexity. However, Table 3 also shows that the patch size cannot be too small, as the monocular depth estimation requires a FoV large enough to assume the depth scale. We have also observed that increasing the number of patches (e.g. ¿=26) degrades performance because more patches also increase the overlap area, which in turn may exacerbate the discrepancy problem. Therefore, we chose to use a relatively small number of patches N=18 with a relatively large resolution of 256×256 to strike a balance between efficiency and performance.

- Image encoder and iteration times.
  We compare the performance of different image encoders. As shown in Table 4, the performance of ResNet34 [13] is superior to that of ResNet18 with higher complexity. This shows the potential of our approach as it can be combined with more complex encoder networks. We also looked at the effect of iteration. We used a 2-iteration framework for training because we expected the trained network to be

Table 1: Quantitative results of depth estimation of Stanford2D3D ,Matterport3D and D data sets. It is worth noting that our method ContextFusion achieves state-of-the-art performance in all datasets, significantly better than the existing work.

Datasets	**Methods**	**Abs Rel ↓**	**Sq Rel ↓**	**RMSE ↓**	**RMSE(log) ↓**	$\delta 1 \uparrow$	$\delta 2 \uparrow$
Stanford2D3D [1]	FCRN [17]	0.1887	0.5774	0.7230	0.2907	0.9207	0.9731
	Re-trained [28]	0.1898				0.9212	0.9742
	Effective with fusion [28]	0.1009				0.9525	0.9864
	UniFuse with fusion [14]	0.6213				0.4679	0.9659
	HoHoNet [23]					0.9578	0.9884
	ContextFusion Ours (1-iter)	**0.0961**	**0.3450**	**0.3715**	**0.1699**	**0.9894**	**0.9904**
	ContextFusion Ours (2-iter)	0.0954	0.3447	0.3713	0.1694	0.9890	0.9901
Matterport3D [3]	FCRN	0.2409	0.7464	0.6704	0.3436	0.8495	0.9666
	Re-trained	0.2200				0.8498	0.9680
	Effective with fusion	0.2003				0.8814	0.9781
	UniFuse with fusion	0.5535				0.6250	0.9744
	HoHoNet	0.1732	0.5025	0.5913	0.2905	0.9266	0.9828
	ContextFusion Ours (1-iter)	**0.0909**	**0.1768**	**0.2252**	**0.1275**	**0.9975**	**0.9996**
	ContextFusion Ours (2-iter)	0.0902	0.1765	0.2197	0.1269	0.9979	0.9997
D [31]	FCRN	0.6716	2.9411	2.3498	0.8220	0.7295	0.9015
	Re-trained	0.5589				0.7392	0.9415
	Effective with fusion	0.3770				0.7913	0.9681
	UniFuse with fusion	1.0000				0.9475	0.9535
	HoHoNet	0.1487	0.2911	0.5473	0.2204	0.9713	0.9940
	X-FCN-L	0.4500				0.9825	0.9980
	Scaredś convolution	0.0835		0.2052	0.0791	0.9986	0.9999
	ODE-conv			0.5713	0.2348	0.9960	0.9980
	ContextFusion Ours (1-iter)	**0.0469**	**0.0127**	**0.1890**	**0.0735**	**0.9981**	**0.9998**
	ContextFusion Ours (2-iter)	0.0458	0.0127	0.1839	0.0735	0.9986	0.9999

Table 2: Ablation of individual components. Starting with the baseline approach without geometric fusion or transformer, we add one component at a time. We used ResNet34 for all.

Methods	**# Params**	**FPS**	**Abs Rel**	**Sq Rel**	**RMSE**
Baseline	66.2M	8.9	0.3188	2.5672	0.4258
Baseline + geometric fusion (1-iter)	66.2M (2.3K)	8.8	0.3324	0.0768	0.5024
Baseline + geometric fusion + transformer (1-iter)	86.4M (+22.4M)	8.6	0.2354	0.05463	0.4237
Baseline + geometric fusion + transformer (2-iter)	**87.2M (+21.8M)**	**5.4**	**0.0883**	**0.0435**	**0.4034**

Table 3: Comparing different patch configurations and their performance metrics.

#patch	Patch size	Patch FoV	Abs Rel↓	Sq Rel↓	RMSE↓
9	256x256	130	0.1203	0.0879	0.4523
10	128x128	90	0.1108	0.0928	0.3989
10	**256x256**	**90**	**0.0998**	**0.0456**	**0.3408**
33	256x256	70	0.2207	0.0876	0.4204
56	128x128	60	0.1809	0.0079	0.5431

Table 4: Exploring ablation effects through a comparative study of encoder models and iteration numbers.

Encoder	#iters	FPS↑	Abs Rel↓	Sq Rel↓	RMSE↓
ResNet18	1	**9.2**	0.1192	0.0702	0.5324
ResNet18	2	5.1	0.0964	0.0704	0.5089
ResNet18	3	4.2	**0.0889**	0.0783	**0.4327**
ResNet18	4	3.0	0.0789	**0.0598**	0.5024
ResNet34	1	**10.2**	0.0902	0.0705	0.5237
ResNet34	2	6.1	0.1003	0.0624	**0.4989**
ResNet34	3	4.1	**0.0998**	**0.0603**	0.5324
ResNet34	4	3.2	0.0992	0.0627	0.5499

able to handle different types of 3D coordinates. For the tests, we compared 1-4 iterations on each trunk. As you can see from Table 4, there is a significant improvement from 1 to 2 litres, a slight improvement from 2 to 3 litres, and no improvement from 3 to 4 litres. Considering the performance and speed trade-offs, we chose a 1 - or 2-bit setup.

5 Conclusion

In this paper, we propose a new pipeline ContextFusion for monocular depth estimation. In order to solve the spherical distortion problem in the image and improve the scalability of high-resolution input, we use the tangent image representation based on the dwarf projection. To mitigate differences between patches, we introduce a geometric perception fusion mechanism that fuses 3D geometric features with image features. A self-focused converter has been integrated into our pipeline to globally aggregate information from patches for more consistent patch predictions. We further extend geometric perception fusion with an iterative refinement scheme that further improves depth estimation with more structural details. We demonstrate that ContextFusion effectively reduces distortion and significantly improves depth estimation performance. Our experiments show that our approach achieves state-of-the-art performance on several data sets.

Acknowledgments

This article has been funded by the Fifth Research Project on Vocational Education Teaching Reform in Jiangsu Province (Project No. ZYB705), Title: Study on the Effectiveness of Online Learning Intervention Strategies from the Perspective of Learning Analysis in the Post Pandemic Erat.The authors gratefully acknowledge the anonymous reviewers for their valuable comments.

References

[1] Iro Armeni, Sasha Sax, Amir R. Zamir, and Silvio Savarese, "Joint 2d-3d-semantic data for indoor scene understanding," 2017.

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko, "End-to-end object detection with transformers," 2020.

[3] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, and Yinda Zhang, "Matterport3d: Learning from rgb-d data in indoor environments," 2017.

[4] Hong Xiang Chen, Kunhong Li, Zhiheng Fu, Mengyi Liu, and Yulan Guo, "Distortion-aware monocular depth estimation for omnidirectional images," *IEEE Signal Processing Letters*, vol. PP, no. 99, pp. 1–1, 2021.

[5] Xinjing Cheng, Peng Wang, Yanqi Zhou, Chenye Guan, and Ruigang Yang, "Ode-cnn: Omnidirectional depth extension networks," 2020.

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, and Sylvain Gelly, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations*, 2021.

[7] Marc Eder, Pierre Moulon, and Li Guan, "Pano popups: Indoor 3d reconstruction with a plane-aware network," 2019.

[8] Marc Eder, True Price, Thanh Vu, Akash Bapat, and Jan Michael Frahm, "Mapped convolutions," 2019.

[9] Marc Eder, Mykhailo Shvets, John Lim, and Jan Michael Frahm, "Tangent images for mitigating spherical distortion," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[10] Clara Fernandez-Labrador, Jose M. Facil, Alejandro Perez-Yus, Cedric Demonceaux, and Josechu Guerrero, "Corners for layout: End-to-end layout recovery from 360 images," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1255–1262, 2020.

[11] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow, "Unsupervised monocular depth estimation with left-right consistency," *IEEE*, 2017.

[12] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel Brostow, "Digging into self-supervised monocular depth estimation," 2018.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," *IEEE*, 2016.

[14] Hualie Jiang, Zhe Sheng, Siyu Zhu, Zilong Dong, and Rui Huang, "Unifuse: Unidirectional fusion for 360 panorama depth estimation," *IEEE Robotics and Automation Letters*, vol. PP, no. 99, pp. 1–1, 2021.

[15] Diederik Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *Computer Science*, 2014.

[16] Georgios Kopanas, Julien Philip, Thomas Leimkühler, and George Drettakis, "Point-based neural rendering with per-view optimization," *Computer Graphics Forum*, vol. 40, no. 4, 2021.

[17] Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab, "Deeper depth prediction with fully convolutional residual networks," *IEEE*, 2016.

[18] Yuyan Li, Zhixin Yan, Ye Duan, and Liu Ren, "Panodepth: A two-stage approach for monocular omnidirectional depth estimation," 2022.

[19] Juan Ting Lin, Dengxin Dai, and Luc Van Gool, "Depth estimation from monocular images and sparse radar data," 2020.

[20] Ilya Loshchilov and Frank Hutter, "Sgdr: Stochastic gradient descent with warm restarts," *arXiv e-prints*, 2016.

[21] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid, "Segmenter: Transformer for semantic segmentation," 2021.

[22] Yu Chuan Su, Dinesh Jayaraman, and Kristen Grauman, "Pano2vid: Automatic cinematography for watching 360^{circ} videos," in *Springer, Cham*, 2016.

[23] Cheng Sun, Min Sun, and Hwann Tzong Chen, "Hohonet: 360 indoor holistic understanding with latent horizontal features," 2020.

[24] Lidia Talavera-Martínez, Pedro Bibiloni, and Manuel González-Hidalgo, "An encoder-decoder cnn for hair removal in dermoscopic images," 2020.

[25] Keisuke Tateno, Nassir Navab, and Federico Tombari, "Distortion-aware convolutional filters for dense prediction in panoramic images," in *Computer vision - ECCV 2018*, 2018.

[26] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin, "Attention is all you need," *arXiv*, 2017.

[27] Fu En Wang, Hou Ning Hu, Hsien Tzu Cheng, Juan Ting Lin, Shang Ta Yang, Meng Li Shih, Hung Kuo Chu, and Min Sun, "Self-supervised learning of depth and camera motion from 360 deg videos," 2018.

[28] Fu En Wang, Yu Hsuan Yeh, Min Sun, Wei Chen Chiu, and Yi Hsuan Tsai, "Bifuse: Monocular 360 depth estimation via bi-projection fusion," *IEEE*, 2020.

[29] Jiachen Yang, Tianlin Liu, Bin Jiang, Wen Lu, and Qinggang Meng, "Panoramic video quality assessment based on non-local spherical cnn," *Multimedia, IEEE Trans. on (T-MM)*, vol. 23, no. 000, 2021.

[30] Nikolaos Zioulis, Antonis Karakottas, Dimitrios Zarpalas, Federico Alvarez, and Petros Daras, "Spherical view synthesis for self-supervised 360 depth estimation," *IEEE*, 2019.

[31] Nikolaos Zioulis, Antonis Karakottas, Dimitrios Zarpalas, and Petros Daras, "Omnidepth: Dense depth estimation for indoors spherical panoramas," 2018.

[32] Shuangquan Zuo, Yun Xiao, Xiaojun Chang, and Xuanhong Wang, "Vision transformers for dense prediction: A survey," *Knowledge-based systems*, 2022.

Biography

Chunhua Wang biography. Chunhua Wang is with School of Information Engineering, Suzhou Industrial Park Institute of Service Outsourcing. Her interests are deep learning, computer application technology, image processing.

Ying Gao biography. Ying Gao is with School of Information Engineering, Suzhou Industrial Park Institute of Service Outsourcing. Her interests are daya analysis, computer application technology,pattern recongnition.

Zhenyu Wang biography. Zhenyu Wang is with Faculty of computer science and technology,Qilu University of Technology (Shandong Academy of Sciences). His interests are telecommunications engineering, data analysis,computer application.