Upright Adjustment of Panoramic Images Based on 3D Coordinate Mapping Matrix

Han Li[a,1], **Yilin Guo**[a,1], **Lei Zhong**[b] **and Jianfeng Li**[a,*]

[a]College of Electronic and Information Engineering, Southwest University, China
[b]The University of Edinburgh, UK

Abstract. Non-upright panoramic images often suffer from distortion due to camera tilt, which compromises the accuracy of downstream tasks. We propose a novel panoramic upright adjustment method based on 3D coordinate mapping estimation, which fundamentally reformulates the task from a 2D projection problem to a 3D unit spherical mapping problem. Our method employs an end-to-end neural network to directly generate an upright panoramic image from a non-upright input. The key innovation of our approach lies in the use of a 3D Coordinate Mapping Matrix (3D CMMatrix) instead of the traditional 2D CMMatrix. By leveraging the inherent 3D structure of panoramic images, our method effectively captures the spatial continuity of the entire spherical space, eliminating the discontinuous issues that arise at the edges of non-upright panoramic images when using 2D coordinate mapping. The network consists of an encoder that extracts tilt features from the non-upright image and transforms them into a 3D CMMatrix, and a decoder that gradually upsamples the 3D CMMatrix to match the resolution of the original image. This 3D-based approach not only resolves edge artifacts but also significantly improves the overall quality of the upright image. Experimental results demonstrate that our proposed method achieves state-of-the-art performance, outperforming existing methods.

1 Introduction

Non-upright panoramic images are caused by the tilt of panoramic cameras. When the camera is tilted, its direction becomes misaligned with the scene, resulting in distorted images as shown in Figure 1(a). In contrast, the corresponding upright image of the same scene, shown in Figure 1(b), maintains proper alignment with the scene. It is evident that Figure 1(b) is more suitable for scene understanding because it preserves the spatial consistency of the scene and reduces visual distortions. Therefore, today's downstream panoramic image tasks, such as panoramic depth estimation [30, 2, 20], panoramic semantic segmentation [24, 28, 29], and panoramic scene understanding [8, 31, 32], are all based on upright images like Figure 1(b).

The process of correcting the camera's incorrect orientation to generate upright images is known as upright adjustment. To achieve this, a straightforward approach is to first estimate the inclination of the camera and then compensate for the estimated inclination by resampling the non-upright image. This method, commonly referred to as the traditional method [23, 10, 11, 12], divides the upright adjustment task into two subtasks: inclination estimation and image resampling. However, this approach relies on accurate inclination estimation and may introduce artifacts during resampling.

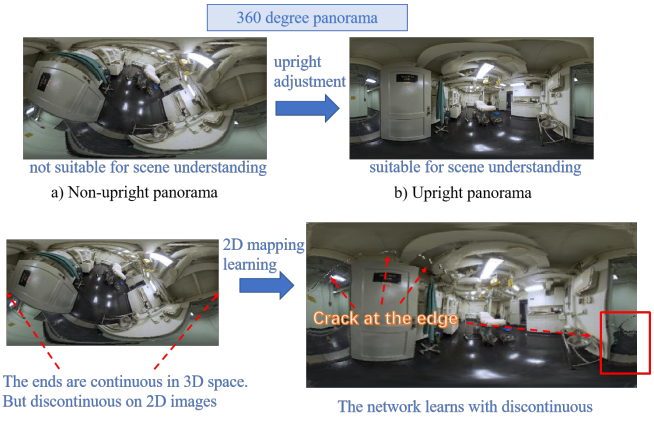

Figure 1. Due to the limitations of two- dimensional projection, current proposed 2D CMMatrix based networks fails to connect the left and right sides in the learning process, leading to the appearance of discontinuity.

Another method is to use an end-to-end deep learning network to perform upright adjustment of panoramic images. Unlike the traditional method, which requires explicit inclination estimation, the end-to-end network learns to directly map non-upright images to upright images. Through end-to-end learning, the network model automatically extracts relevant features directly from the input data. For example, Chen [4] et al. proposed an end-to-end network that achieved the first real-time online upright reconstruction of panoramic images using deep learning. Similarly, Liu [19] et al. proposed a network that extracts features to obtain pixel displacement information, enabling the direct generation of upright panoramic images from non-upright inputs.

When adjusting non-upright images, some current practices treat panoramic images as two-dimensional images, which is a straightforward approach. Following this idea, we utilized the ConvNeXt network [21] to extract the latent space of the image and subsequently reshaped this latent space into a 2D Coordinate Mapping Matrix (2D CMMatrix). The 2D CMMatrix is then upsampled to the same size as the input image. However, after conducting network training, we observed that the generated upright images exhibited cracks, particularly at the left and right boundaries of the original image, as shown in Figure 1(c).

This issue arises because the left and right sides of a panoramic

* Corresponding Author. Email: popqlee@swu.edu.cn.
[1] Equal contribution.

image represent a continuous space in the real world, but the two-dimensional projection used in the network fails to capture this continuity. As a result, the network cannot properly connect the left and right boundaries during the learning process, leading to excessive error oscillations at the edges. For example, as depicted in Figure 1(c), the leftmost and rightmost points of the panoramic image should represent adjacent scenes. However, the network trains them as two separate parts, causing the edges to fail to converge and resulting in visible cracks.

To solve this issue, we employ two strategies: (1) Inspired by the Pano-style Shift Windowing scheme (PSW) [17], we extend the edges of the non-upright panoramic image before performing convolutional operations. This extension compensates for the information loss caused by the convolution operation and helps the network better handle the continuity of the panoramic image. (2) considering that panoramic imaging is based on continuous unit spherical coordinates, we propose imposing coordinate constraints in three-dimensional space. Specifically, we reorganize the 2D CMMatrix into a 3D CMMatrix, which better represents the spatial continuity of the panoramic image. By performing upsampling in three-dimensional space, we ensure that the left and right boundaries of the image are properly connected, eliminating the cracks observed in the 2D approach.

Ultimately, this method solves the issue of discontinuities and cracks in the image and significantly improves the quality of the upright images. To the best of our knowledge, we are the first to adopt a three-dimensional coordinate mapping matrix to constrain the network for upright adjustment. Our main contributions are as follows:

1. We innovatively use a 3D CMMatrix instead of a 2D CMMatrix to constrain the end-to-end upright adjustment network and perform upsampling in three-dimensional space, which better captures the spatial continuity of panoramic images.

2. We extend the image borders to compensate for CNN's convolutional information loss and design an upsampling block using group convolution, where the x, y, and z axes do not interfere with each other.

3. We employ pyramid constraints layer by layer to guarantee the stable and convergent learning of the network for the mapping matrix. The experimental results demonstrate that our proposed method achieves superior performance compared to existing approaches.

2 Related works

Over the years, numerous studies have been conducted on upright adjustment. In this section, we will introduce two aspects: traditional upright adjustment methods and current deep-learning-based approaches.

2.1 Traditional image upright adjustment

Traditional upright adjustment methods primarily rely on inclination estimation, where the rotation angle is calculated to perform upright correction. For example, Gallagher et al. [7] proposed an algorithm that automatically removes the tilted appearance of images by detecting vanishing points and calculating the rotation angle. It assumes that the vanishing points correspond to the principal directions of the scene (e.g., horizontal and vertical lines), enabling the algorithm to estimate the tilt and correct the image orientation. This approach is particularly effective for structured environments with clear geometric patterns. Lee et al. [15] introduced a method based on human perception research to straighten human constructs in images. It uses a combination of edge detection and line segment analysis to identify dominant orientations and adjust the image accordingly. An et al. [1] proposed an adjustment method for planar objects based on the 2D Manhattan World assumption, which posits that most structures in the scene align with a dominant set of orthogonal directions. By detecting these directions, the method estimates the tilt and performs upright adjustment. Gakne et al. [6] presented a method that estimates rotation by detecting vertical and horizontal lines of vanishing points. Jung et al. [11] adopted the Manhattan World assumption to adjust 360-degree spherical panoramic images without prior information. Kawai et al. [14] proposed an automatic algorithm for robust upright adjustment in both indoor and outdoor scenes. It combines line segment detection with a probabilistic model to estimate the tilt angle, ensuring reliable performance across diverse environments.

These traditional methods rely on feature extraction and assumptions, such as detecting vanishing points and vertical/horizontal lines, to estimate tilt and perform upright adjustment. While effective, these approaches are often cumbersome and complex. In contrast, our method employs an end-to-end network that directly generates upright panoramic images from non-upright inputs, significantly simplifying the process.

2.2 Deep-learning-based image upright adjustment

With the advancement of deep learning, many methods have been proposed for upright adjustment. Jeon et al. [10] introduced a framework based on convolutional neural networks to estimate 2D rolling and adjust rotated panoramic images. Jung et al. [12] proposed a method to automatically estimate the orientation of VR images and return upright results. Davidson et al. [5] integrated geometric and semantic cues to reduce non-upright errors during image capture. By combining information from scene geometry and object semantics, the method achieves more robust tilt estimation, even in challenging environments. Jung et al. [13] proposed a network combining CNN and GCN to map images onto a sphere for upright adjustment. The CNN extracts local features, while the GCN captures global relationships between different parts of the image. This hybrid approach enables the model to handle both local distortions and global orientation, resulting in highly accurate upright corrections. Shan et al. [26] introduced a multi-scale shallow geometric feature attention mechanism to address geometric deformations caused by camera tilt. The attention mechanism ensures that the most relevant features are prioritized, improving the overall accuracy of the upright adjustment. Chen et al. [4] proposed an end-to-end network to estimate rolling and pitching angles for upright reconstruction. The network is trained to directly predict the tilt angles from the input image, eliminating the need for intermediate steps. Most of these methods still focus on tilt estimation, similar to traditional approaches. Recently, Liu et al. [19] transformed the task into a pixel displacement mapping problem, generating upright images from non-upright inputs. However, their method still processes images in two-dimensional coordinates.

Unlike these approaches, we reorganize the 2D mapping coordinate matrix into a 3D coordinate mapping matrix (3D CMMatrix). For the first time, we use a 3D CMMatrix to constrain the network, aligning more closely with the inherent logic of panoramic images. Experimental results demonstrate that our method achieves superior adjustment outcomes, offering a robust solution for panoramic upright adjustment.

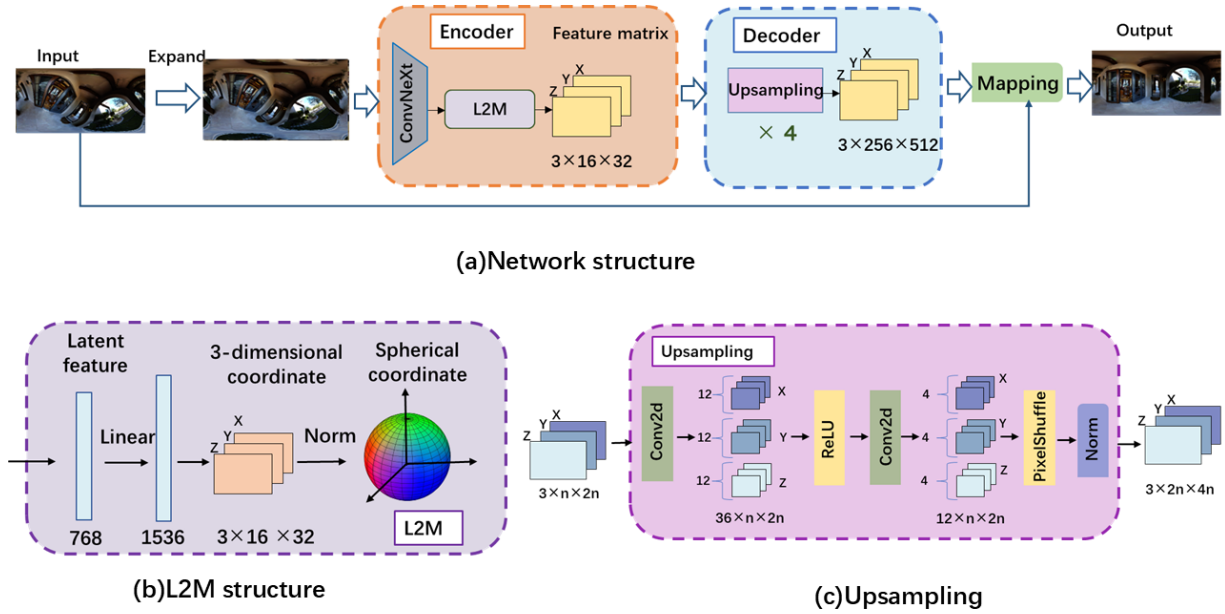

Figure 2. The proposed network Frame: (a) Network structure: Encoder for generating the small-sized 3D CMMatrix. Decoder for the refined 3D CMMatrix by group convolution and pixelshuffle strategies. Mapping for upright adjustment by 3D CMMatrix. (b) L2M structure: converting the latent features to spherical coordinates. (c) Upsampling: refine the 3D CMMatrix to the same size of the input.

3 Methods

3.1 Network Framework

The comprehensive network architecture is illustrated in Figure 2(a), where a non-upright panoramic image serves as the input. Initially, border expansion is performed to mitigate information loss at the image borders due to 2D imaging. In the encoder stage, the ConvNeXt network is used to extract the non-upright panorama features. The ConvNeXt network [21] is a deep convolutional neural network for image classification tasks. It is trained on large datasets and has a deep understanding of image features [9, 16, 22]. Then, the one-dimensional features are transformed through the L2M (Latent to Matrix) model to obtain a 16×32 3D CMMatrix, which adheres to unit spherical coordinates. In the decoder stage, the 3D CMMatrix is gradually expanded through four upsampling layers to obtain coordinate information consistent with the size of the original image. Ultimately, the original image is mapped onto this refined coordinate system, yielding the desired upright image output.

3.2 Border Expansion

In our method, during the data processing stage, we employ matrices to perform slice-and-stitch operations on non-upright panoramic images, extending them by 32 pixels in all directions (up, down, left, and right). As illustrated in Figure 3(a), the original image without expansion is shown. Figure 3(b) schematically represents the left and right extension process. Here, the left portion of the cut image is translated and spliced directly onto the right side of the original image, and the same process is applied to the other side to achieve extension of the left and right boundaries. Similarly, Figure 3(d) illustrates the up-and-down extension process, where the image that has already undergone left-right extension is used. In this step, the

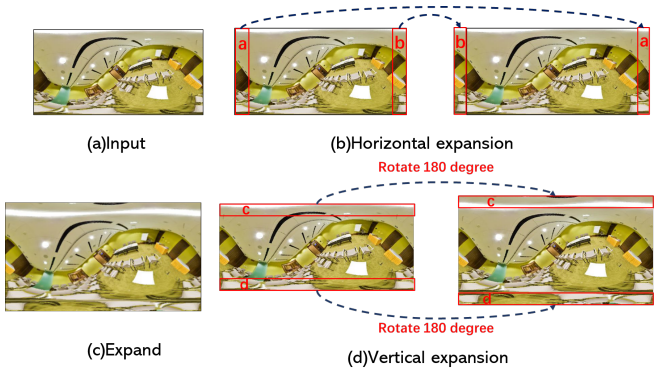

Figure 3. Border expansion: (a) the 360 panorama input. (b) horizontal expansion. (c) expanded panorama. (d) vertical expansion.

top part of the image is sliced, rotated 180°, and spliced at the top of the image, while the bottom extension is performed analogously to complete the up-and-down extension of the image. The resulting expanded image is shown in Figure 3(c). By cutting and rotationally splicing specific parts of the non-upright image, we enlarge the size of the original panoramic image and properly extend its boundary regions. This enables the network to extract more continuous features from the boundaries of the non-upright panoramic image, ultimately enhancing the accuracy of image mapping constraints.

3.3 Latent to Matrix

Figure 2(b) introduces the entire process of the L2M module. It takes the latent feature space extracted by the pre-trained ConvNeXt-tiny network [21] as its input. The input latent feature space is a one-dimensional vector with 768 channels. To obtain the desired vector, a fully connected layer is used to integrate the features, outputting a

tensor with 1536 channels. We consider that although different non-upright images may have different scenes, if they have the same tilt angle, their panoramic image representations will exhibit a common pattern. As shown in Figure 4, when two upright images from different scenes are both rotated by the same angles (pitch 45° and roll 45°), similar curved shapes emerge in the images. This observation guided us in designing a mapping matrix to represent deep features, where each element in the matrix represents a mapping coordinate for the final upright adjustment of the mapped image.

Figure 4. Two upright images from different scenes are both rotated by the same angles (pitch 45° and roll 45°), similar curved shapes emerge in the images

The current Liu's method [19] is based on pixel displacement. Its essence lies in using a 2D coordinate mapping matrix. However, this approach has limitations when applied to panoramic images. Essentially, panoramic images should have a continuous space between their left and right borders. Mapping them onto a 2D matrix ignores the continuity of panoramic images, resulting in a disconnection due to the limitations of 2D projection. During the learning process, the network struggles to link the left and right sides, leading to excessive error oscillations at the edges and thus ambiguities arise. Specifically, the leftmost point and the rightmost point in the image are adjacent scenes. Due to the presence of errors, convergence cannot be achieved at the edges, resulting in cracks appearing in the final mapped and vertically adjusted panoramic image.

However, the mapping coordinates of panoramic images are essentially continuous in 3D space, without any segmentation issues. Therefore, unlike previous methods, we innovatively use a three-dimensional matrix to transform potential features into unit spherical coordinates. Therefore, unlike previous methods, we propose an innovative approach that leverages a three-dimensional matrix to transform latent features into unit spherical coordinates. This transformation is designed to capture the geometric and spatial relationships within the data more effectively, enabling a richer representation of the underlying structure. The process begins by reshaping the one-dimensional feature vector into a set of three-dimensional (x, y, z) coordinates. Specifically, the first 512 channels are assigned to the x-coordinate, the next 512 to the y-coordinate, and the final 512 to the z-coordinate. This reshaping operation effectively maps the high-dimensional feature space into a 3D spatial domain, where each point (x_i, y_i, z_i) represents a unique feature combination. Next, we normalize these 3D coordinates to lie on the surface of a unit sphere. This normalization is achieved by dividing each coordinate by its Euclidean norm, ensuring that all points satisfy the condition $x_i^2 + y_i^2 + z_i^2 = 1$. Following normalization, we organize the normalized 3D coordinates into a structured grid. Finally, this process yields a 3D Coordinate Mapping Matrix (CMMatrix) with dimensions of 3×16×32. The 3D CMMatrix is a compact and structured representation of the transformed features, where each element corresponds to a specific spatial location on the unit sphere.

3.4 Upsampling Block

In the encoder, we have already obtained the tilt feature information of the image with a size of 16×32. Since we need to map this information to the input image of 256×512, we need to perform upsampling. During the upsampling process of the coordinate mapping matrix, we found that there is no strict continuity constraint between each coordinate and its neighboring coordinates in the matrix. As shown in Figure 4, point P in Figure (b) will be corrected to point P' in Figure (a), while its adjacent points will be corrected to points like P" after adjustment. Therefore, unlike previous methods that use neighboring interpolation for upsampling, we employed PixelShuffle upsampling to ensure that each coordinate information is calculated independently and not contaminated. This process transforms a low-resolution input image of H × W into a high-resolution image of rH × rW through sub-pixel operations. Considering that the coordinates of x, y and z are also independent of each other, to prevent confusion of feature information among the three channels during the convolution process, we adopted group convolution. The specific process is shown in Figure 2(c). After passing through four upsampling modules, we obtain a 3D coordinate mapping matrix with the same resolution as the input image. This ensures that information from each channel is generated independently, maintaining the precision of the coordinate information.

To ensure the stability of upsampling at each layer, we adjust the size of the convolutional kernels for each layer. Specifically, for the first layer with an image size of 16x32, we set the convolutional kernel size to 3; for the second layer with an image size of 32x64, we set the convolutional kernel size to 5; for the third layer with an image size of 64x128, we set the convolutional kernel size to 7; and for the last layer with an image size of 128x256, we set the convolutional kernel size to 9. This helps maintain a consistent workload for each layer of the network. Additionally, the 3D coordinate mapping matrix obtained from each upsampling module is used as an output for the decoder.

3.5 Mapping

The 3D coordinate mapping matrix obtained through upsampling requires mathematical operations to convert the 3D coordinates into 2D coordinates for image mapping. First, let us introduce the mathematical formulas for converting 2D coordinates to 3D coordinates. Based on the length and width of the panorama, the corresponding zenith angle θ and azimuth angle ϕ on the spherical surface can be mapped. Then, using ϕ_i and θ_i, the orthogonal coordinates (x, y, z) of the pixel point P(u_p, v_p) on the unit sphere are calculated. The two formulas are as follows:

$$\phi_i = 2\pi\frac{u_p}{W}, \theta_i = \pi\frac{v_p}{H} \tag{1}$$

$$x = \cos\phi_i \sin\theta_i, y = \sin\phi_i \sin\theta_i, z = \cos\theta_i \tag{2}$$

W: Width of the image, H: Height of the image

We use the inverse functions of equation1 and equation2 to transform the mathematical formula of the obtained 3D coordinates into 2D coordinates:

$$\phi_i' = \arctan 2(\Delta y_p, \Delta x_p), \theta_i' = \arccos(\Delta z_p) \tag{3}$$

$$\Delta u = \frac{W}{2\pi}\phi_i', \Delta v = \frac{H}{\pi}\theta_i' \tag{4}$$

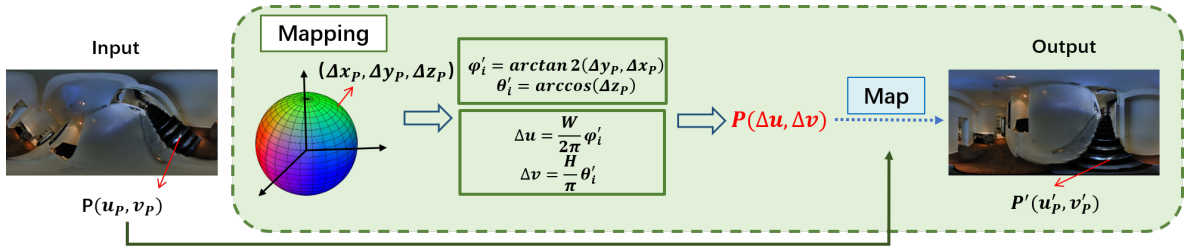

Figure 5. Mapping procedure: Utilizing a 3D unit sphere model, coverting the 3D CMMatrix to 2D coordinates in order to adapt the final 2D panorama upright adjustment.

we generate the estimated 3D coordinate point $(\Delta x_p, \Delta y_p, \Delta z_p)$ through the network, obtaining the corresponding 2D coordinate point $P(\Delta u, \Delta v)$ using equation3 and equation4. Then, we apply a simple mapping function called grid sample to generate the upright image. This function fills input values to specified positions based on the coordinate information provided by the 2D mapping coordinates. As shown in Figure 5, assuming the pixel coordinate of the non-upright image is $P(u_p, v_p)$, and the corresponding upright image coordinate is $P'(u_p', v_p')$, then $P'(u_p', v_p') = P(u_p, v_p) \bigoplus P(\Delta u, \Delta v)$ (where the symbol $\bigoplus$ represents the mapping operation in the process). By combining the image coordinate information obtained from the decoder with this function's mapping, we get the output image. By resizing the input image, it can be mapped with different-sized 3D coordinate mapping matrices obtained from the decoder, resulting in output images of different sizes.

3.6 Loss Function

In our image generation network, we employ L1 loss for regularization to achieve high-quality image generation. However, images generated solely with a single L1 constraint often suffer from defects in detail, as the network may fail to converge if constraints are only imposed at the final output stage. To address this, we introduce multi-level image constraints [18, 25], which ensure the stability of network convergence by applying constraints at multiple resolutions. Specifically, we resize the target upright image and enforce image constraints with the output images at corresponding resolutions, as illustrated in Figure 6.

We denote the Pyramid Loss as:

$$loss_{py} = loss_{32} + loss_{64} + loss_{128} + loss_{256} \tag{5}$$

Here, each term (e.g. $loss_{32}$, $loss_{64}$) represents the L1 loss computed at a specific resolution level, ensuring that the network learns to generate consistent details across all scales. This multi-resolution approach helps the network capture both global structures and fine-grained details, leading to higher-quality outputs.

In addition to the Pyramid Loss, we introduce a 3D Ground Truth Grid Loss to address the spatial variations inherent in upright adjustment tasks. Convolutional networks, by design, are translation-invariant and struggle to model global spatial transformations, such as rotations or tilts, which are critical for upright adjustment. To overcome this limitation, we impose an intermediate mapping constraint. First, we calculate the 2D mapping ground truth grid offline using the tilt angle based on a spherical model. The details of this process are described in [19]. Next, we use Equations 1 and 2 to derive the 3D Ground Truth Grid (denoted as "3D Gridtruth") from the 2D mapping ground truth grid. Finally, we apply a ground-truth constraint between the 3D CMMatrix obtained from the encoder and the "3D Gridtruth", ensuring that the network learns accurate spatial transformations during training. This intermediate constraint enables the network to better model the geometric relationships required for upright adjustment, improving both convergence stability and output quality. This is denoted as:

$$loss_{grid} = L_1\,(3D\ CMMatrix,\ 3D\ Gridtruth) \tag{6}$$

The total loss is expressed as:

$$loss = loss_{py} + loss_{grid} \tag{7}$$

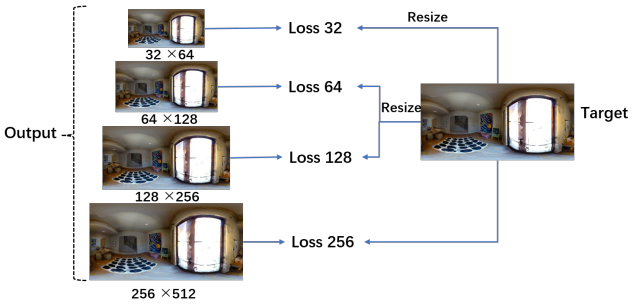

Figure 6. Pyramid Constraint, which can ensure the stability of network convergence.

4 Experiment

4.1 Dataset and Training Details

Dataset: In this paper, we use Matterport3D [3] and SUN360 [27] for evaluation like previous papers. Furthermore, we used the 3D60 dataset [33] for depth estimation experiments. For all images, we apply random angle rotations within the range of [-90°, 90°]. We allocate 70% of the images for training, 15% for validation, and the remaining 15% for testing within the dataset.

Training Details: All network models are trained using the Adam optimizer on a TITAN RTX with 24GB of memory. The batch size is set to 4. The learning rate is set to 1×10^{-4} for the first 1-20 epochs and 5×10^{-5} for subsequent epochs.

Notice: Our paper have compared all major and representative methods, following the evaluation protocols used in their original papers. Early approaches focus on angle estimation (Tables 2–3), while recent methods like Liu's [19] adopt end-to-end correction and are evaluated by image quality metrics (Tables 4–6). Chen's method [4] supports both types and appears in both settings (Tables 2 and 4). Although the metrics vary across tables, each was chosen to match the corresponding baseline.

4.2 Ablation Analysis

In this section, we conducted a comparison to analyze the effect of extending the input training images. Additionally, we introduced 3D CMMatrix Loss, the pyramidal image-level Loss, and group convolutions in the upsampling layers. An ablation study was performed to investigate the impact of them. Given the challenge in distinguishing differences in perspectives, we adopted FID (Frechet Inception Distance) to evaluate the entire test set under various conditions. FID is a widely used metric in image generation tasks, as it quantifies the similarity between generated and real images by comparing their feature distributions. A lower FID score indicates better quality and diversity in the generated outputs, making it a robust measure for assessing our model's performance across different conditions.

Each experiment was trained for the same number of epochs, as detailed in Table 1. As shown at the first row, there is a drastic decrease in image quality when we perform mapping using 2D CMMatrix. A comparison between the third row and the sixth row reveals that border expansion can help improve image quality. Furthermore, a comparison between the fifth and sixth rows shows that the incorporation of group convolutions enhances the image quality. By comparing the data in the second, fourth, and sixth rows, it is evident that the choice of loss function significantly impacts the results. Specifically, omitting multi-level image constraints and using an image constraint with a size of 16×32 both tend to reduce image quality to a certain extent and slow down convergence speed.

$loss_{16}$	$loss_{grid}$	$loss_{256}$	$loss_{py}$	Expand	Group Conv	3D	2D	FID↓
	✓		✓	✓	✓		✓	39.8227
	✓	✓		✓	✓	✓		19.9865
	✓		✓		✓	✓		19.2931
✓			✓	✓	✓	✓		18.9239
	✓		✓	✓		✓		18.7748
	✓		✓	✓	✓	✓		18.4060

Table 1. Ablation analysis

4.3 Upright Evaluation

To facilitate a direct comparison with previous works, we trained a model to estimate angles from the latent features. This model takes the 3D coordinate matrix obtained from the encoder as input, reshapes the 3D coordinates into a 1D feature vector, and processes them through fully connected layers followed by an activation function to produce two normalized values between 0 and 1. These values correspond to the predicted pitch and roll angles, which are used to estimate the tilt. Previous upright adjustment methods based on angle estimation have primarily relied on the Sun360 dataset [27]. The results of our angle estimation model are presented in Table 2. The numbers in the table represent the percentage of the difference between the estimated tilt angle of the non-upright images and the ground truth, relative to the entire test sample. We achieved complete correction for 53.3% of the testset, significantly outperforming existing methods, while 94.9% of the images had an error within 5 degrees, slightly lower than [26].

Additionally, the most recent work by Shan et al. [26] also utilized the M3D dataset for tilt angle estimation. To ensure a fair comparison, we trained and evaluated our model on the same M3D dataset. The results, presented in Table 3, show that our method generally outperforms [26] across various metrics. Notably, it can be observed that [26] exhibits significant instability, performing far better indoors than outdoors, particularly for small angle errors. This inconsistency may stem from its reliance on features that are more prevalent in indoor environments, such as structured lines and geometric patterns. In contrast, our approach demonstrates greater universality and consistency across different environments, making it more robust and reliable for real-world applications. This universality is achieved through a combination of robust feature extraction and adaptive normalization techniques, which ensure stable performance regardless of the scene's complexity or environmental conditions.

Dataset	Method	1°	2°	3°	4°	5°	12°
Sun360	[26] (CVPR)	31.2	72.9	89.8	**95.5**	**97.5**	**99.5**
	[4] (PCS)	29.9	65.3	80.3	86.3	89.2	95.2
	[5] (ECCV)	19.7	53.6	75.5	87.2	92.6	98.4
	[12] (VR)	7.1	24.5	43.9	60.7	74.2	97.9
	Ours	**53.3**	**83.0**	**90.4**	93.5	94.9	97.7
The numbers represent the percentage of the difference between estimation and groundtruth							

Table 2. Tilt angle estimation comparison on SUN360 Dataset

Dataset	Method	1°	2°	3°	4°	5°	12°
M3D	[26] (CVPR)	**70.3**	89.2	93.0	94.5	95.5	97.4
	Ours	66.2	**90.3**	**94.8**	**96.3**	**97.1**	**98.6**
The numbers represent the percentage of the difference between estimation and groundtruth							

Table 3. Tilt angle estimation comparison on M3D Dataset

4.4 Image quality, Spatial and Temporal Costs

Image Quality: The methods in above section only estimate the angles without correcting the images. Recently, two papers [4][19] has done the generated image quality evaluation. Following their evaluation criteria, we calculated the Normalized Root Mean Square Error (NRMSE) and Normalized Mean Absolute Error (NMAE) between the ground truth and the output of our model. As shown in Table 4, the quality of our generated images is superior than theirs. In Figure 7, the first column displays the input non-upright images, the second column shows the ground truth, the third column presents the direct output from our end-to-end network, and the last column represents the method by Liu et al. [19]. In comparison, the upright images generated by our network are closer to the ground truth, with fewer jagged edges.

Spatial and Temporal Costs: We tested our network on PyTorch using a GPU. As shown in Table 5, the average time for processing a single sample is 0.024s, while the method by Liu et al. [19] takes an average of 0.009s. Since our network operates in a 3D-to-2D manner, involving mathematical operations for the transformation and output constraints at each level, it requires significantly more time than the lightweight network of Liu et al.[19] . However, we believe that the improvement in image quality justifies this increase in processing time. The primary spatial cost of our network lies in the pre-trained model, which requires only 113MB of storage space, which is relatively small in the field of deep learning. It can be seen that our network effectively balances spatial and temporal costs.

Evaluation	Ours	[19] (WACV)	[4] (PCS)
NRMSE↓	**0.1512**	0.2268	0.3225
NMAE↓	**0.0829**	0.1363	0.2180
FID↓	**18.40**	31.21	-

Table 4. Image quality comparison with other methods

Evaluation	Ours	[19] (WACV)
Time↑	0.024s	0.009s
Space↓	113MB	163MB

Table 5. Image quality, time, space comparison

Input	Abs Rel↓	Sq Rel↓	RMS	RMSlog	$\delta < 1.25^3$↑
Non-upright[-90°,90°]	0.4525	0.7797	1.257	0.5211	0.7984
Upright by [19]	0.0902	0.0657	0.3522	0.1409	0.9856
Upright by ours	**0.0583**	**0.0288**	**0.2422**	**0.0924**	**0.9964**
Ground truth	0.0377	0.0102	0.1602	0.0598	0.9991

Table 6. Depth estimation on 3D60 using the same method with different input

4.5 Downstream task evaluation

In this section, we estimate depth using the method proposed by Liu et al. [20] and utilize samples from the 3D60 dataset[27] for depth estimation. The dataset samples are all ground truth, and we rotate the images by random angles within the range of [-90°, 90°] to obtain a non-upright image dataset. Then, both the non-upright images and the corrected images through our network are input into the depth estimation network. The qualitative evaluation results are shown in Table 6. Compared to the ground truth, the depth estimation performance for non-upright images is relatively poor when using depth estimation network, as existing depth estimation models are all trained on upright images, whereas the images generated by our method are closer to the ground truth. Additionally, our method outperforms that of Liu et al.[19]. This experiment demonstrates that our network can output high-quality upright images for depth estimation. Figure 8 shows our visual depth results.

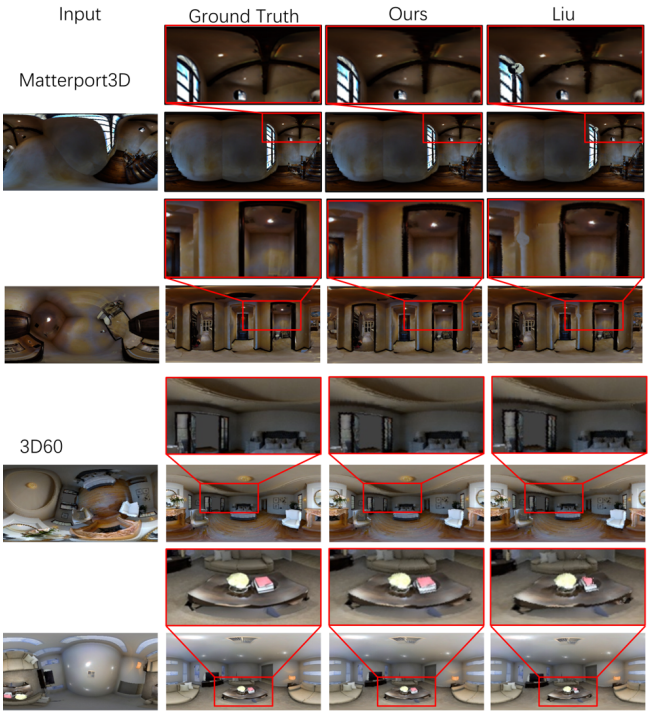

Figure 7. The left is the input, the second column is the Ground Truth, the third column is the result of our method, the 4th column is the result of Liu[19]

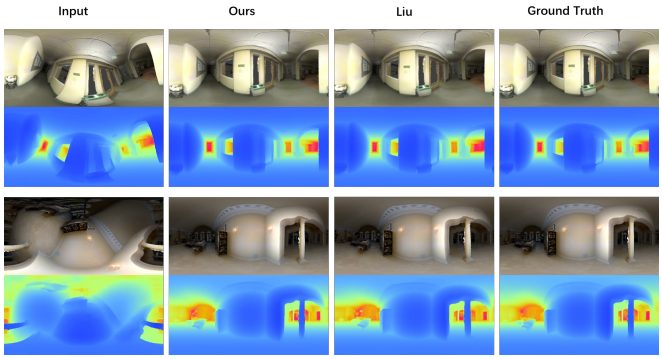

Figure 8. Depth estimation samples from 3D60

5 Conclusions

Compared to previous methods, We innovatively use a 3D CMMatrix to constrain the end-to-end upright adjustment network, and perform upsampling in three-dimensional space. Experimental results show that our method effectively improves the quality of generated images compared to previous approaches, and resolves the issue of cracks appearing in the generated upright images.

Acknowledgements

This work was supported in part by the Natural Science Foundation of Chongqing, China under Grant CSTB2024NSCQ-MSX0437.

References

[1] J. An, H. I. Koo, and N. I. Cho. Rectification of planar targets using line segments. *Machine Vision and Applications*, 28:91–100, 2017.

[2] J. Bai, H. Qin, S. Lai, J. Guo, and Y. Guo. Glpanodepth: Global-to-local panoramic depth estimation. *IEEE Transactions on Image Processing*, 2024.

[3] 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.

[4] H. Chen, S. Li, and J. Li. An end-to-end network for upright adjustment of panoramic images. *Procedia Computer Science*, 222:435–447, 2023.

[5] B. Davidson, M. S. Alvi, and J. F. Henriques. 360 camera alignment via segmentation. In *European Conference on Computer Vision*, pages 579–595. Springer, 2020.

[6] P. V. Gakne and K. O'Keefe. Monocular-based pose estimation using vanishing points for indoor image correction. In *2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–7. IEEE, 2017.

[7] A. C. Gallagher. Using vanishing points to correct camera rotation in images. In *The 2nd Canadian Conference on Computer and Robot Vision (CRV'05)*, pages 460–467. IEEE, 2005.

[8] S. Gao, K. Yang, H. Shi, K. Wang, and J. Bai. Review on panoramic imaging and its applications in scene understanding. *IEEE Transactions on Instrumentation and Measurement*, 71:1–34, 2022.

[9] M. A. Hassanien, V. K. Singh, D. Puig, and M. Abdel-Nasser. Predicting breast tumor malignancy using deep convnext radiomics and quality-based score pooling in ultrasound sequences. *Diagnostics*, 12(5):1053, 2022.

[10] J. Jeon, J. Jung, and S. Lee. Deep upright adjustment of 360 panoramas using multiple roll estimations. In *Computer Vision–ACCV 2018: 14th Asian Conference on Computer Vision, Perth, Australia, December 2–6, 2018, Revised Selected Papers, Part V 14*, pages 199–214. Springer, 2019.

[11] J. Jung, B. Kim, J.-Y. Lee, B. Kim, and S. Lee. Robust upright adjustment of 360 spherical panoramas. *The Visual Computer*, 33:737–747, 2017.

[12] R. Jung, A. S. J. Lee, A. Ashtari, and J.-C. Bazin. Deep360up: A deep learning-based approach for automatic vr image upright adjustment. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 1–8. IEEE, 2019.

[13] R. Jung, S. Cho, and J. Kwon. Upright adjustment with graph convolutional networks. In *2020 IEEE International Conference on Image Processing (ICIP)*, pages 1058–1062. IEEE, 2020.

[14] N. Kawai. A method for rectifying inclination of panoramic images. In *ACM SIGGRAPH 2019 Posters*, pages 1–2. 2019.

[15] H. Lee, E. Shechtman, J. Wang, and S. Lee. Automatic upright adjustment of photographs with robust camera calibration. *IEEE transactions on pattern analysis and machine intelligence*, 36(5):833–844, 2013.

[16] M. Lin, J. Wu, J. Meng, W. Wang, and J. Wu. Screening of retired batteries with gramian angular difference fields and convnext. *Engineering Applications of Artificial Intelligence*, 123:106397, 2023.

[17] Z. Ling, Z. Xing, X. Zhou, M. Cao, and G. Zhou. Panoswin: a pano-style swin transformer for panorama understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17755–17764, 2023.

[18] H. Liu, H. Li, H. Fu, R. Xiao, Y. Gao, Y. Hu, and J. Liu. Degradation-invariant enhancement of fundus images via pyramid constraint network. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 507–516. Springer, 2022.

[19] J. Liu, H. Chen, S. Li, and J. Li. Generation of upright panoramic image from non-upright panoramic image. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5261–5270, 2024.

[20] J. Liu, Y. Xu, S. Li, and J. Li. Estimating depth of monocular panoramic image with teacher-student model fusing equirectangular and spherical representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1262–1271, 2024.

[21] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11976–11986, 2022.

[22] X. Ma, Y. Zhou, H. Wang, C. Qin, B. Sun, C. Liu, and Y. Fu. Image as set of points. *arXiv preprint arXiv:2303.01494*, 2023.

[23] A. T. Martins, P. M. Aguiar, and M. A. Figueiredo. Orientation in manhattan: Equiprojective classes and sequential estimation. *IEEE transactions on pattern analysis and machine intelligence*, 27(5):822–827, 2005.

[24] S. Orhan and Y. Bastanlar. Semantic segmentation of outdoor panoramic images. *Signal, Image and Video Processing*, 16(3):643–650, 2022.

[25] J. Pan, W. Cui, X. An, X. Huang, H. Zhang, S. Zhang, R. Zhang, X. Li, W. Cheng, and Y. Hu. Mapsnet: Multi-level feature constraint and fusion network for change detection. *International Journal of Applied Earth Observation and Geoinformation*, 108:102676, 2022.

[26] Y. Shan, H. Chen, J. Zhang, S. Li, and J. Li. Multi-scale attention-based inclination angles estimation for panoramic camera. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1322–1330, 2024.

[27] J. Xiao, K. A. Ehinger, A. Oliva, and A. Torralba. Recognizing scene viewpoint using panoramic place representation. In *2012 IEEE conference on computer vision and pattern recognition*, pages 2695–2702. IEEE, 2012.

[28] Y. Xu, K. Wang, K. Yang, D. Sun, and J. Fu. Semantic segmentation of panoramic images using a synthetic dataset. In *Artificial Intelligence and Machine Learning in Defense Applications*, volume 11169, pages 90–104. SPIE, 2019.

[29] K. Yang, X. Hu, Y. Fang, K. Wang, and R. Stiefelhagen. Omnisupervised omnidirectional semantic segmentation. *IEEE Transactions on Intelligent Transportation Systems*, 23(2):1184–1199, 2020.

[30] Y. Yang, C. Zhang, Y. Zhao, and Y. Zhang. Panoramic depth estimation algorithm based on attention module. In *2022 3rd International Conference on Computer Vision, Image and Deep Learning & International Conference on Computer Engineering and Applications (CVIDL & ICCEA)*, pages 1232–1235. IEEE, 2022.

[31] C. Zhang, Z. Cui, C. Chen, S. Liu, B. Zeng, H. Bao, and Y. Zhang. Deeppanocontext: Panoramic 3d scene understanding with holistic scene context graph and relation-based optimization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12632–12641, 2021.

[32] Y. Zhang, S. Song, P. Tan, and J. Xiao. Panocontext: A whole-room 3d context model for panoramic scene understanding. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VI 13*, pages 668–686. Springer, 2014.

[33] 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)*, pages 448–465, 2018.