This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3119367, IEEE Access

Article

3D Reconstruction with Spherical Cameras

Yan Cui [1,2], Qingling Chang [1], Qiang Liu [2], Xin Yang [1], Yajiang Huang[2], Shichao Chen[2], Fei Ren[1], Didier stricker[3*]

[1]Faculty of Intelligent Manufacturing, Wuyi University, Jiangmen 529000 China;
[2]Zhuhai 4Dage Network Technology, Zhuhai 519000 China;
[3]Deutsches Forschungszentrum für Künstliche Intelligenz.
[*]Correspondence: didier.stricker@dfki.de

Abstract: The goal of image-based 3D reconstruction is to establish a high-quality 3D expression from images. In order to achieve a high-resolution and real-time 3D model, inspired by the open source COLMAP, we propose a novel framework (3DMAP) to reconstruct 3D scenes based on spherical cameras. Unlike traditional methods which focus on building a 3D plane by Poisson distribution function, our method illustrates the key processes of 3D reconstruction: locating the camera based on global feature, estimating the scene's relative depth from monocular panoramic images, and obtaining a high-quality 3D surface reconstruction. In the camera locating part, we use a global descriptor augmentation model to build a labeled panorama dataset **GDAP**, in which the images are captured by our designed spherical cameras; In the depth estimation part, we propose a new network UMDE that can estimate the depth of both indoor and outdoor scenes; Finally, in the 3D surface reconstruction section, we turn the reconstruction problem to a graph optimization problem, called GraphFit, in which, we optimize the point clouds with s-t graph and smoothing method successively. We conduct experiments on our own dataset to demonstrate the effectiveness of our proposed 3DMAP framework. Experimental results show that our 3DMAP has achieved good evaluation scores and visual effects.

Keywords: panoramic image dataset, depth estimation, 3D reconstruction, s-t graph, smoothing, spherical camera

I. INTRODUCTION

3D reconstruction is very important in computer vision and computer graphics. At present, due to its low cost and high precision, image-based 3D reconstruction technology is widely used in augmented reality (AR), autonomous vehicles, Unmanned Aerial Vehicle (UAV) and so on. So, researchers have made great progress in image-based 3D reconstruction and proposed the common solution "COLMAP", which contains the main steps: calculating the camera position with images, estimating the depth map of each image, generating surface reconstruction (SR) and getting mesh with texture. But COLMAP performs not well in scene reconstruction based on panoramic images. Analysis finds that the main challenges lie in the several aspects. First, existing feature extracting methods cannot deal with the wide FOV spherical to calculate camera pose well for lacking labeled panoramic image dataset. Second, the effective depth estimation networks cannot fit both in indoor and outdoor scene simultaneously. Finally, the popular surface reconstruction algorithms limited by the fitting function to balance the energy. To solve these problems and utilize the rich information of wide FOV panoramic images, we propose a new framework according to COLMAP logic, called "3DMAP". It mainly contains three parts, camera location, depth estimation and surface reconstruction.

Camera location (GDAP**).** The camera location is usually calculated with key points extracted from images. so, the accuracy of camera pose depends on the feature extracting methods. The traditional feature extracting algorithms, such as SIFT[1], SURF[2], AKAZE[3] are robust but slow due to the violent matching operation. It will be slower when processing the panoramic images, what's worse, it cannot deal with the distortion brought by the wide FOV. The convolutional neural networks (CNNs) based patch feature extraction algorithms[4-7] surpass those traditional ones. Han et al [4] proposed a unified method that combines the learning feature representation and the learning feature comparison function to train the patch matching system. Zagoruyko et al [5] learn directly from image data a general similarity function for patches that can implicitly take into account various types of transformations and effects. Tian et al [6] propose the L2-Net to learn high performance descriptor in Euclidean space via the CNN. GEORGESCU[7] et al proposed an approach that combines automatic features learned by CNN and handcrafted features computed by the bag-of-visual-words (BOVW) model. But most of them are supervised, which need costly labeled datasets, but the labeled dataset is scarce seriously, especially the panoramic image dataset. On the other hand, for the

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3119367, IEEE Access

panoramic images can provide rich information, it has attached the attention of researchers, and many vision tasks have turned to spherical images. To utilize the rich information of panoramic, we provide a Learning based Global Descriptor Augmentation Panoramic Image Dataset (GDAP). It contains more than 1,000,000 labeled panoramas of about 10,000 scenes including indoor and outdoor scenes, and the images are captured by our own designed spherical cameras[8].

Depth estimation (UMDE). In image-based 3D reconstruction, the image depth is usually estimated with the deep learning methods. Considering the depth maps estimated from monocular images are conform with regularity and more visually acceptable, we mainly research the monocular depth estimation in this paper. The previous learning based monocular depth estimation networks[9-13], Eigen et al.[9] propose the first paper about learning base monocular depth estimation and base this paper Laina et al. [10] first proposed a predicted network based on ResNet[11]. There papers are use in perspective. To predict the depth from panorama, Zioulis et al.[12] ,which use the ResNet [11] to predict the depth. Cao et al [13] propose to formulate depth estimation as a pixel classification task. But most of them are trained either on indoor or outdoor scenes due to the big difference between the two kinds of scenes. However, recently, the demand for reconstruction in hybrid scenes has grown rapidly. Moreover, as far as we know, there is no single model to fit these two kinds of scenes simultaneously. To meet the requirements, we propose a Unified Mono Depth Estimation (UMDE) network which is trained on our own dataset called Hybrid Scene Panoramic Image with Depth Map (HSPD), and the HSPD dataset contains both indoor and outdoor scenes.

3D surface reconstruction (GraphFit). Most of the traditional reconstruction algorithms[14,15] use the idea of regression. However, the effects of them are impacted by point clouds' quality. In order to improve the performance, some scholars transformed the reconstruction tasks to optimization tasks[16, 17]. Nan et al.[16] proposed an effective optimization algorithm called Polyfit, and it has achieved a great improvement, but it is difficult to balance its composite energy equation. Therefore, we propose a novel algorithm based on graph optimization to get a better consequence. It builds a s-t graph[18] and then smoothes the coarse 3D model to get the precision reconstruction result.

Our work makes the following contributions:

- We propose a new solution 3DMAP to reconstruct 3D model with spherical cameras. It consists three main sections, GDAP, a learning based global descriptor augmentation panoramic image dataset; UMDE, a unified depth estimation algorithm fit in both indoor and outdoor scenes; GraphFit, a surface reconstruction method transforming the reconstruction task to graph optimization problem.
- We propose a panoramic image labeled dataset based on spherical camera (GDAP). The dataset contains over 1,000,000 images of about 10,000 scenes and the labeled content includes key-points and their matching relationships extracted by the global augmentation feature descriptor model according to the SuperGlue[19]. Part of our data will be made available at https://github.com/4Dage-Tec/Dataset.
- We propose a Unified Mono Depth Estimation (UMDE) network that can be used for both indoor and outdoor scenes simultaneously. The network is trained on our own hybrid scene dataset (HSPD) with depth ground truth captured by LIDAR.
- We propose a 3D surface reconstruction algorithm (GraphFit) based on graph optimization. In the process, we propose an energy equation to establish the s-t graph, and then smooth the coarse 3D structure with our provided smoothing algorithm.

According to the framework 3DMAP proposed in this article, related works are reviewed in the following three aspects: Global Descriptor Augmentation, Monocular depth estimation, and 3D surface reconstruction.

2. RELATED WORKS

A. GLOBAL DESCRIPTOR AUGMENTATION

Generally, in the Structure From Motion (SFM)[20] system, the purpose of feature matching is to estimate camera's poses. Traditional algorithms like SIFT[1], SURF[2], AKAZE[3], ORB[21] extract key-points and calculate hand-crafted descriptors. The works[4-7]are also illustrated based on the same idea. Although these algorithms have a good performance in some fields, they cannot deal with the global context well. Additionally, in recent years they have been surpassed by some learning based local feature extraction algorithms[22]. To enhance the feature matching process, some proposed to merge global information into key-points descriptors [23, 24]. It has been proven that global augmentation improved the robustness of feature matching a lot. To merge global information into the feature descriptors, ContextDesc[23] integrates the geometric and visual context information into learning-based descriptors. Furthermore, SpuerGlue[19] integrates the neighbor key-points information from self-attention and cross-attention with a GNN network. Subsequently, LoFTR[25] uses a transformer with self-attention and cross-attention layers to process the dense local features extracted from the convolution backbone. Whereas the performance of SpuerGlue[19] is poor on panoramic images, especially when the distance between poses becomes large. In this paper, to overcome the limitation, we establish a dataset to

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3119367, IEEE Access

enforce our SFM system. We record the correspondences between key-points from panoramic images captured in two near locations in the dataset. With its help, we can get precise camera poses to reconstruct the accurate 3D model.

B. MONOCULAR DEPTH ESTIMATION

Monodepth estimation boomed in recent years because of CNN's success. The first CNN-based model to estimate depth from a single RGB image is Eigen et al.[9]. Later on, Laina et al. [11] first proposed a predicted network based on ResNet[10]. And then Zioulis et al.[13] also proposed a depth estimation network based on ResNet [10] called Omnidepth to predict the depth and provide a dataset with the depth of only indoor scene. However, most of the above methods are trained with pure CNNs. To improve the accuracy of depth estimation, [10-13, 26-29] integrate Conditional Random Fields (CRF) into a deep neural network. For example, Liu et al [26] pointed out that because depth estimation can naturally be expressed as a continuous conditional random field (CRF) learning problem. So they proposed a deep convolutional neural field model for estimating depth from a single image to explore the capabilities of deep CNN and continuous CRF. Wang et al [27] proposed a unified joint depth and semantic prediction framework. The consistency of the two tasks is learned through joint training and enforced at different stages of the whole framework to improve the performance of the two tasks. And use two layers of HCRF to strengthen the synergy between global and local predictions. Xu et al. [28] proposed a sequential network based on multi-scale CRFs to estimate depth. Xu et al [29] uses continuous CRF to fuse multi-scale information from different layers of the front-end CNN, it also designed a structured attention module which automatically adjusts the amount of information transferred between the corresponding features of different scales. Hu et al.[30]proposed a compound loss function and got clearer edges of the object. All of the above methods are based on regression without order. Fu et al.[31] proposed the spacing-increasing discretization to discretize depth and transform the depth estimation to an ordinal regression problem. Farooq et al.[32] followed Fu et al.[31] and proposed a module based on Vision Transformer(ViT) to discretize depth with the adaptive bins. Since all of them are supervised, it requires a large amount of expensive labeled data. Therefore, many scholars start to focus on unsupervised monocular depth estimation. A lot of excellent works were published, such as[33-37]. Godard et al. [33] proposed to use stereo pairs images to predict disparity based on the left-right consistency, but it requires a large number of paired images as training data. Zhou et al. [34] proposed a framework, which contains a pose network and a depth network to estimate ego-motion and depth from videos. Yin et al.[36] proposed a multi-task framework called GeoNet. It is a jointly unsupervised learning framework for monocular depth, optical flow and ego-motion estimation from videos. Studies proved that both of the above supervised and unsupervised methods could not suit the indoor and outdoor scenes at the same time with a big gap in between. Especially, for the large scene SFM, it is very important to estimate depth from a scene that composed of both indoor and outdoor parts. In this paper, we proposed a unified learning-based model to estimate the depth maps of hybrid scenes. As far as we know, we are the first to propose such a model.

C. 3D SURFACE RECONSTRUCTION

3D reconstruction from sampled point clouds has been a major problem in both computer vision and computer graphics. A large number of traditional surface reconstruction methods fit point clouds with Possion [14, 15]. Though Poisson has achieved good performance on object surface reconstruction, the effect on scenes reconstruction is not ideal for the high requirement of point clouds. In fact, most of these point clouds have shown to be noisy (outlier, missing, etc.), especially in indoor scene. In addition, indoor scenes contain many planes, and the surface fitted by the Poisson function is not smooth enough. Many scholars are concerned about how to obtain a better indoor scene reconstruction. They either supposed simple vertical walls [38-40] or the Manhattan world[39-41]. However, these assumptions bring in obvious limitations. In order to achieve a better reconstruction of indoor scenes, many works generated planes from point clouds first, and then reconstruct the indoor scenes with the planes[16, 17, 42, 43]. Mura et al.[43]proposed a pipeline to reconstruct general 3D interior architectures, significantly increasing the range of real-world architectures. Liu et al. proposed to use the FloorNet [26] to automatically reconstruct a floorplan with an RGB-D video. He *et al* [44] proposed DABNet, a deformable context feature pyramid module was designed in this network to improve the adaptive modeling capability of multiscale features. Furthermore, a boundary-weighted loss function was proposed to direct the network to focus on cloud boundary information and optimize the relevant detection results. DABNet can detect the complex clouds with high accuracy, not only a clearer boundary but also a lower false-alarm rate. Most of the above works reconstruct indoor scenes by regression, but Nan et al.[16] transforms the reconstruction task into a Mixed-Integer Programming (MIP) task, which opens up a new way of solving reconstruction tasks and achieves a good performance. However, there is a serious problem in [16]. It is difficult to find a balance point of each energy item for the complex energy equation. In the end, inspired by [18], we propose an algorithm to deal with the surface reconstruction from multi-view depth maps. The algorithm transforms the reconstruction task into a graph optimization task, and it includes three parts. It generates polygons and polyhedrons at first, and then it selects

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3119367, IEEE Access

polygons and polyhedrons. Finally, it smoothes the coarse structure with a smoothing algorithm.

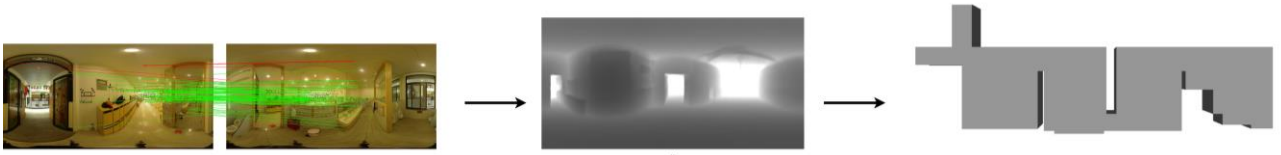


Figure 1. **The key processes of the reconstruction system.**

3. MATERIALS AND METHODS

This section consists of four parts, the overview and the three key aspects of the reconstruction system.

A. overview

According to the classic 3D reconstruction logic, such as COLMAP[45], we propose our own framework 3DMAP. The entire process COLMAP[45] contains the following steps, collecting image data, calculating the accurate camera position through the structure-from-motion (SFM)[46] algorithm, generating the depth map of each images through the multi-view-stereo (MVS)[47] algorithm, performing surface reconstruction(SR) to generate the 3D model. Finally, using texture mapping (TM) algorithm to get mesh with texture. Similar with COLMAP [45], our 3DMAP system composed with three key aspects. (Figure 1). Firstly, it extracts feature to locate the camera, which retraining the SuperGlue[47] model on our provided panoramic datasets (GDAP) to improve the accuracy of the SFM algorithm on panoramic images. Then, we use the unified learning based monocular depth estimation (UMDE) to meet the increasing the requirements of hybrid scenes' reconstruction in MVS algorithm. Finally, in GrapFit, we use the graph optimization 3D surface reconstruction algorithm to improves the appreciation of the 3D model. In this section, we mainly describers the three improved processes, that is, a learning based global descriptor augmentation panoramic image datasets, a unified learning based monocular depth estimation model and a surface 3D reconstruction algorithm.

B. A LEARNING BASED GLOBAL DESCRIPTOR AUGMENTATION PANORAMIC IMAGES DATASET

In the image-based 3D reconstruction process, camera locating is the basic step. Generally, it locates the camera according to the feature matching of images. But most existing feature extraction and matching methods perform poorly on panoramic images, mainly due to lacking the panoramic image datasets.

The performances of existing perspective image-based methods. The traditional patch feature extraction algorithms SIFT [1], AKAZE [3] use image pyramid to enhance their performance. The hand-crafted descriptor cannot meet the requirements of locating the camera well. Though the CNN based algorithms[36, 48, 49] use the learning descriptor, the descriptors are always local. So some works[19, 24] try to exploit the global geometric information with an attention mechanism. Those methods worked very well on perspective images except on panoramic images. In order to get the good performance of feature matching on panoramic images, we test the current state-of-the-art global method SuperGlue[19] on different FOV images, and the results are shown in Figure 2. We indicate the confidence of key-point matching with color. The colors from blue to red indicate a gradual increase in confidence, and the color bar is on the right side in Figure 2. From the figure, we can see that panoramic images contain more information than the three perspective images in total (FOV-30, FOV-45 or FOV-60), and the matching result contains many wrong matches compared with its perspective peers. There are mainly two reasons that caused the failure. One is the obvious differences between perspective and panoramic images for the feature's distribution. Two is the large distortion of panoramic images.

The GDAP datasets. Analysis finds, lacking labelled panoramic image datasets is one key factor that leads to the failure performance of existing methods. So, we propose a learning based global descriptor augmentation panoramic image dataset (GDAP). It contains over 1,000,000 high-resolution images from about 10,000 scenes covering room, corridor, street and more. These images were captured by our camera 4D Kankan pro[9] which has eight fish eyes lens calibrated with checkerboards, and four of them in up and the other four in bottom. The images are captured by the spherical camera [8], and the label information contains all key-points extracted from each panoramic image and an index of the corresponding key-points of the image pairs (which are usually captured in near location). Some of the GDAP data are made available at https://github.com/4Dage-Tec/Dataset.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3119367, IEEE Access

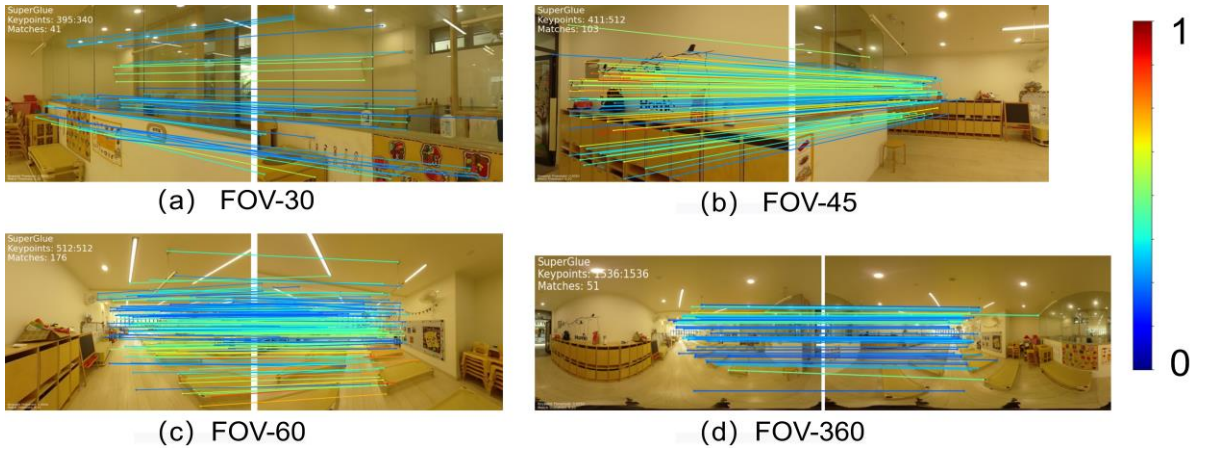


FIGURE 2. Comparison of the effects of SuperGlue[19] under different FOV. (a) FOV-30, the matches number is 41 of 395:340 keypoints. (b) FOV-45, the match number is 103 of 411:512 keypoints. (c) FOV-60, the matches number is 176 of 512:512 keypoints. (d) FOV-360, the matches number is 51 of 1536:1536 keypoints. The color bar indicates the matching confidence change.

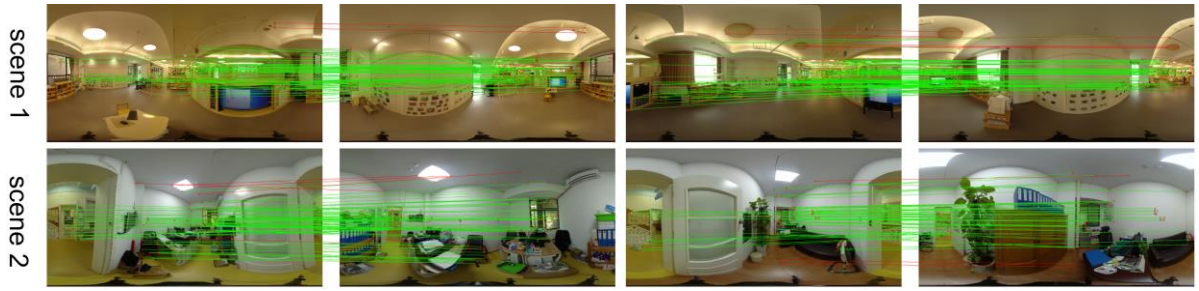


FIGURE 3. Above represents two examples of indoor scenes and the labeled information (SuperGlue with SFM). The green and red lines in the right group images represent the initial feature matches by SuperGlue. The green lines represent the inliers, the red lines represent the outliers (filtered with SFM system)

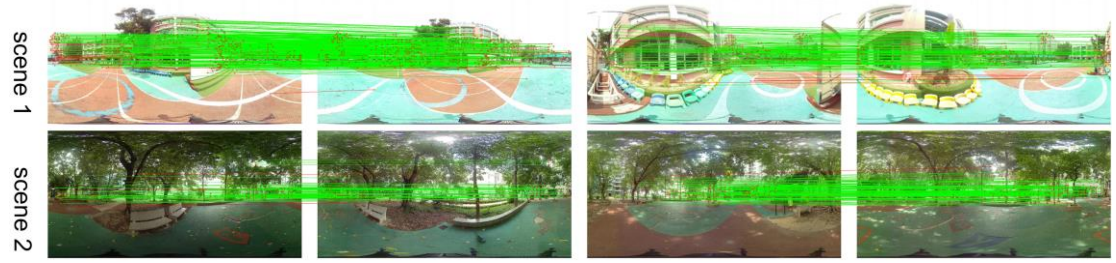


FIGURE 4. The two examples of outdoor scenes and the labeled information (same as Figure 3).

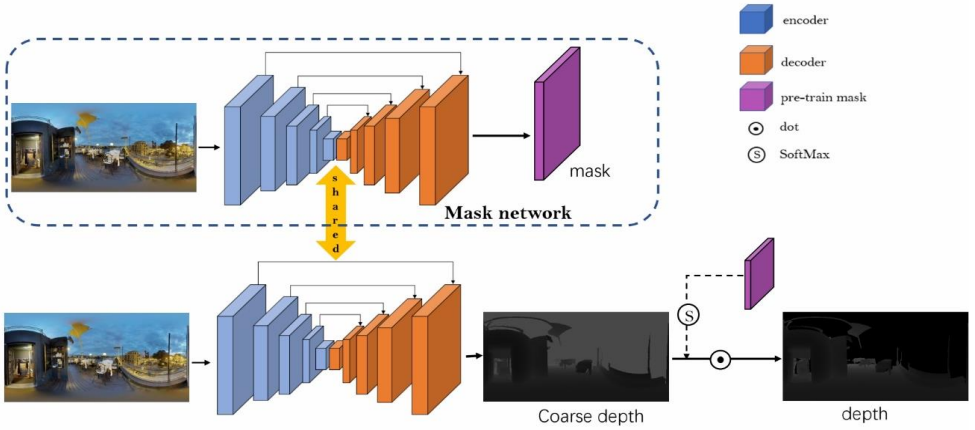


Figure 5. The architecture of our depth estimation framework. The Purple layer is our mask layer to remove the distant view.The backbone is ResUnet[13].The coarse depth map is without masked.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3119367, IEEE Access

Camera locating method based on panoramic images. In order to get more correct matches of panoramic images, we do the following four operations. Firstly, we split a panoramic image into six perspective images and use a pre-trained model (eg. SuperGlue[19]) to infer and get feature matching results. Then we reproject those matching results back into panoramic images. Secondly, with the help of our SFM system, we priorly get those cameras' location information. We feed those matching results into a SFM pipeline to get reliable sparse point clouds. And to improve the matching results' reliability, we only keep those feature matching results with more than four tracks (namely more than four cameras which can observe the feature). Thirdly, we train a global learning-based feature extraction and matching model on the dataset, as shown in Figure 3 (indoor) and Figure 4(outdoor). Finally, we make an inference with our trained model on our dataset to get more reliable corresponding key-points pairs and repeat the process.

C. Unified Mono Depth Estimation on Indoor and Outdoor Panoramic Scenes

The requirements of a unified depth estimation model. Depth estimation is an important step in image-based scene 3D reconstruction. The previous learning based depth estimation models are trained either on indoor [16] or outdoor scenes[13]] separately, due to the big difference between them, such as the light, the object distance, the background distance and etc. Moreover, in a mixed scene, the main prediction error comes from the error brought by the distant view. Therefore, the model will iterate to reduce the long-range error during training, thereby affecting the prediction accuracy of the close-range. As far as we know, there is no single model to suit these two kinds of scenes at the same time. However, recently, the demands for 3D reconstruction of hybrid scenes growing rapidly, and a unified model is necessary to our 3D reconstruction system. In addition, in our study, we found that the depth maps estimated from monocular images is more continuous and structured (its point clouds are more visually pleasing, which is very important to our later reconstruction step) than binocular images. The reason is that binocular image relies on disparity map deeply which drifted a lot with a small disturbance of transformation matrix between left and right (or up and down) cameras. On the other hand, the data was more difficult to collect. Though some excellent works focus on binocular depth estimation(360SD), they usually use the synthetic dataset to train and test their model, which is very difficult to transfer to real scenes. To this end, we propose a Unified Mono Depth Estimation (UMDE) network to mitigate the above problems.

The UMDE network. The aim of UMDE is to estimate the depth of outdoor or indoor scenes. For the indoor scenes, the common depth estimation network (eg. U-ResNet[13]) can predict the depth directly, but for the outdoor scenes, the depth map estimated by U-ResNet[13] (or other usual depth estimation network) contains distant views (like: sky, distant objects etc.) which affects the reconstruction. In order to weaken the impact, we introduce the mask network to filter the distant background. So, we design the UMDE network, and the network choose U-ResNet[13] as the backbone and uses the Binary Cross Entropy loss to train the network in our dataset, which includes the indoor and outdoor panoramic images captured with our spherical camera[8] and the ground truth contains the image depth captured by LIDAR and the semantic information (distance or close view). The network architecture is shown in Figure 5. It includes two subnets, a mask sub-network and a depth estimation sub-network. The two subnets share the U-ResNet[13] and they cooperate to predict the depth map of the input image. First, the mask sub-network trains a mask layer. Second, the depth estimation sub-network first predicts a coarse depth of the input image with the pre-trained network in the mask sub-network, and then it calculates the depth map of the image with the coarse depth and mask layer.

The mask sub-network. The mask sub-network is to remove the distant view of the coarse depth, and the essence of the mask network is a semantic segmentation network, and its main function is to segment the image into distant and non-distant parts. As described above, it uses U-ResNet[13] as backbone and uses Binary Cross Entropy loss to train the network on our own dataset. The loss function is defined as

$$Loss = -\left(y \cdot \log(\hat{y}) + (1-y) \cdot \log(1-\hat{y})\right) \quad (1)$$

Where y is the label, $\hat{y}$ is the predicted value.

The trained mask layer divides the image pixels into two categories, distant views and close views or objects. The pixel of the distant view is set to 0, and the others are set to 1.

The depth estimation sub-network. In the depth estimation sub-network, we first take the U-ResNet[13] pre-trained in mask sub-network to estimate the coarse depth. Then, we calculate the mask layer with the softmax function. Thus, the pixels of distant view will equate to 0, and the others will become the same probability. Finally, we do a dot product of the coarse depth map and the mask to filter those distance backgrounds. The loss function is defined as

$$L_{total} = \alpha L_{smooth} + \beta L_{normal} + \gamma L_{SSIM} \quad (2)$$

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3119367, IEEE Access

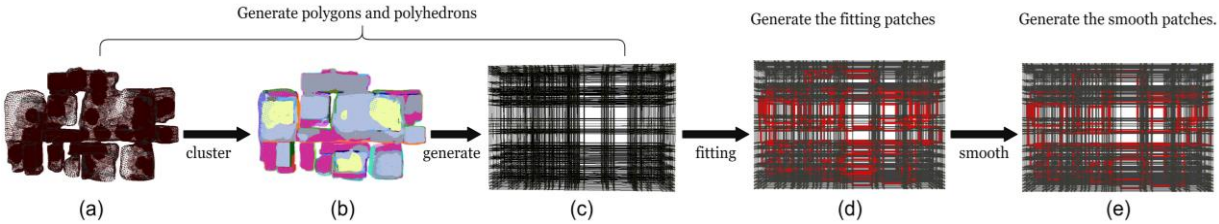


Figure 6. The five steps of reconstruction from point clouds. (a) Exterior point cloud of free space. (b) A segment of planar clusters. (c) Generate polygons and polyhedrons. (d)Result of min-cut. (e) Result of smoothing.

Where L_{smooth} is Smooth L1 loss[50] L_{SSIM} is SSIM loss[4], and in this paper we set α =0.8, β =0.1, =0.1. L_{normal} define as

$$l_{normal} = \frac{1}{n}\sum_{i=1}^{n}\left(1-\frac{\left\langle n_i^d, n_i^g \right\rangle}{\sqrt{\left\langle n_i^d, n_i^d \right\rangle}\sqrt{\left\langle n_i^g, n_i^g \right\rangle}}\right) \quad (3)$$

Where n_i^d define as eq 4 n_i^g define as eq 5:

$$n_i^d \equiv [-\nabla_x(d_i), -\nabla_y(d_i), 1]^T \quad (4)$$

$$n_i^g \equiv [-\nabla_x(g_i), -\nabla_y(g_i), 1]^T \quad (5)$$

Where d_i is the predict depth and g_i is the ground truth.

With the mask layer to remove the long-range view, not only the problem of the long-range prediction error can be solved, but also the network can iterate in the direction of improving the accuracy of the close-range prediction.

Additionally, we train the mask sub-network first because we find that if we train the two subnets together, it is hard to converge and easy to stop in local minimal point.

D. Cubic Reconstruction From AI-DepthMaps (GraphFit)

In this section, we propose a novel algorithm of 3D surface reconstruction from multi-view depth maps GraphFit. Most of the traditional methods treat reconstruction as a regression task, such as the Poisson function [15]. However, these methods require high precision point clouds. What's worse, the point clouds of the scene have too many missing points and noise. Thus, it is difficult to reconstruct the scene well with the traditional method. In this case, some researchers focus on transforming reconstruction problems into optimization problems to deal with this problem. For example, Nan et al.[15] and Coudron et al.[17]. Nan et al.[15] proposed PolyFit[15], which transforms the reconstruction task to an Integer-Programming task, and has achieved a great improvement in the smooth performance. However, it is difficult to balance the weight of each part in the compound energy equation composed of fitting, smoothing, and point coverage, because the three parts of the compound energy equation are mutually exclusive. As a result, we take the divide-and-rule strategy to avoid this problem. Based on this, we propose an algorithm GraphFit, consisting of three main parts shown in Figure 6.

GraphFit. It transforms the reconstruction task to graph optimization task, and the whole process contains three sections, generating polygons and polyhedrons, generating the fitting patches and generating the smooth patches.

1)Generating Polygons and Polyhedrons

Polygons and polyhedrons. In this part, we will generate polygons and polyhedrons from the multi-view depth maps estimated from the previous step. Firstly, we generate the free space from the depth maps following and obtain free space from point clouds following Kazhdan et al.[15]. (Figure 6-(a)), and extract exterior point clouds. Secondly, we cluster the point clouds. We have combined normal vectors, adjacency, and plane distribution to cluster point clouds based on the meansift[51] (Figure 6-(b)). Finally, we get the initial planes from point clouds [52] and generate polygons and pollyhedrons based on initial planes with[15] (Figure 6-(c)).

2)Generating the fitting patches

Given N polygons $P = \{p_i \mid 1 \le i \le N\}$ generated from the previous part, we will select a subset of these polygons that can optimally describe the geometry of the scene. To achieve this goal, we use the divide-and-rule idea to propose two energy equations to deal with fitting and smoothing separately.

Building graph. As we know that each two neighbor polyhedrons have only a common polygon. We regard each polyhedron as a vertex, and the polygons shared by adjacent polyhedrons as edges to build the s-t graph. In our s-t Graph, the initial s-vertices are those polyhedrons containing the camera, and the initial t-vertices are those not in free space. Furthermore, we propose an energy equation $E_{fitting}(x)$ to calculate the weight of each edge in s-t graph. $E_{fitting}(x)$ is to evaluate the fitting level of the polygon to the point cloud. After that, we transform the 3D model to an s-t graph. The energy equation $E_{fitting}(x)$ is defined as

$$E_{fitting}(x) = \min\left(\left\{ i \in N \middle| \sum_{j \in R(x,i)} \left(D_j - \left| \frac{w_x + n_x^T C_i}{n_x^T V_{ij}} \right| \right)^2 \right\}\right) \quad (6)$$

Where the x is a polygon, and N represents those cameras that can observe the plane containing the x and the D_j is the true depth value of pixel j. $R(x,i)$ is the rasterized pixel set of the projection of the x on the i th camera. C_i is

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3119367, IEEE Access

the position of the i th camera. w_x and n_x^T define the equation of the plane where x is locating. V_{ij} is the spatial direction of j pixel on the camera i. Then, we use the min-cut algorithm to find a [S, T], which is the smallest capacity like [14]. Thus, we get a high fitting surface structure of polygon set from min-cut. After that, we define a patch as a set of polygons that are adjacent and coplanar in our fitting surface structure and get an initial patch set. The result is shown in (Figure 6(d)), and the red line is our patch set.

3)Generating the smooth patches

Smooth model. The fitting energy equation (4) will make the generated model fit the point cloud as much as possible. However, the loss of point cloud and the appearance of noise will cause gaps and protrusions on the surface of the fitting model. To deal with this problem, a smooth energy equation is defined to smooth the fitting model operate on patch set given pre-step. The energy equation $E_{smooth}(n)$ is defined as

$$E_{smooth}(n) = n * R_0 \tag{7}$$

Where the n represents the current patch number, R_0 is the mean original smooth residual of each patch calculated from the min-cut. As we know, the smooth process will increase the fitting residuals. That is, they are inverse. In order to balance the fitting and smoothing residuals, we establish the relationship R_0 between them. The R_0 is defined as

$$R_0 = \frac{a \times \min(\sum_{x \in S_0} E_{fitting}(x) + b, 0)^2 + c}{n_0} \tag{8}$$

Where S_0 represents the original patches, n_0 is the number of S_0, the a, b, and c are super parameters. In this paper, for the sum $E_{fitting}(x)$ is between about 1000 and 20000 according to our study on our dataset, we set a= -0.0001, b=-10000, c=10000 in order to adaptively smooth. To smooth the 3D structure, we iterate the patches. We first select a patch randomly, and then divide the s-t Graph into multiple s-vertices closures and t-vertices closures along with the plane containing the selected patch. To decrease the count of patches, we cut some s-vertices closures down from current s-vertices, or merge some t-vertices closures back to current s-vertices. After that, we calculate the comprehensive energy of the big section and the whole model with formula (7) separately and then compare the value of the two parts' energy. If the energy of the big section is smaller than that of the whole model, we say that the iteration is successful. Then, we keep the big section left to iterate continuously until the comprehensive energy cannot diminish any further. And the comprehensive energy equation is defined as

$$E_{total}(S, n) = \sum_{x \in S} E_{fitting}(x) + E_{smooth}(n) \tag{9}$$

Where S is the patch set in the current model, and the n is the number of S. The smooth result is shown in (Figure 6-(e)), the red line is the smooth patches.

4. Experiments

In the experiment, we mainly test our dataset, depth estimation network, and 3D surface reconstruction. Besides these three parts, it describes the dataset used to estimate depth maps, the implementation details, and the ablation studies.

A. DATASETS

GADP. The dataset we generate contains over 1,000,000 high-resolution images from about 10,000 scenes covering room, corridor, street and etc. The images are captured by a spherical camera [8], and the label information contains all key-points extracted from each panoramic image and an index of the corresponding key-points of the image pairs (which are usually captured in near location).

HSPD. The dataset that we use to train our UMDE model contains 100,000 panorama images, labeled depth captured by the LIDAR equipment, and the semantic information (distance or close view). The obtained sparse depth map is projected back to RGB images. This dataset includes rich scenes and high-resolution images, while the ratio of the outdoor scenes and the indoor scenes is 1:9.

B. IMPLEMENTATION DETAILS

1)Training Configuration

We set the initial learning rate lr=0.0001, batch_size=6, use Adam as our optimizer and set $\alpha = 0.0001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, epsilon= 10e-8.

2)Environment

Our network is implemented by PyTorch-1.5.0 and train in 2080ti. The version of python is 3.6.9, and the operating system is Ubuntu 18.04 LTS.

C. Experimental Results and analyzations

This part mainly conducts comparative experiments from three aspects of feature matching based on our dataset, depth estimation, and surface reconstruction.

1) Feature matching and camera location with our Dataset(GADP)

The feature matching results. To prove the effectiveness of our dataset, we first contrast the self-attention and cross-attention of SuperGlue[19] and our model on the same panoramic image, as shown in Figure 7. Then, we compare the test results on the same panoramic images of SuperGlue[19] trained on perspective image dataset and our panoramic dataset separately, and the indoor scenes results are shown in Figure 8 (indoor) and the outdoor scenes results are shown in Figure 9 (outdoor), in which, green lines indicate the correct matches and the red lines indicate the wrong matches, the same as in Figure 3.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3119367, IEEE Access

Analyzations. Figure 7 shows the comparisons of SuperGlue[19] and our model tested on the same panoramic image, and the statistic feature matching results can be seen in Table 1, from which we find that base on the same number key-points, the matches number of our model is more than SuperGlue. Thus, we learn that SuperGlue[19] pre-trained on perspective images focuses on a small area of the whole panorama, which decreases its global information. While the model trained on our dataset can span its attention to the whole image making it more robust.

Furthermore, Figure 8 and Figure 9 show the comparisons of different trained dataset SuperGlue[19] tested on the same images of indoor and outdoor scenes. The inliers are statistic in Table 2. From Figure 8, Figure 9 and Table 2, we can see the inliers of our panoramic dataset trained model are much more than trained on perspective image dataset neither in indoor scenes nor in outdoor scenes. Additionally, in Figure 8, the rotation of the image pairs in the scene 1 is large, and the transform of the image pairs in the scene 2 is distant. The results prove that for the strict situation (with sharp change), our model trained on our panoramic dataset is also effectiveness. The Figure 9 is similar with the Figure 8.

TABLE 1. the feature key-points and matches numbers

Attentions	Models	Key-points Num	Matches Num
Self-attention	SuperGlue	1536:1536	51
	Our model	1536:1536	187
Cross-attention	SuperGlue	1536:1536	51
	Our model	1536:1536	187

TABLE 2. The inliers of feature matching

models / inliers / scenes		Trained on perspective dataset	Trained on our panoramic dataset
Indoor scenes	Scene 1	21/26	100/120
	Scene 2	161/169	434/443
Outdoor scenes	Scene 1	40/52	280/298
	Scene 2	136/162	325/353

The camera location results. We train the SuperGlue[19] on our panoramic dataset, combining SFM to get the high performance, like the more correct feature matches and inliers on panoramic image. With the correct feature matches, we can locate the camera accurately. Figure 10(indoor) and Figure 11(outdoor) are the results of the camera pose located through the matched feature. Specifically speaking, Figure 10 displays the camera locations of the image in scene 2 of Figure 8, and Figure 11 displays the camera locations of the image in scene 2 of Figure 9.

Analyzations. The important target of providing panoramic dataset is to locate the spherical camera pose. From Figure 10 and Figure 11, we can count the location results in Table 3 of the two scenes in Figure 8 and Figure 9. And the results tell us that there are overlaps among camera poses located with model trained on perspective image dataset. In contrast, the camera poses located with model trained on our dataset is similar with the ground truth. The overlap will seriously impact the following 3D reconstruction or lead to the 3D reconstruction almost impossible. The correct camera location can prove the value of our GADP dataset furthermore.

TABLE 3. The camera locations of two scenes location num is mean the number of location point that not overlapping

scenes / location num / models	Scene 2 in Figure 8	Scene 2 in Figure 9
Ground truth	11	10
SuperGlue trained on perspective image dataset	5	3
SuperGlue trained on our panoramic image dataset	11	10



FIGURE 7. Comparison of the self-attention and cross-attention of SuperGlue[19] and our model on the same panoramic image. With the same key-points 1536:1536, (a) self-attention and (b) cross-attention both get the matches 51 and 187 separately. The color bar indicates the attention weight.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3119367, IEEE Access

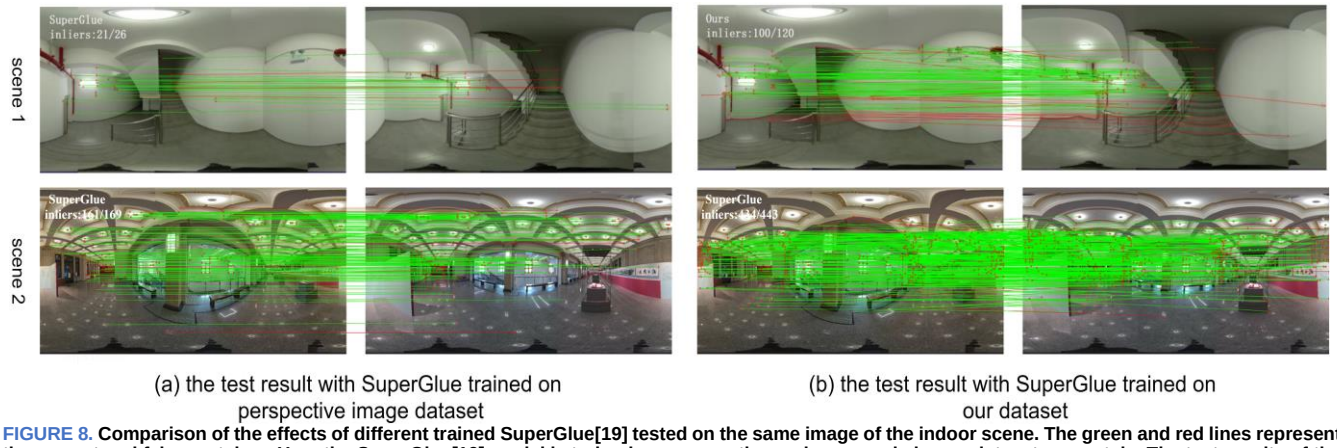


FIGURE 8. **Comparison of the effects of different trained SuperGlue[19] tested on the same image of the indoor scene. The green and red lines represent the correct and false matches. Here the SuperGlue[19] model is trained on perspective and panoramic image dataset separately. The test results of the two models: (a) is trained on perspective images, the inliers are 21/26 in the up large rotation image pairs, and 161/169 in the down long distance transform image pairs. (b) is trained on our panoramic dataset and the correspondence inliers are 100/120 and 434/443 respectively.**

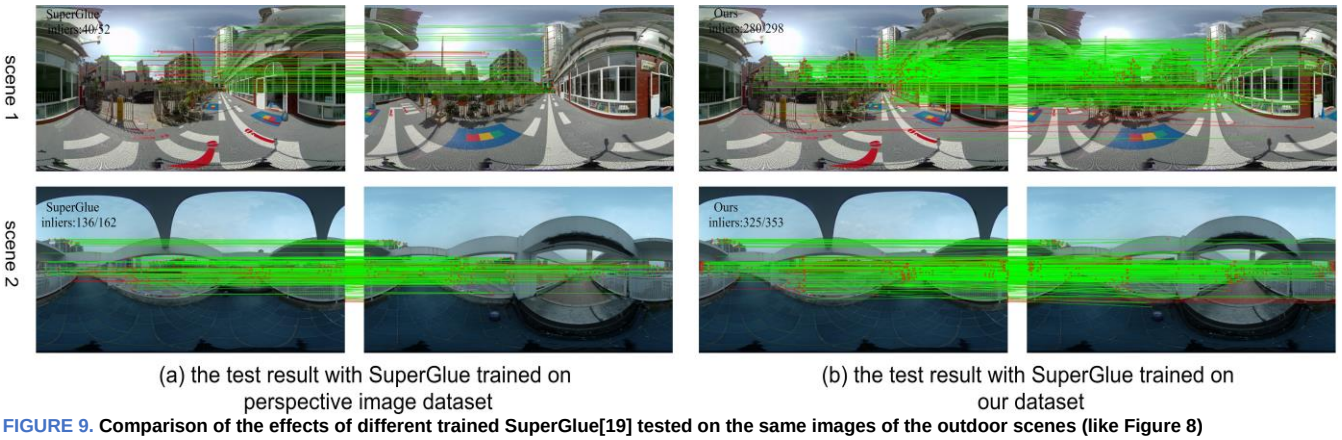


FIGURE 9. **Comparison of the effects of different trained SuperGlue[19] tested on the same images of the outdoor scenes (like Figure 8)**

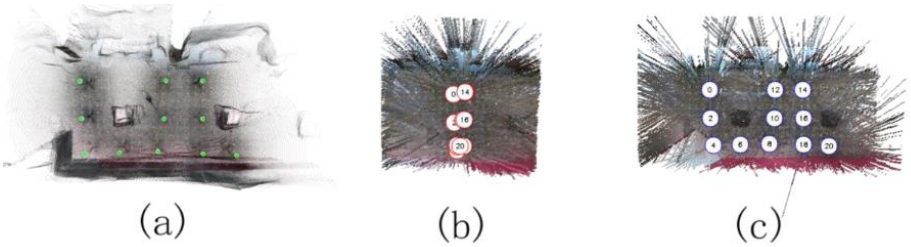


FIGURE 10. **The indoor scene's Camera loction with our dataset GADP of scene 2 in Figure 8. (a) Groudtruth with AI-depth map using ICP[53](green spots are Groudtruth pose of Panoramic images). (b)SFM[20] with SuperGlue pre-trained on perspective dataset (C)SFM[20] with SuperGlue pre-trained on our dataset.**

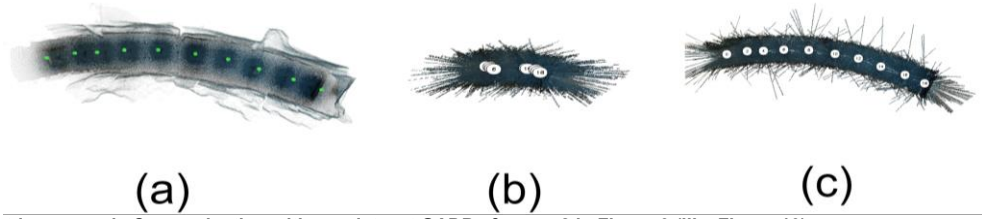


FIGURE 11. **The outdoor scene's Camera loction with our dataset GADP of scene 2 in Figure 9 (like Figure 10)**

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3119367, IEEE Access

2)UMDE Depth Estimation Results and analyzations

The depth estimation results. In order to test the performance of our depth estimation network, we compare our model with Omnidepth [13] in four groups of indoor scenes and outdoor scenes separately. And the results are shown in Figure 12 (indoor) and Figure 13 (outdoor). To evaluate the predicted depth, we select the Mean Absolute Error(MAE), the Absolute relative difference (Abs_Rel), the accuracy with threshold DELTA, proposed in [9], and the time cost to process a picture (Time(s)) as the evaluation indexes, the experiment results are shown in Table 4.

Analyzations. Figure 12 shows that our method predicts more detail structure preserved in ground truth. While the result predicted by Omnidepth losses the detail information. Furthermore, in the mixed scenes, our model removes the distant view which impacts the reconstruction, but Omnidepth does not remove it. In Figure 13, we find that our method removes the background precisely and retains the complete building information in outdoor scenes. On the contrary, the Omnidepth[13] estimates the wrong background (the building information is seriously lost due to the background is not removed, and the building cannot be distinguished from the background).

As far as we know, there is almost no other panorama dataset containing indoor and outdoor scenes simultaneously. Although Matterport3D have indoor and outdoor scenes, they removed the distant view of the image handcraft. Though, we have tested our model and Omnidepth on Matterport3D without training again, the results are unideal.

The intermediate outputs of depth estimation. In order to illustrate our depth estimation model in detail, we also give the intermediate outputs of our unified model in Figure 14.

For our depth estimation model consists of two subnets, the two sub-networks predict coarse depth and semantic segmentation result parallelly, and then an elementwise product between the coarse depth and the semantic segmentation (or mask layer) was applied to get the final depth of input image. So, we choose the coarse depth (or the depth without mask) and the mask semantic segmentation as the intermediate outputs. In order to display the differences between the intermediate outputs and the final depth, we show the ground truth and the final depth at the same time.

Analyzations. The advantage of Our UMDE model is that it can both deal with the indoor and outdoor scenes. From the comparation results of Figure 12 and Figure 13, we know the advantage mainly to the outdoor scenes. So, we just display the intermediate outputs of outdoor cases. From Figure 14, we can see the depth predicted without mask layer not only contains the near views, but also includes the distant views, like the sky, which will impact the following 3D reconstruction, so we should remove the distant views in coarse depth. Look again the mask semantic segmentation, in which the pixel values of distant views are just o (black). According to the process of the UMDE, with the product operation of coarse depth and mask layer, we can get the depth removed the distant views impacting the 3D reconstruction seriously of outdoor scenes, which ensures the effects of the 3D reconstruction of large scenes. So, we can derive that the performance of UMDE depth estimation is high.

3D point clouds results. In order to prove our depth estimation method (removing the distant view of depth maps without any pre-processing) is efficient, we design another experiment to project the depth map to 3d point clouds. We choose four difference scenes including indoor and outdoor scenes. The results are shown in Figure 15.

Analyzations. From Figure 15, we find in the outdoor scene, Omnidepth[13] did not remove the distant view, and in these point clouds, there are wrapped object over the scene, and it severely impacts the reconstructed model. On the contrary, the point clouds projected by the depth predicted with our model and the ground truth have removed the distant view, and the reconstruct models are very nice. From the results of Omnidepth tested on different dataset (indoor dataset, outdoor dataset, mix dataset), we can learn, for the indoor scenes, our model can produce comparative results like Omnidepth. And for the outdoor scenes, our model is much better than Omnidepth. Obviously, it is because our mask removes some difficult things for depth estimation (like trees and other small outdoor things) from the result. Though it is a bit unfair to compare on outdoor scenes, our goal is not to recover as much as depth information from RGB image but to get as much as useful information to recover a visually pleasing model. As for mix dataset, the Omnidepth model did not perform better than ours. And from the point cloud results, we know that Omnidepth model cannot predict distant things very well. To keep continuous, it always tries to warp outdoor scene as a whole, which is not what we expected for our latter pipeline. According to the results shown in Figure 15, we find that the 3D point clouds projected from the depth maps estimated by our UMDE model are well, proving the UMDE's high performance furthermore.

The failure cases of UMDE. To more comprehensively illustrate our model, we provide some failure cases in Figure 16. It contains three scenes, from indoor and outdoor scenes including a nearly successful case, a failure case and a totally failure case.

Analyzations. In fact, it was not a totally failure case on case 1. Our model predicted the mask successfully. While when it removed distant views from the image, it removed more than we wanted. As we can see, it also removed half of the building from the result (the right building). The reason is that though we got many outdoor data, it was not that much enough to train a very clear edge on our dataset. We need to add more annotated outdoor images in our dataset. The case 2 was predicted using our model on Matterport3D dataset. As we can see, our model did not predict mask very well on

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3119367, IEEE Access

this image. The reason is that our dataset is actually captured by real panoramic cameras, so were our training dataset. While the data in Matterport3D we got in the same way as Unifuse[54] told in their paper, was more or less rendered images and lost a lot of distant information in the RGB image. The case 3 was a totally failure case. It removed more than we wanted. The building on the right was totally removed in the result. Except for dataset reason, we will try other methods such as (Laplacian pyramid fusion) to avoid the failures in future.

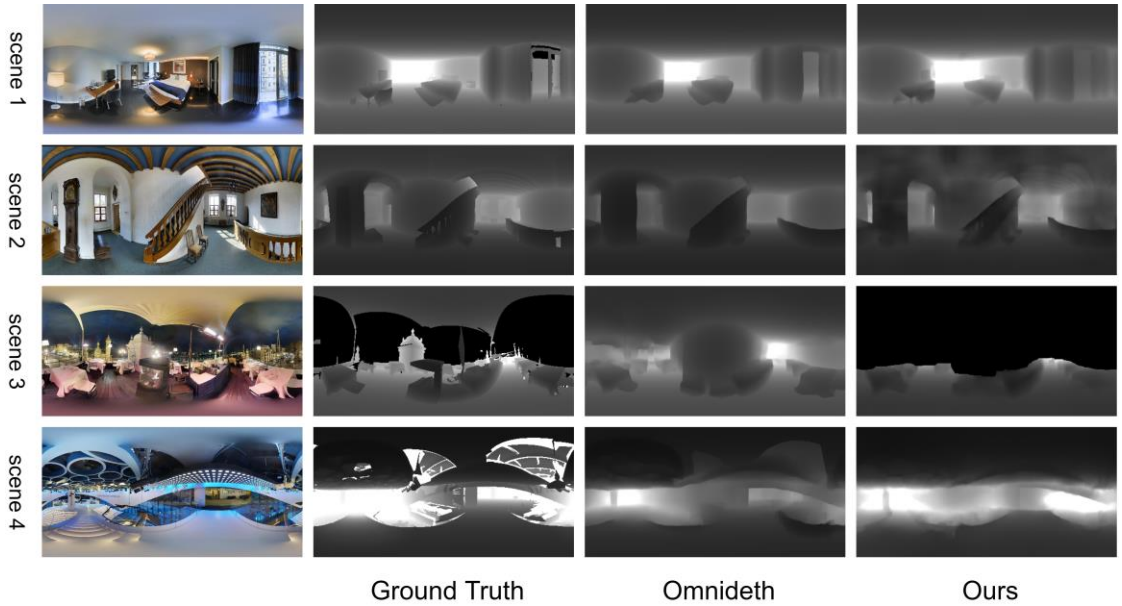


FIGURE 12. **Qualitative results of UMDE tested on our own dataset(indoor). From left to right are input Panoramic images, ground truth, Omnidepth [13], and our method in sequence.**

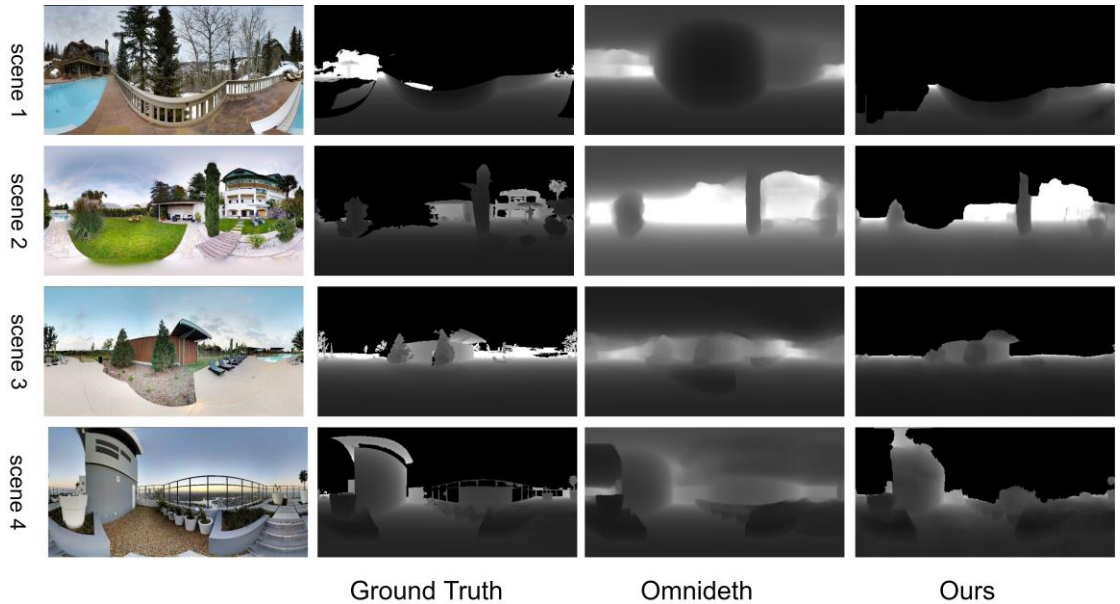


FIGURE 13. **Qualitative results of UMDE were tested on our own dataset(outdoor). From left to right are panoramic input images, ground truth, Omnidepth [13], and our sequence method.**



FIGURE 14. **The intermediate outputs of our strategy in detail. From left to right are the input, ground truth, the initial prediction, the mask of semantic segmentation, and the final estimation result.**

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3119367, IEEE Access

TABLE 4. Evaluation results of depth estimation on the HSPD test set. We divide the test set into two parts, which are indoor and outdoor sets. The indoor (or outdoor) means the model test in indoor (or outdoor) part, and the mix means the model tested with the entire test set. The shown values of the evaluated methods are the two networks trained and tested in our dataset.

Scene	Model	ABS_REL	MAE	DELTA1	DELTA2	DELTA3	TIMES
Indoor	OmniDepth	0.173	0.461	0.809	0.945	0.981	0.201
	Ours	0.115	0.263	0.875	0.972	0.987	0.199
Outdoor	OmniDepth	0.286	0.577	0.739	0.886	0.938	0.201
	Ours	0.113	0.438	0.835	0.935	0.953	0.199
MIX	OmniDepth	0.230	0.519	0.774	0.915	0.960	0.201
	Ours	0.114	0.356	0.854	0.953	0.970	0.199

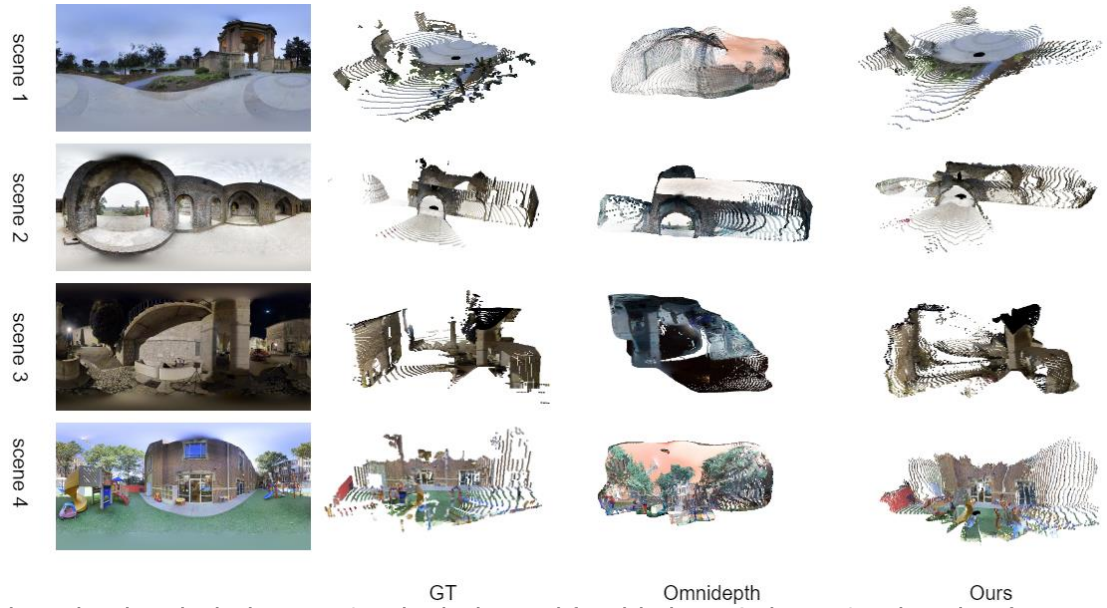


FIGURE 15. The result projects the depth maps as 3D point clouds. From left to right: input RGB images, 3D point projects from ground-truth depth maps, results of Omnidepth [13], and our method, respectively.

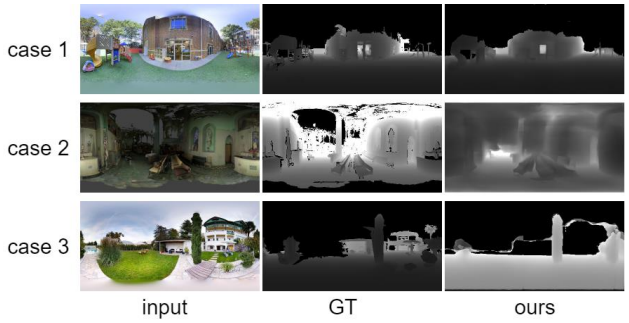


FIGURE 16. Some failure cases with our model. From left to right are input, ground truth and ours

3)GraphFit (3D reconstruction) Results and analyzations

In order to evaluate the performance of our 3D surface reconstruction algorithm, we compare the model with other state of the art models, provide some failure cases and analyze the time cost.

Comparations with other models and analyzations. We compare the model with Poisson [14] and Polyfit [16]. The results are shown in Figure 17. We select four scenes and make four group tests. The first group is generated by Poisson[14]. Basically, the surfaces of these models are rough and undulating. Although the reconstruction planes of the models generated by Polyfit [16] are flat, they lack detailed information compared with the parts in the red circles. Additionally, there is a prominent part in Poisson[14], and Polyfit[16] did not reconstruct this part, but our model reconstructs it clearly. Furthermore, our reconstruction results truthfully reflect the presence of bumps in this area. The second group also proves the same conclusion. In the third group, we can see that the area in the red circle is not flat and convex in Poisson[15], while the area in Polyfit [16] is smooth, but it lost some information. In contrast, the area is flat and clear in our model.

Especially, the fourth group is the aerial view of a two-story indoor scene. From the red circle, we can clearly see that the model reconstructed by our method has a more detailed structure than the Polyfit's one[16]. What's worse, it takes a long time to run the Polyfit[16] on the fourth scene (the two-story indoor scene).

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3119367, IEEE Access

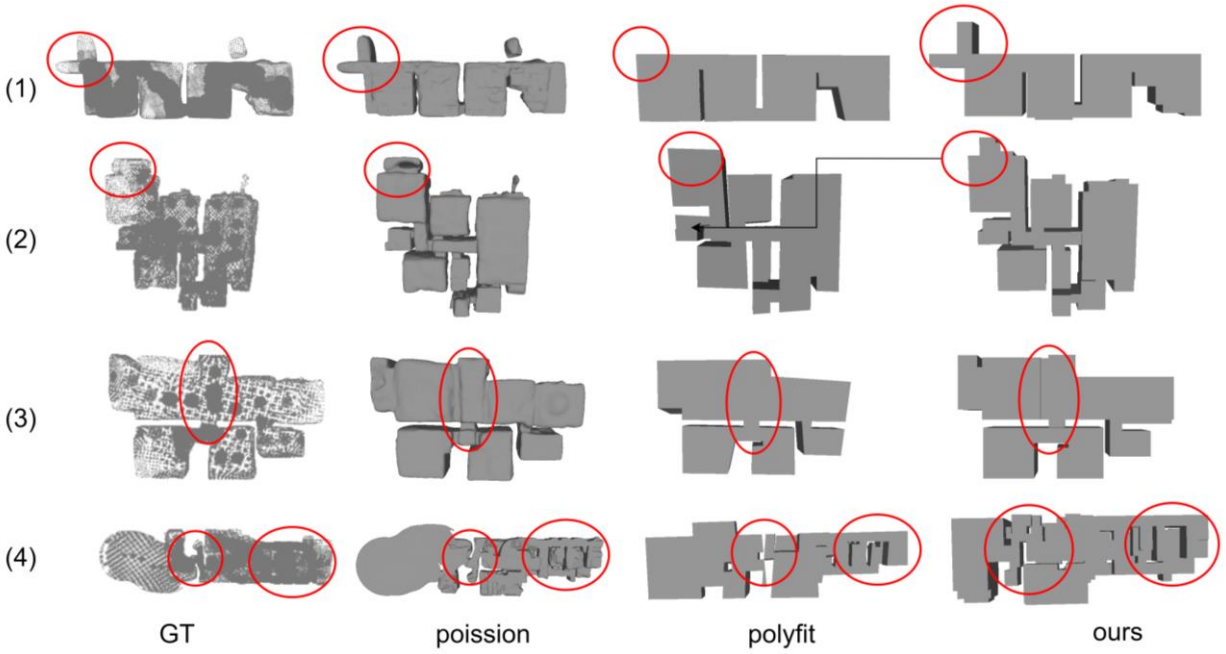


FIGURE 17. Qualitative results on our dataset. From top to bottom is a comparison of the effects in 4 different scenarios from left to right, ground truth(point cloud)results of [14], results of [16], and our method, respectively.

The failure cases and analyzations. In order to better analyze the performance of our model, among the plenty of experiments, we find the open scenes, the curve plane and the miss situation may fail somehow, so we selected the three type scenes that failed to rebuild with our model for analysis. These failure cases are shown in Figure 18. From the reconstruction results in the first row, we can see that our algorithm and Poisson[14] only reconstruct the ground in the outdoor scene. The reason is that the essence of our algorithm is to find the boundary of free-space. In the open outdoor scenes, the boundary of the camera's field of view is infinite, so the wrong walls and ceilings may be generated. It can be seen from the red circle area in the second row that our model does not reflect the effect of surface curvature compared with the result of Poisson reconstruction. In the third row, the red circle areas show that our model is compared with the result of Poisson reconstruction, and our reconstruction result shows the loss of objects. The reason why the surface cannot be reconstructed and the object is lost is that Polyfit[16] preprocessed clustering algorithm is difficult to get ground-truth. The input and our algorithm uses the same clustering algorithm as Polyfit[16], so these problems occur.

The Complexity analysis. In order to further study the performance of our algorithm, we compare the time complexity of our algorithm with Polyfit[16] and Poisson[14]. The results show in Figure 19. It can be seen from Figure 19 that our algorithm is faster than Polyfit[16], but slower than Poisson[14]. In theory, the clustering part depends on the number of point clouds. The complexity of our algorithm is $O(n^2)$. The generated graph part depends on the number of polygons generated by the clustering. The complexity is O(n). Smooth Part, depending on the number of polygon patches, the complexity is $O(n^2)$. Compared with Poisson[14] which use matching cube, we use a deterministic model, which has low requirements for point cloud accuracy and good regression results. Compared with the binary optimization used by Polyfit[16] (relying on third-party optimization libraries, the optimization time and effect are difficult to control).The efficiency of our algorithm is more stable.

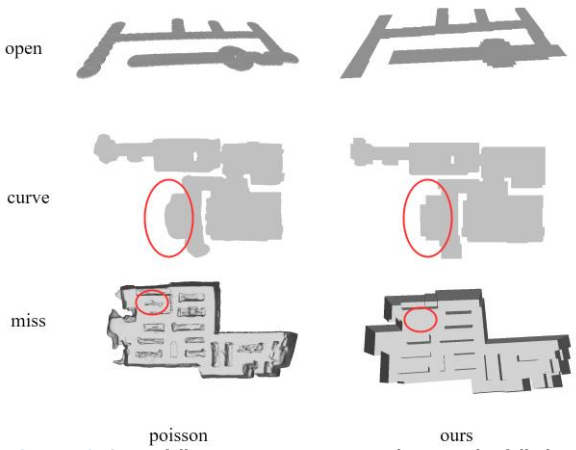


FIGURE 18. Some failure cases. From top to bottom, the failed case in outdoor, the failed case in curved walls failed, and the failed case in reconstruction of some objects. From left to right are poisson[14] and ours

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3119367, IEEE Access

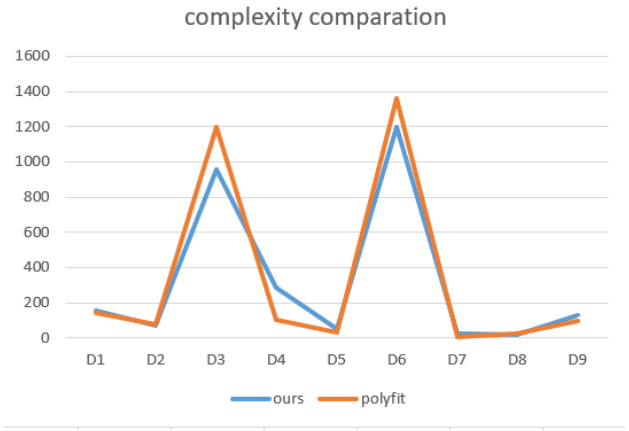


FIGURE 19. The comparison of time complexity about ours and others algorithm. The ordinate is the time corresponding to the reconstruction scene, and the abscissa is the different scene.

Table 5: Statistics on the examples presented in Figure 15. Points cloud is meaning the original number of points in the clouds. Original patches refer to the patch number before smoothing. Smoothed patches indicate the patch number after smoothing. The original energy is the energy of the model before smoothing, and smoothed energy is the energy of the model after smoothing. (include fitting energy and smoothing energy)

Scenes	Points	Planes	Original patches	Smoothed patches	Original energy	Smoothed energy
scene1	223K	82	555	124	10983	6514
scene2	140K	53	258	44	16004	9257
scene3	442 K	115	1117	188	12065	6794

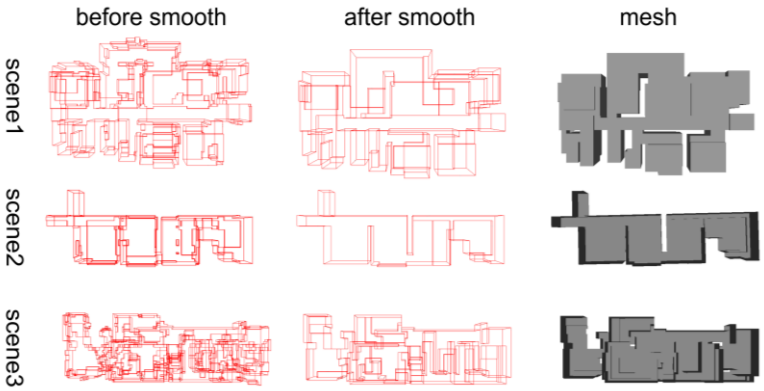


FIGURE 20 The results compared with smoothing and without smoothing. From left to right: Wireframe before smoothing, wireframe after smoothing, final model, respectively.

D. ABLATION STUDIES

In order to further analyze our method, we design an ablation test to verify the effectiveness of our smoothing algorithm.

In this part, we compare two sets of experiments with and without the smooth algorithm. Through Figure 20 and Table 5, we can clearly see that our optimization algorithm not only greatly smoothes the wireframe, but also keeps the overall structure of the model. According to Table 5, it can be seen that our smoothing algorithm can reduce the number of patches by 80% on average. With our smoothing algorithm, we can increase the smoothness by four times. Furthermore, the three groups' results show that the total model's energy including fitting energy and smooth energy reduces about 40% after smooth. A set uses our complete model, and another set only uses our s-t graph to build the model without smooth. The compared results are shown in Figure 20 and Table 5.

CONCLUSIONS

We propose an efficient resolution 3DMAP to reconstruct scenes with a spherical camera. It includes a labeled panoramic image dataset (GADP) to locate the camera pose, a unified monocular depth estimation framework (UMDE), and a novel reconstruction algorithm (GraphFit) based on graph optimization. GADP is the first large labeled panorama dataset. It contains rich label information, and it will make an important contribution to computer vision, especially for the 3D reconstruction. UMDE is the first unified framework to deal with the indoor and outdoor scenes simultaneously for making the indoor reconstruction more natural. Finally, GraphFit transforms the surface reconstruction to graph optimization. Likewise, it reduces

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3119367, IEEE Access

complexity and makes a simple solution possible for difficult problems. Besides the detail description of the proposed methods, we also provide supplements of experiments and analyzations to evaluate the performance of our resolution of 3D reconstruction based on spherical cameras. Overall, the set of resolutions integrate the traditional and deep learning methods to get the ideal effectiveness. In order to further improve the effect of reconstruction, our future works will focus on the following:

- Enrich our datasets and improve the accuracy of labeling.
- Improve the accuracy of depth estimation in outdoor scenes based on our proposed method.
- Use a learnable method to determine the threshold of our smooth algorithm.
- Explore more possible fields of application based our dataset and algorithm.

ACKNOWLEDGEMENT: We thank Shenzhen Xiao 'an Technology Co., Ltd (http://www.xiaofang-yun.com.cn/#/) for the 3D house scans and annotations.

REFERENCES

1. Lowe, D. G., Distinctive image features from scale-invariant keypoints. *International journal of computer vision* **2004,** 60, (2), 91-110.
2. Bay, H.; Ess, A.; Tuytelaars, T.; Van Gool, L., Speeded-up robust features (SURF). *Computer vision and image understanding* **2008,** 110, (3), 346-359.
3. Alcantarilla, P. F.; Solutions, T., Fast explicit diffusion for accelerated features in nonlinear scale spaces. *IEEE Trans. Patt. Anal. Mach. Intell* **2011,** 34, (7), 1281-1298.
4. Han, X.; Leung, T.; Jia, Y.; Sukthankar, R.; Berg, A. C. In *Matchnet: Unifying feature and metric learning for patch-based matching*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015; 2015; pp 3279-3286.
5. Zagoruyko, S.; Komodakis, N. In *Learning to compare image patches via convolutional neural networks*, Proceedings of the IEEE conference on computer vision and pattern recognition, 2015; 2015; pp 4353-4361.
6. Georgescu, M.-I.; Ionescu, R. T.; Popescu, M., Local learning with deep and handcrafted features for facial expression recognition. *IEEE Access* **2019,** 7, 64827-64836.
7. Tian, Y.; Fan, B.; Wu, F. In *L2-net: Deep learning of discriminative patch descriptor in euclidean space*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017; 2017; pp 661-669.
8. Cui, Y., Method and system for visual localization based on dual dome cameras. In Google Patents: 2021.
9. Eigen, D.; Puhrsch, C.; Fergus, R., Depth map prediction from a single image using a multi-scale deep network. *arXiv preprint arXiv:1406.2283* **2014**.
10. He, K.; Zhang, X.; Ren, S.; Sun, J. In *Deep residual learning for image recognition*, Proceedings of the IEEE conference on computer vision and pattern recognition, 2016; 2016; pp 770-778.
11. Laina, I.; Rupprecht, C.; Belagiannis, V.; Tombari, F.; Navab, N. In *Deeper depth prediction with fully convolutional residual networks*, 2016 Fourth international conference on 3D vision (3DV), 2016; IEEE: 2016; pp 239-248.
12. Cao, Y.; Wu, Z.; Shen, C., Estimating depth from monocular images as classification using deep fully convolutional residual networks. *IEEE Transactions on Circuits and Systems for Video Technology* **2017,** 28, (11), 3174-3182.
13. Zioulis, N.; Karakottas, A.; Zarpalas, D.; Daras, P. In *Omnidepth: Dense depth estimation for indoors spherical panoramas*, Proceedings of the European Conference on Computer Vision (ECCV), 2018; 2018; pp 448-465.
14. Kazhdan, M.; Bolitho, M.; Hoppe, H. In *Poisson surface reconstruction*, Proceedings of the fourth Eurographics symposium on Geometry processing, 2006; 2006.
15. Kazhdan, M.; Hoppe, H., Screened poisson surface reconstruction. *ACM Transactions on Graphics (ToG)* **2013,** 32, (3), 1-13.
16. Nan, L.; Wonka, P. In *Polyfit: Polygonal surface reconstruction from point clouds*, Proceedings of the IEEE International Conference on Computer Vision, 2017; 2017; pp 2353-2361.
17. Coudron, I.; Puttemans, S.; Goedemé, T. In *Polygonal reconstruction of building interiors from cluttered pointclouds*, Proceedings of the

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3119367, IEEE Access

European Conference on Computer Vision (ECCV) Workshops, 2018; 2018; pp 0-0.

18. Jancosek, M.; Pajdla, T., Exploiting visibility information in surface reconstruction to preserve weakly supported surfaces. *International scholarly research notices* **2014,** 2014.
19. Sarlin, P.-E.; DeTone, D.; Malisiewicz, T.; Rabinovich, A. In *Superglue: Learning feature matching with graph neural networks*, Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020; 2020; pp 4938-4947.
20. Pagani, A.; Stricker, D. In *Structure from motion using full spherical panoramic cameras*, 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops), 2011; IEEE: 2011; pp 375-382.
21. Rublee, E.; Rabaud, V.; Konolige, K.; Bradski, G. In *ORB: An efficient alternative to SIFT or SURF*, 2011 International conference on computer vision, 2011; Ieee: 2011; pp 2564-2571.
22. Luo, Z.; Shen, T.; Zhou, L.; Zhu, S.; Zhang, R.; Yao, Y.; Fang, T.; Quan, L. In *Geodesc: Learning local descriptors by integrating geometry constraints*, Proceedings of the European conference on computer vision (ECCV), 2018; 2018; pp 168-183.
23. Luo, Z.; Shen, T.; Zhou, L.; Zhang, J.; Yao, Y.; Li, S.; Fang, T.; Quan, L. In *Contextdesc: Local descriptor augmentation with cross-modality context*, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019; 2019; pp 2527-2536.
24. Zhang, J.; Sun, D.; Luo, Z.; Yao, A.; Zhou, L.; Shen, T.; Chen, Y.; Quan, L.; Liao, H. In *Learning two-view correspondences and geometry using order-aware network*, Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019; 2019; pp 5845-5854.
25. Sun, J.; Shen, Z.; Wang, Y.; Bao, H.; Zhou, X., LoFTR: Detector-Free Local Feature Matching with Transformers. *arXiv preprint arXiv:2104.00680* **2021**.
26. Liu, F.; Shen, C.; Lin, G. In *Deep convolutional neural fields for depth estimation from a single image*, Proceedings of the IEEE conference on computer vision and pattern recognition, 2015; 2015; pp 5162-5170.
27. Wang, P.; Shen, X.; Lin, Z.; Cohen, S.; Price, B.; Yuille, A. L. In *Towards unified depth and semantic prediction from a single image*, Proceedings of the IEEE conference on computer vision and pattern recognition, 2015; 2015; pp 2800-2809.
28. Xu, D.; Ricci, E.; Ouyang, W.; Wang, X.; Sebe, N. In *Multi-scale continuous crfs as sequential deep networks for monocular depth estimation*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017; 2017; pp 5354-5362.
29. Xu, D.; Wang, W.; Tang, H.; Liu, H.; Sebe, N.; Ricci, E. In *Structured attention guided convolutional neural fields for monocular depth estimation*, Proceedings of the IEEE conference on computer vision and pattern recognition, 2018; 2018; pp 3917-3925.
30. Hu, J.; Ozay, M.; Zhang, Y.; Okatani, T. In *Revisiting single image depth estimation: Toward higher resolution maps with accurate object boundaries*, 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), 2019; IEEE: 2019; pp 1043-1051.
31. Fu, H.; Gong, M.; Wang, C.; Batmanghelich, K.; Tao, D. In *Deep ordinal regression network for monocular depth estimation*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018; 2018; pp 2002-2011.
32. Bhat, S. F.; Alhashim, I.; Wonka, P., AdaBins: Depth Estimation using Adaptive Bins. *arXiv preprint arXiv:2011.14141* **2020**.
33. Godard, C.; Mac Aodha, O.; Brostow, G. J. In *Unsupervised monocular depth estimation with left-right consistency*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017; 2017; pp 270-279.
34. Zhou, T.; Brown, M.; Snavely, N.; Lowe, D. G. In *Unsupervised learning of depth and ego-motion from video*, Proceedings of the IEEE conference on computer vision and pattern recognition, 2017; 2017; pp 1851-1858.
35. Yang, Z.; Wang, P.; Xu, W.; Zhao, L.; Nevatia, R. In *Unsupervised learning of geometry from videos with edge-aware depth-normal consistency*,

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3119367, IEEE Access

Proceedings of the AAAI Conference on Artificial Intelligence, 2018; 2018.

36. Yin, Z.; Shi, J. In *Geonet: Unsupervised learning of dense depth, optical flow and camera pose*, Proceedings of the IEEE conference on computer vision and pattern recognition, 2018; 2018; pp 1983-1992.
37. Zhan, H.; Garg, R.; Weerasekera, C. S.; Li, K.; Agarwal, H.; Reid, I. In *Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018; 2018; pp 340-349.
38. Turner, E.; Zakhor, A. In *Floor plan generation and room labeling of indoor environments from laser range data*, 2014 international conference on computer graphics theory and applications (GRAPP), 2014; IEEE: 2014; pp 1-12.
39. Oesau, S.; Lafarge, F.; Alliez, P., Indoor scene reconstruction using feature sensitive primitive extraction and graph-cut. *ISPRS journal of photogrammetry and remote sensing* **2014,** 90, 68-82.
40. Ochmann, S.; Vock, R.; Wessel, R.; Klein, R., Automatic reconstruction of parametric building models from indoor point clouds. *Computers & Graphics* **2016,** 54, 94-103.
41. Turner, E.; Zakhor, A. In *Watertight as-built architectural floor plans generated from laser range data*, 2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization & Transmission, 2012; IEEE: 2012; pp 316-323.
42. Dolgov, S.; Khoromskij, B. N.; Litvinenko, A.; Matthies, H. G., Polynomial chaos expansion of random coefficients and the solution of stochastic partial differential equations in the tensor train format. *SIAM/ASA Journal on Uncertainty Quantification* **2015,** 3, (1), 1109-1135.
43. Mura, C.; Mattausch, O.; Pajarola, R. In *Piecewise - planar reconstruction of multi - room interiors with arbitrary wall arrangements*, Computer Graphics Forum, 2016; Wiley Online Library: 2016; pp 179-188.
44. He, Q.; Sun, X.; Yan, Z.; Fu, K., DABNet: Deformable contextual and boundary-weighted network for cloud detection in remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing* **2021**.
45. COLMAP. https://github.com/colmap/colmap
46. Schonberger, J. L.; Frahm, J.-M. In *Structure-from-motion revisited*, Proceedings of the IEEE conference on computer vision and pattern recognition, 2016; 2016; pp 4104-4113.
47. Schönberger, J. L.; Zheng, E.; Frahm, J.-M.; Pollefeys, M. In *Pixelwise view selection for unstructured multi-view stereo*, European Conference on Computer Vision, 2016; Springer: 2016; pp 501-518.
48. Dusmanu, M.; Rocco, I.; Pajdla, T.; Pollefeys, M.; Sivic, J.; Torii, A.; Sattler, T. In *D2-net: A trainable cnn for joint description and detection of local features*, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019; 2019; pp 8092-8101.
49. Tian, Y.; Yu, X.; Fan, B.; Wu, F.; Heijnen, H.; Balntas, V. In *Sosnet: Second order similarity regularization for local descriptor learning*, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019; 2019; pp 11016-11025.
50. Girshick, R. In *Fast r-cnn*, Proceedings of the IEEE international conference on computer vision, 2015; 2015; pp 1440-1448.
51. Comaniciu, D.; Meer, P., Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on pattern analysis and machine intelligence* **2002,** 24, (5), 603-619.
52. Fischler, M. A.; Bolles, R. C., Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM* **1981,** 24, (6), 381-395.
53. Low, K.-L., Linear least-squares optimization for point-to-plane icp surface registration. *Chapel Hill, University of North Carolina* **2004,** 4, (10), 1-3.
54. Jiang, H.; Sheng, Z.; Zhu, S.; Dong, Z.; Huang, R., Unifuse: Unidirectional fusion for 360 panorama depth estimation. *IEEE Robotics and Automation Letters* **2021,** 6, (2), 1519-1526.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/