Unsupervised Learning of Depth and Ego-Motion from Cylindrical Panoramic Video

Alisha Sharma
Laboratories for Computational Physics & Fluid Dynamics
Naval Research Laboratory
Washington, D.C. USA

Jonathan Ventura
Department of Computer Science & Software Engineering
California Polytechnic State University
San Luis Obispo, CA USA
jventu09@calpoly.edu

***Abstract*—We introduce a convolutional neural network model for unsupervised learning of depth and ego-motion from cylindrical panoramic video. Panoramic depth estimation is an important technology for applications such as virtual reality, 3d modeling, and autonomous robotic navigation. In contrast to previous approaches for applying convolutional neural networks to panoramic imagery, we use the cylindrical panoramic projection which allows for the use of the traditional CNN layers such as convolutional filters and max pooling without modification. Our evaluation of synthetic and real data shows that unsupervised learning of depth and ego-motion on cylindrical panoramic images can produce high-quality depth maps and that an increased field-of-view improves ego-motion estimation accuracy. We also introduce Headcam, a novel dataset of panoramic video collected from a helmet-mounted camera while biking in an urban setting.**

***Keywords*-computer vision, structure-from-motion, unsupervised learning, panoramic video**

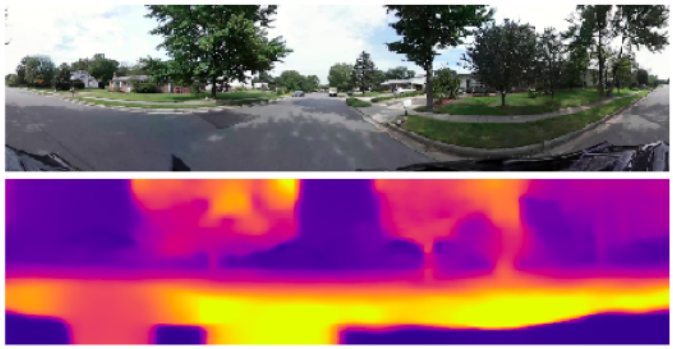


Fig. 1: An example of predicting depth (bottom) from a single panoramic input (top) using our proposed method. The depth is color-coded so that brighter pixels are closer.

I. Introduction

Understanding the structure of a 3D scene is an important problem in many fields, from autonomous vehicle navigation to free-viewpoint rendering of virtual reality (VR) content. The ability to automatically infer scene depth in panoramic video would be especially useful for free-viewpoint rendering in a VR headset [1], for example.

Given a color image, the scene depth is unknown and must either be inferred from single-view or multi-view cues or acquired with a different sensor. Single-image inference is especially interesting since most consumer content is captured from a single viewpoint without special hardware for depth estimation such as time-of-flight ranging or structured light sensors. Unfortunately, predicting 3D structure from a single image is extremely challenging. The number of confounding factors (e.g. varied texture, lighting, occlusions, and object movement) makes it an ill-posed problem: a single image could represent many possible 3D scenes.

Early attempts at estimating scene structure from motion (also known as SfM) focused on directly analyzing factors such as the geometry and flow of the image [2]–[4]. However, these models were often fragile in the face of occlusions, object motion, and other inconsistent, but real-world, conditions. In the past several years, many exciting advances have been made in estimating scene structure and ego-motion—motion of the observer—using deep neural networks.

Early research relied on labeled data for training [5], [6]. Unfortunately, labeled 3D footage is expensive to create, limiting the quantity and diversity of available training data. This limitation has triggered a promising new area of research: unsupervised SfM models, which figure out scene attributes such as depth and ego-motion without requiring labeled data. In the past few years, several unsupervised models have been proposed with comparable performance to the supervised state-of-the-art [7]–[11], lowering the cost and expanding the diversity of potential training datasets.

A. Panoramic Projection Models

While much progress has been made for pinhole perspective images, researchers have only recently started to apply these deep networks to panoramic input. There are many compelling applications for computer vision with non-pinhole projection images, such as robotic vision with omnidirectional cameras.

To process panoramic imagery in a convolutional neural network (CNN), we need to choose a panoramic projection model that maps spherical coordinates to image coordinates. The three most common options are spherical or equirectangular projection, cube map projection, and cylindrical projection.

Spherical projection has the advantage of representing the entire sphere in a rectangular image. The disadvantage is the distortions at the poles caused by the projection (see Figure 2). Because of these distortions, properly processing spherical

panoramic input in a CNN requiring expensive modifications to the model layers [12], [13].

Cube map projection also represents the entire sphere and avoids distortions at the poles; however, it introduces discontinuities between the faces of the cube. To use cube maps as input to CNNs, we need to run the model on each face separately and use a careful padding strategy and loss functions to encourage agreement between the six outputs [14], [15].

Cylindrical projection has been relatively less explored for CNN input. Unlike the other projection models, cylindrical projection is continuous and avoids increasing distortion toward the poles (Figure 2). The trade-off is that the cylinder cannot represent the entire sphere; the top and bottom are cut off. Despite that small disadvantage, we argue that cylindrical panoramas are ideal for use in CNNs they allow for standard convolutional layers and only require a simple horizontal wrap padding. Furthermore, in most applications, the top and bottom areas are relatively unimportant (usually consisting of sky or ceiling at the top and ground, vehicle, or camera mount at the bottom).

Despite the advantages of cylindrical projection, to our knowledge, deep networks for depth prediction have not yet been applied to cylindrical panoramic imagery. In this paper, we address this gap by proposing and evaluating a novel unsupervised learning approach that estimates depth and ego-motion from cylindrical panoramas. We achieve this by modifying the architecture of Zhou et al. [8], with improvements from later papers, to use cylindrical panoramic projection.

This work has three major contributions:

1) We present CylindricalSfMLearner, an unsupervised model for estimating structure from motion given cylindrical panoramic input.
2) We evaluate our method on synthetic and real data to validate our approach.
3) We provide a new dataset of panoramic street-level videos suitable for unsupervised learning of depth and ego-motion.

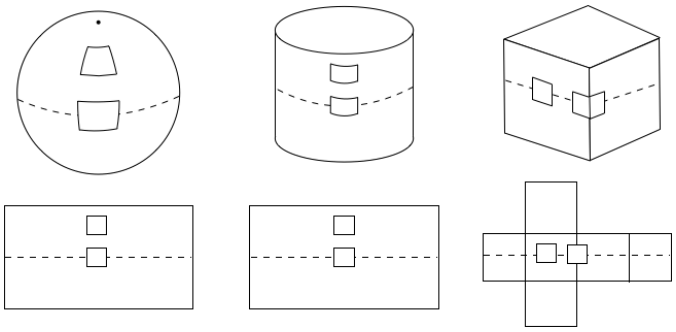


Fig. 2: Comparison of convolutional filtering using equirectangular projection (left), cylindrical projection (middle), and cube map projection (right). A square filter on the equirectangular projection (bottom left) maps to differently sized areas on the sphere (top left) [12]. In contrast, a square filter on the cylindrical projection (bottom middle) always projects to the same area on the cylinder (top middle). This property of cylindrical panoramas allows us to apply convolutional neural network layers to a cylindrical panorama without needing to model position-dependent effects on the receptive field, as in previous works [16]–[18]. Cube map projection (right) introduces seams and discontinuities, necessitating the use of multi-image inference and careful padding strategies [14], [15].

II. Related Work and Background

A. Supervised Monocular Depth Prediction

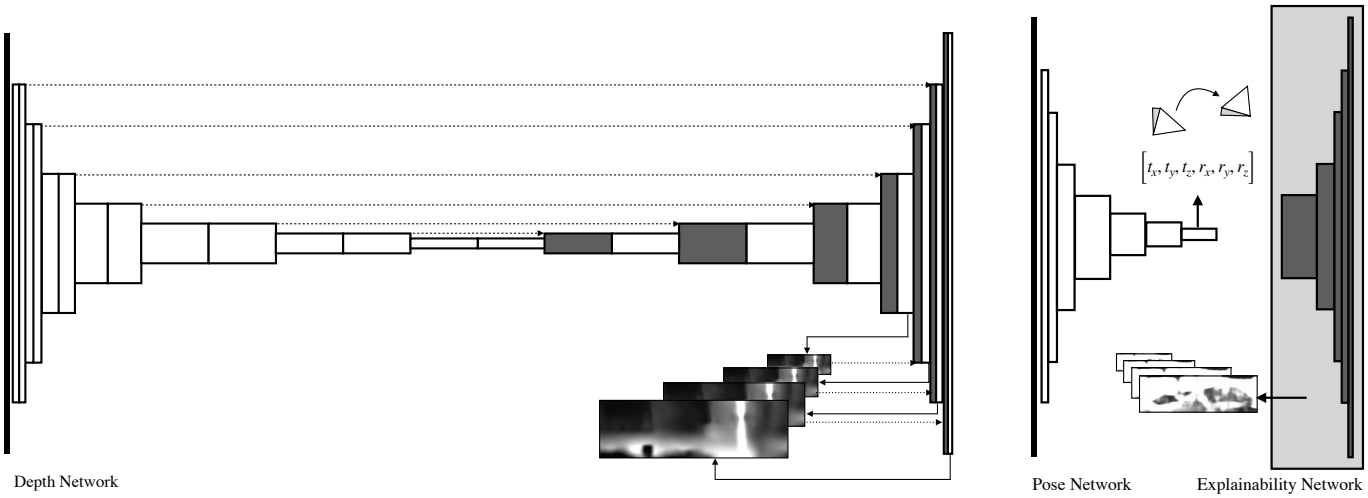
Early research focused on detecting structure from *stereo*—or multi-source—imagery. Stereo SfM is much more constrained than detecting structure from *monocular*—single-source—input, but the stereo input requirement limits the model's flexibility. Eigen [5] proposed a different approach using deep neural networks. They presented a supervised model for estimating depth maps from monocular input images. Their model was composed of two stacks—one for coarse estimation and one for fine estimation—and joined the two predictions.

B. Supervised to Unsupervised Models

While supervised models for single-image depth prediction demonstrate excellent performance, collecting labeled footage is very expensive, increasing training cost and limiting the size and diversity of datasets. This limitation triggered some researchers to turn towards unsupervised models. Godard et al., taking inspiration from previous stereo techniques, proposed a model that was trained on unlabeled stereo footage. Their trained model outperformed the previous supervised state-of-the-art on urban scenes and performed reasonably well on unrelated datasets [7].

Zhou et al. removed the constraint of needing stereo training footage [8]. They proposed an unsupervised model composed of jointly-trained depth and pose CNNs using a loss function tied to novel view synthesis. They found that their unsupervised model performed comparably to supervised models on the known datasets and reasonably well when tested against a completely unknown data set. Unfortunately, while the model could be trained on monocular footage, it assumes a given camera calibration, which prevents arbitrary footage from the web from being used as training data.

In a concurrent study, Vijayanarasimhan et al. addressed this shortcoming by explicitly modeling scene geometry [9]. Inspired by geometrically-constrained Simultaneous Localization and Mapping (SLAM) models and Godard's work on left-right consistency, they proposed a model capable of detecting both ego-motion and object motion—as well as depth and object segmentation—from uncalibrated monocular images. Building upon those previous works, Mahjourian et al. proposed a completely unsupervised model with explicit geometric scene modeling [10]. Their model introduced a new 3D loss function and added a new principled mask for handling unexplainable input.



(a) The depth network consists of a seven-layer encoder followed by a seven-layer decoder with a skip-layer architecture. The network returns the multi-scale disparity predictions, which can then be converted to depth predictions.

(b) The pose network consists of a five-layer encoder followed by two pose layers and returns the pose as a six-element vector. The explainability network adds a five-layer decoder and returns the multi-scale explainability mask.

Fig. 3: The SfMLearner model [8] consists of two jointly-trained CNN stacks. The left diagram shows the depth CNN, and the right diagram shows the pose/explainability CNN.

C. Beyond Pinhole Projection

All of the models previously discussed take pinhole images as input. However, pinhole images have a serious disadvantage: objects can move out of the field of view. Many applications, such as robotic navigation or virtual reality, benefit from the 360° field of view. Previous research tackled this omnidirectional SfM problem using direct methods with some success [19], but comparatively little research has been done with 360° imagery and CNNs.

Spherical input is particularly challenging for CNNs because spheres cannot be perfectly represented by a rectangular grid. In equirectangular projection, a common spherical projection method, this results in significant distortion in the polar regions of the image that propagate error through the convolutional layers. The traditional solution to this has been to use additional parameters and data augmentation to correct the distortions [20], but recently, researchers have presented several CNNs designed for direct spherical input. Several of these approaches aim for full rotational invariance by using signal processing techniques to model spherical convolutional layers [12], [13], [21]. While this approach is effective, it is also expensive; these models are severely limited in their input data size, making them impractical for most problems.

As full rotational invariance is often not required, several researchers have suggested a lighter-weight alternative: replacing normal convolutional filters with distortion-aware filters [16]–[18]. In this approach, the standard rectangular CNN filter is replaced by a filter that samples points based on the image distortion, correcting the polar distortion effects. These models can be trained with one projection model and tested with another, allowing them to utilize the large body of pinhole datasets. There were several other approaches aside from full rotational invariance and distortion-aware filters, including graph CNNs [22], style transfer [23], and increased filter sizes in the polar regions [24].

To avoid the distortions in the equirectangular projection, Cheng et al. [14] and Wang et al. [11] use cube maps as input to CNNs for saliency prediction and unsupervised depth and ego-motion. They run a network on each cube face separately and use padding and specialized loss functions to encourage coherence between the outputs.

Cylindrical projection is another way of achieving 360° views around a given axis. Unlike spherical panoramas, cylindrical panoramas do not capture the full 3D space. However, they have a major benefit: they can be mapped exactly to a single plane, removing the issues of polar distortion and discontinuities. Despite this benefit, little research has been done using deep networks with the cylindrical projection model [25], [26]. Furthermore, while some spherical CNN networks have made simplifications, including cropping of the polar regions [16] and distortion-aware filters [16], to the best of our knowledge, no previous work has applied a cylindrical CNN to the structure-from-motion problem. Our work introduces the first CNN model designed to predict depth and pose from cylindrical panoramic input.

III. Methods

In this work, we present an unsupervised convolutional model that jointly estimates the depth map from a single

cylindrical panoramic image and ego-motion from a short image sequence.

A. Model Architecture

Our architecture is based on that of Zhou et. al [8], an unsupervised model designed to predict depth and ego-motion in monocular pinhole images. The architecture, illustrated in Figure 3, is a convolutional network consisting of two jointly-trained stacks: (a) a depth network to estimate the depth map, (b) a pose network/explainability mask to estimate the change in the pose in image sequences and handle unexplainable input. The depth network follows the DispNet [27] skip-layer architecture, with seven contracting layers and seven expanding layers, outputting a multi-scale depth prediction. The pose network (PoseExpNet) consists of five contracting convolutional layers and three pose layers, outputting the predicted translation and rotation between the source and target views. The explainability network consists of a final five upconvolution layers and returns a multi-scale explainability mask, which masks "unexplainable" motion.

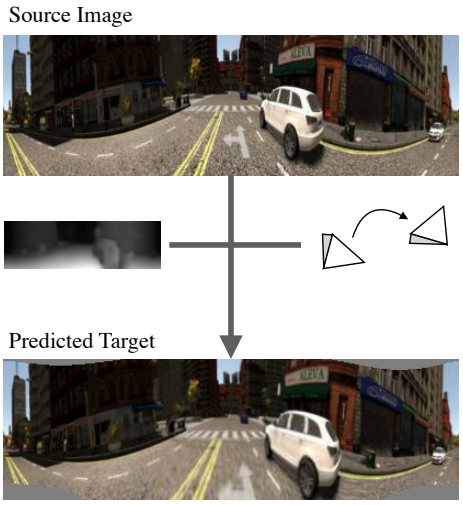


Fig. 4: We use panoramic view synthesis as a supervisor: the source panorama, depth, and pose transformation are used to synthesize a target view, and the loss is computed as the difference between the actual and synthesized views. As the synthesized view improves, the depth and pose predictions improve.

The learner uses *view synthesis*—the prediction of a target frame given source frames—as a self-supervisory signal (Figure 4). The network takes one target frame and several neighboring source frames as input. At each training step, the joint DispNet and PoseExpNet stacks predict (a) the depth of the target frame and (b) the pose of each source frame in the target frame's coordinate system. These predictions are then applied to the source images to synthesize the target image through *projective inverse warping*.

Let $\mathcal{D}$ be the predicted depth map for the target image. The corresponding 3D point $X^{i,j}_{\text{target}}$ for pixel i, j in the target image is found by unprojection according to the predicted depth value at $\mathcal{D}^{i,j}$.

$$X^{i,j}_{\text{target}} = \text{unproject}(i, j, \mathcal{D}^{i,j}) \tag{1}$$

This 3D point is then transformed into the source image's coordinate frame according to $\mathcal{P}_{\text{source}}$ the predicted pose of the source image.

$$X^{i,j}_{\text{source}} = \text{transform}(\mathcal{P}_{\text{source}}, X^{i,j}_{\text{target}}) \tag{2}$$

Finally, the source image is sampled using bilinear interpolation at the projection of $X^{i,j}_{\text{source}}$.

$$\mathcal{I}^{i,j}_{\text{proj}} = \text{sample}(\mathcal{I}_{\text{source}}, \text{project}(X^{i,j}_{\text{source}})) \tag{3}$$

The learner then tries to minimize the photometric error between the $\mathcal{I}_{\text{target}}$ and $\mathcal{I}_{\text{proj}}$, the source image warped into the target image's coordinate frame. See Figure 6 for an illustration. The learner reduces the photometric error by improving the depth and pose predictions [8], allowing the network to learn scene structure from monocular images without labeled depth maps.

For this model, we use a three-part objective function. The main component is the *photometric loss* ($\mathcal{L}_{\text{pixel}}$), which minimizes the difference between synthesized views and the target view. This is regularized by the *smooth loss* ($\mathcal{L}_{\text{smooth}}$), which minimizes the second derivatives with respect to the depth and the optional *explainability loss* ($\mathcal{L}_{\text{exp}}$), which makes the model more resilient to anomalous input (e.g. moving objects). If λ_s and λ_e represent the smooth and explainability weights, the total loss can be written as follows:

$$\mathcal{L} = \sum_{\text{scales}} \left(\sum_{\text{sources}} \mathcal{L}_{\text{pixel}} + \lambda_s \mathcal{L}_{\text{smooth}} + \sum_{\text{sources}} \lambda_e \mathcal{L}_{\text{exp}} \right) \tag{4}$$

If $\mathcal{I}$ is an RGB image, $\mathcal{D}$ is the depth prediction, and $\mathcal{E}$ is the explainability mask, the three loss components at pixel i, j at each scale can be written as follows:

$$\mathcal{L}^{i,j}_{\text{pixel}} = \sum_{i,j} \mathcal{E}^{i,j} \left| \mathcal{I}^{i,j}_{\text{proj}} - \mathcal{I}^{i,j}_{\text{target}} \right| \tag{5}$$

$$\mathcal{L}^{i,j}_{\text{smooth}} = \left| \frac{\delta^2 \mathcal{D}^{i,j}}{\delta x^2} \right| + \left| \frac{\delta^2 \mathcal{D}^{i,j}}{\delta x \delta y} \right| + \left| \frac{\delta^2 \mathcal{D}^{i,j}}{\delta y \delta x} \right| + \left| \frac{\delta^2 \mathcal{D}^{i,j}}{\delta y^2} \right| \tag{6}$$

$$\mathcal{L}^{i,j}_{\text{exp}} = \sum \text{softmax}\left(\mathcal{E}^{i,j}\right) \tag{7}$$

We also experimented with the image-aware depth smoothness loss term introduced by Wang et al. [11]:

$$\mathcal{L}^{i,j}_{\text{smooth}} = e^{-\nabla^2 \mathcal{I}^{i,j}} \left(\left| \frac{\partial \mathcal{D}^{i,j}}{\partial x^2} \right| + \left| \frac{\partial \mathcal{D}^{i,j}}{\partial x \partial y} \right| + \left| \frac{\partial \mathcal{D}^{i,j}}{\partial y^2} \right| \right) \tag{8}$$

The advantage of this smoothing term is that it reduces the depth smoothing effect at image edges.

Two major modifications were required to allow for cylindrical input: 1) the view synthesis functions were modified to account for cylindrical projection, and 2) the convolutional layers, resampling functions, and loss were modified to preserve horizontal wrapping.

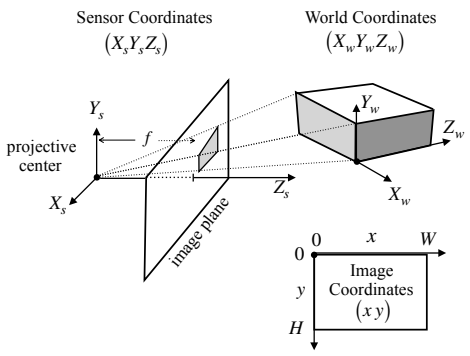


Fig. 5: Pinhole projection model. The 3D world (on the world coordinate system) is projected on a flat image plane; the image plane is a focal length f away from the projective center along the Z_s axis (on the sensor coordinate system). The result is a $W \times H$ rectangular image.

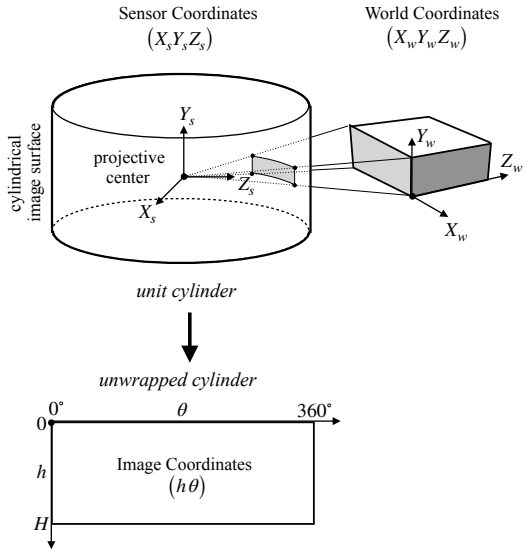


Fig. 6: Cylindrical projection model. In contrast with pinhole projection, this projects an image onto a curved cylindrical surface. The final result is a rectangular image with height H and a width representing the full 360°.

1) Camera Projection and Cylindrical Panoramas: The view synthesis function introduced by Zhou et al. [8] works by warping the source images onto the target image's coordinate frame using the predicted target inverse depth map and the source image relative pose. To adapt this process to work with cylindrical input, we modified the mapping functions between the pixel, camera, and world coordinate frames.

Most structure-from-motion systems expect *pinhole projection* images as input. Pinhole projection images project a 3D scene from the world coordinate system onto a flat image plane; this process can be described by the focal length f, principle point c, and the image height H and width W, as shown in Figure 5.

In contrast, *cylindrical projection* projects the 3D world onto a curved cylindrical surface, as seen in Figure 6. The goal of this process is to take a 3D point in the world coordinate system and project it onto a rectangular cylindrical panorama. This requires projecting the 3D point onto the cylindrical image surface and converting the image surface into a Cartesian coordinate system.

The transformation between the sensor and pixel coordinate systems can be described by the following equations. A 3D point $P = (x_s, y_s, z_s)$ in the sensor coordinate system projects to a 2D point $Q = (\theta, h)$ on the unit cylinder around the origin according to the following formula:

$$\begin{bmatrix} \theta \\ h \end{bmatrix} = \begin{bmatrix} \arctan\left(\frac{x_s}{z_s}\right) \\ \frac{y_s}{\sqrt{x_s^2 + z_s^2}} \end{bmatrix} \tag{9}$$

The inverse projection from the unit cylinder to a 3D point in the sensor coordinate system is as follows:

$$\begin{bmatrix} x_s \\ y_s \\ z_s \end{bmatrix} = \begin{bmatrix} d \sin\theta \\ dh \\ d \cos\theta \end{bmatrix} \tag{10}$$

where d is the depth of the point.

For our experiments, we modified the Tensorflow implementation of SfMLearner by Zhou et al. [8] to use these cylindrical un-projection and projection functions.

2) Horizontal Wrapping: In order to extend the model for cylindrical panoramic images, we modified the convolutional layers, smooth loss function, and 2D projection to account for horizontal wrapping.

Unlike pinhole images, cylindrical images wrap horizontally. For a convolutional layer to work with cylindrical input, it must preserve this horizontal wrapping property, rather than using zero-padding as is typical. Horizontal wrapping can be done by padding the right side of the tensor with columns from the left and vice-versa, as depicted in Figure 7.

We added wrap padding to all convolutions in the network architecture. We also added wrap padding to the smooth gradient loss computation and the bilinear sampler used for view synthesis.

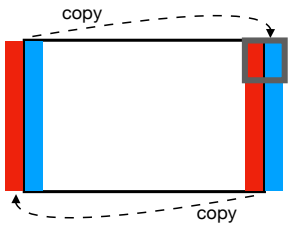


Fig. 7: An example of a horizontally wrapping convolutional layer. The wrapping is achieved by copying the left-most columns to the right side and vice-versa.

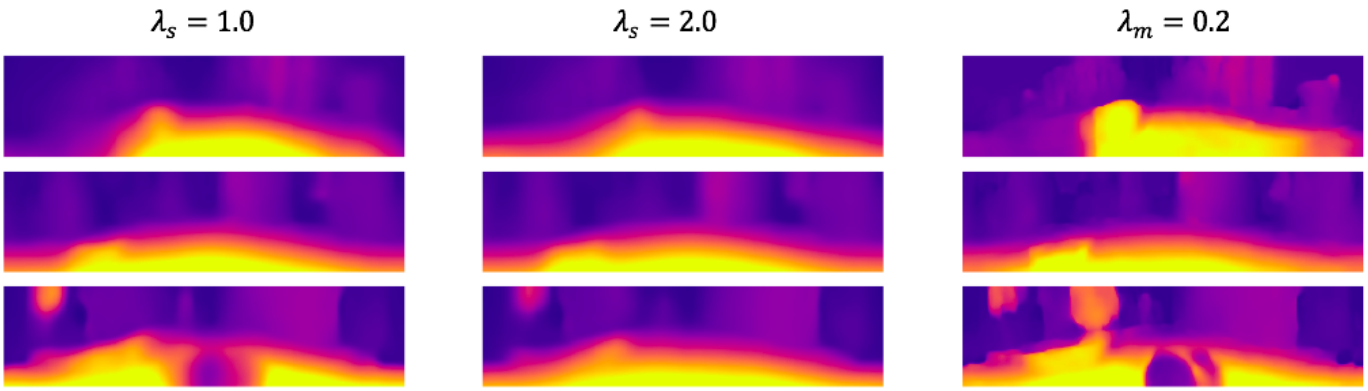


Fig. 8: A comparison of different loss functions. The first and second columns represent second-order depth gradient loss at different weights: $\lambda_s = 1.0$ and $\lambda_s = 2.0$. The rightmost column shows an image-aware first-order gradient loss with a weight $\lambda_m = 0.2$.

	Disparity (lower is better)				Accuracy (higher is better)		
	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Without wrapping	0.3016	2.5437	6.2202	0.3958	0.6689	0.8280	0.9011
With wrapping	**0.2994**	**2.5149**	**6.0375**	**0.3870**	**0.6757**	**0.8358**	**0.9065**

TABLE I: Depth accuracy results when training and testing our network architecture with and without horizontal wrapping. The model trained with horizontal wrapping is better on all metrics.

IV. Experiments

Several experiments were performed to evaluate the performance of our proposed method. We performed an ablation study on synthetic data, comparing the model with and without horizontal wrapping. We also evaluated the accuracy of pose estimation with different input fields of view. Finally, to demonstrate the model's effectiveness on real-world footage, we trained and tested the network on a new panoramic video dataset and qualitatively evaluated the results.

CylindricalSfMLearner was implemented in the open-source machine learning library Tensorflow [28]. Our code is publicly available on GitHub at https://github.com/jonathanventura/cylindricalsfmlearner.

A. Monocular Depth Estimation

While there are many standard datasets such as KITTI [29] and CityScapes [30] that provide standard field-of-view video with registered ground truth depth, we were unable to find a similar dataset containing panoramic video and associated ground truth with suitable characteristics. For this reason, we performed a quantitative evaluation of our work on synthetic data.

SYNTHIA-Seqs [31] is a synthetic dataset of driving data designed to mimic the properties of popular front-view datasets like KITTI [29] and CityScapes [30]. For this experiment, we used sequences 02 and 05 (NYC-like city driving) in the spring-, summer-, and fall-like conditions captured from the left and right stereo cameras.

To prepare the data for training, we first stitched the four perspective views into a single 360° cylindrical panorama. Next, we identified static frames using the global pose ground truth; these frames were excluded from the final formatted dataset using the same technique as SfMLearner [8]. The panoramas were then resized to 512×128 and concatenated to form three-frame sequences. After processing, this dataset contained 4,398 panoramic views. This was split into train, test, and validation subsets comprised of approximately 80%, 10%, and 10% of the data, respectively.

We experimented with various settings of the smoothing term and also alternative smoothing terms such as the image-aware first-order gradient loss from Wang et al. [11]. A comparison is shown in Figure 8. We found that while increasing the smoothness reduces the detail in the depth prediction, it tended to increase the depth accuracy. For the remaining experiments on Synthia, we chose a smoothness term of $\lambda_s = 2.0$ and $\lambda_e = 0$ and trained all models for 60,000 steps.

We compared the model with and without horizontal wrapping to determine whether the wrapping property is important for prediction accuracy. We used a set of disparity and accuracy metrics common in SfM research that each capture a different aspect of the prediction error; for more details, please refer to [5].

Table I shows the results of our experiment. The model trained with horizontal wrapping in the convolutional layers outperformed the model without wrapping on all metrics.

B. Pose Estimation

We hypothesized that one benefit of end-to-end training of depth and ego-motion estimation with panoramic video would be an improvement in pose accuracy. We reasoned that the larger field-of-view (FOV) would provide more rays constraining the camera pose and thus improve pose estimation

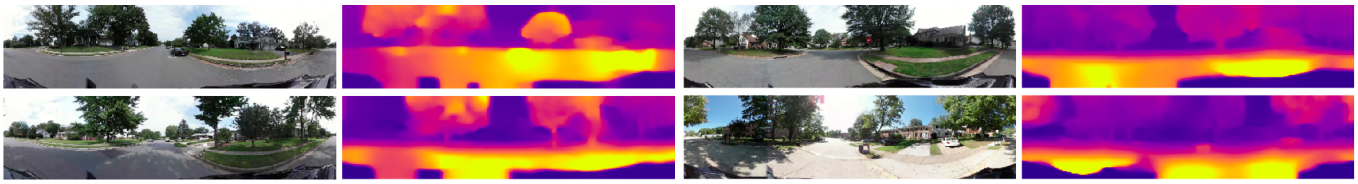


Fig. 9: Depth predictions for the self-collected headcam dataset. Our cylindrical CNN model achieves highly detailed depth predictions on panoramic input. The missing depth predictions at the bottom are due to the helmet being visible in the frame.

accuracy. To test this hypothesis we trained different models on crops of the full cylindrical panorama to simulate cameras of different FOVs. We did not use horizontal wrapping on the cropped images.

We trained the models on data from Synthia sequences 02 and 05 and tested on unseen data from sequences 02, 04, and 05. This split was selected due to the limited number of visually similar Synthia sequences. Sequences 02 and 05 were partitioned randomly into training and testing sets, and no frame seen during training as either source or target frame was used in the testing set.

Table II reports the mean average trajectory error (ATE) [3] achieved by each model on Synthia sequences 02, 04, and 05. Sequence 04 is visually dissimilar from the training data, which explains why the ATE is higher on sequence 04. In general, the higher FOV models have lower ATE.

Seq.	FOV (deg)	Wrap	ATE Mean	ATE Std. Dev.
02	360	Yes	**0.0139**	**0.0210**
02	360	No	0.0143	0.0211
02	270	No	0.0150	0.0230
02	180	No	0.0203	0.0296
02	100	No	0.0324	0.0498
05	360	Yes	0.0138	0.0195
05	360	No	0.0132	**0.0189**
05	270	No	**0.0131**	0.0191
05	180	No	0.0167	0.0224
05	100	No	0.0258	0.0383
04	360	Yes	**0.0342**	0.0351
04	360	No	0.0345	**0.0322**
04	270	No	0.0355	0.0348
04	180	No	0.0384	0.0372
04	100	No	0.0371	0.0359

TABLE II: Evaluation of the effect of input field-of-view on average trajectory error (ATE). In general, a wider field-of-view leads to higher pose accuracy (lower mean ATE). The training data contained images from Sequences 02 and 05 but not Sequence 04.

C. Real data: Headcam Dataset

While the SYNTHIA-Seqs dataset allowed us to explore the effect of cylindrical projection and increasing field of views, its major limitation is that it is synthetic. In order to determine if cylindrical projection is successful with real-world input, we trained and evaluated a model on our own panoramic dataset.

We assembled a panoramic video dataset which we call Headcam. It was collected by affixing a consumer-grade panoramic camera, the 2016 Samsung Gear 360, to a bicycle helmet and biking around neighborhoods in Northern Virginia. This dataset includes about two hours of footage collected over three days and has been released publicly on Zenodo under a Creative Commons Attribution license [32].

This footage was first stitched into equirectangular panoramic video using the Samsung Gear 360 software. It was then broken into frames at 5fps, warped into a cylindrical projection model, resized to 512×128, and formatted into three-frame sequences. The final formatted dataset contains 27,538 frames. Training was conducted on 90% of the data, and qualitative testing was done on the remaining 10%.

This model was trained with the same configuration as above except for the smooth loss term. This model was trained using image-aware smooth loss ($\lambda_m = 0.2$) [11] as opposed to the second-order gradient loss. This change helped to improve the definition of predicted object shapes such as trees and cars.

Predictions on the test set can be seen in Figure 9. Qualitatively, the model generates visually reasonable predictions with well-defined object boundaries. However, the model makes several common errors. Most notably, due to its constant presence frame-to-frame, the model predicts that the helmet at the bottom of the image has a large depth. Future work might explore ways to mitigate this through masking or other techniques. We also noticed occasional holes in the depth prediction in the road, similar to the predictions on Synthia when using a low smoothing term.

V. Conclusion

We introduce the first unsupervised model for learning depth and ego-motion directly from panoramic video input. In contrast to previous work, we use the cylindrical panoramic projection which allows for direct application of existing convolutional neural network models to panoramic input with little modification. Our evaluation on synthetic and real data shows that learning from cylindrical panoramic input is as effective as pinhole projection input in producing accurate and detailed depth predictions.

We also contribute a novel dataset of real panoramic video suitable for unsupervised learning of depth and ego-motion. The dataset was captured on city streets from a helmet-mounted camera while riding a bicycle and thus contains a variety of motions, moving objects, and other challenging conditions, making it a difficult and interesting dataset for future research on learning panoramic structure-from-motion.

While cylindrical projection makes the application of convolutional neural networks to panoramic input simple, this representation has some limitations. First, the top and bottom of the complete spherical field-of-view are not included in the cylindrical representation; however, in most applications, these regions are not important. Second, our model does not achieve full rotation invariance like, for example, the Spherical CNN model of Cohen et al. [12]. Our cylindrical CNN model is invariant to rotation about the vertical axis (yaw) but not rotation about the other two axes (pitch and roll). However, pitch and roll rotations of the camera are relatively rare in street-level driving tasks as evaluated in this paper.

Future work could also include a comparison of our cylindrical model against the various spherical CNN and cube map CNN models that have been proposed, and experimenting with other state-of-the-art depth prediction architectures [33], [34] adapted for cylindrical panoramic input.

ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grant Nos. 1659788 and 1464420. This work was performed as part of an REU program at the University of Colorado, Colorado Springs.

REFERENCES

[1] A. Serrano, I. Kim, Z. Chen, S. DiVerdi, D. Gutierrez, A. Hertzmann, and B. Masia, "Motion parallax for 360° RGBD video," *IEEE Trans. Visualization and Computer Graphics*, vol. 25, no. 5, 2019.

[2] J. R. Bergen, P. Anandan, K. J. Hanna, and R. Hingorani, "Hierarchical model-based motion estimation," in *European Conf. Computer Vision*, 1992.

[3] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: a versatile and accurate monocular SLAM system," *IEEE Trans. Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[4] A. Saxena, M. Sun, and A. Y. Ng, "Make3d: Learning 3d scene structure from a single still image," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 31, no. 5, pp. 824–840, 2009.

[5] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in *Advances in Neural Information Processing Systems*, 2014.

[6] F. Liu, C. Shen, G. Lin, and I. Reid, "Learning depth from single monocular images using deep convolutional neural fields," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 38, no. 10, 2016.

[7] C. Godard, O. Mac Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *Conf. Computer Vision and Pattern Recognition*, vol. 2, 2017.

[8] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised learning of depth and ego-motion from video," in *Conf. Computer Vision and Pattern Recognition*, vol. 2, 2017.

[9] S. Vijayanarasimhan, S. Ricco, C. Schmid, R. Sukthankar, and K. Fragkiadaki, "Sfm-net: Learning of structure and motion from video," *arXiv preprint arXiv:1704.07804*, 2017.

[10] R. Mahjourian, M. Wicke, and A. Angelova, "Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints," in *Conf. Computer Vision and Pattern Recognition*, 2018.

[11] C. Wang, J. M. Buenaposada, R. Zhu, and S. Lucey, "Learning depth from monocular videos using direct methods," in *Conf. Computer Vision and Pattern Recognition*, 2018, pp. 2022–2030.

[12] T. S. Cohen, M. Geiger, J. Koehler, and M. Welling, "Spherical CNNs," in *Int. Conf. Learning Representations*, Vancouver, Canada, 2018.

[13] C. Esteves, C. Allen-Blanchette, A. Makadia, and K. Daniilidis, "Learning SO(3) Equivariant Representations with Spherical CNNs," in *European Conf. Computer Vision*, 2018.

[14] H.-T. Cheng, C.-H. Chao, J.-D. Dong, H.-K. Wen, T.-L. Liu, and M. Sun, "Cube padding for weakly-supervised saliency prediction in 360° videos," in *Conf. Computer Vision and Pattern Recognition*, June 2018.

[15] F.-E. Wang, H.-N. Hu, H.-T. Cheng, J.-T. Lin, S.-T. Yang, M.-L. Shih, H.-K. Chu, and M. Sun, "Self-supervised learning of depth and camera motion from 360° videos," in *Asian Conf. Computer Vision*, 2018.

[16] B. Coors, A. P. Condurache, and A. Geiger, "SphereNet: Learning Spherical Representations for Detection and Classication in Omnidirectional Images," in *European Conf. Computer Vision*, 2018.

[17] K. Tateno, N. Navab, and F. Tombari, "Distortion-Aware Convolutional Filters for Dense Prediction in Panoramic Images," in *European Conf. Computer Vision*, 2018.

[18] Q. Zhao, C. Zhu, F. Dai, Y. Ma, G. Jin, and Y. Zhang, "Distortion-aware CNNs for Spherical Images," in *Int. Joint Conf. Artificial Intelligence*. Stockholm, Sweden: Int. Joint Conf. Artificial Intelligence Organization, Jul. 2018.

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

[20] V. R. Kumar, S. Milz, C. Witt, M. Simon, K. Amende, J. Petzold, S. Yogamani, and T. Pech, "Near-eld Depth Estimation using Monocular Fisheye Camera: A Semi-supervised learning approach using Sparse LIDAR Data," in *Deep Vision: Beyond Supervised Learning*, Jun. 2018, p. 3, extended Abstract.

[21] M. Weiler, F. A. Hamprecht, and M. Storath, "Learning Steerable Filters for Rotation Equivariant CNNs," in *Conf. Computer Vision and Pattern Recognition*, 2018.

[22] R. Khasanova and P. Frossard, "Graph-based classification of omnidirectional images," in *IEEE Int. Conf. Computer Vision Workshops*, 2017.

[23] G. P. de La Garanderie, A. A. Abarghouei, and T. P. Breckon, "Eliminating the Blind Spot: Adapting 3d Object Detection and Monocular Depth Estimation to 360 Panoramic Imagery," in *European Conf. Computer Vision*, Aug. 2018, 00000 arXiv: 1808.06253.

[24] Y.-C. Su and K. Grauman, "Learning spherical convolution for fast features from 360 imagery," in *Advances in Neural Information Processing Systems*, 2017.

[25] C. Esteves, C. Allen-Blanchette, X. Zhou, and K. Daniilidis, "Polar Transformer Networks," in *Int. Conf. Learning Representations*, 2018.

[26] H. Salehinejad, S. Naqvi, E. Colak, J. Barfett, and S. Valaee, "Cylindrical Transform: 3d Semantic Segmentation of Kidneys With Limited Annotated Images," in *IEEE Global Conf. Signal and Information Processing*, Sep. 2018, 00000 arXiv: 1809.10245.

[27] N. Mayer, E. Ilg, P. Hausser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox, "A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation," in *Conf. Computer Vision and Pattern Recognition*, 2016.

[28] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org.

[29] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Conf. Computer Vision and Pattern Recognition*, 2012.

[30] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Conf. Computer Vision and Pattern Recognition*, 2016.

[31] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez, "The SYNTHIA dataset: A large collection of synthetic images for semantic segmentation of urban scenes," in *Conf. Computer Vision and Pattern Recognition*, Jun. 2016, pp. 3234–3243.

[32] A. Sharma and J. Ventura, "Headcam: Cylindrical panoramic video dataset for unsupervised learning of depth and ego-motion," doi.org/10.5281/zenodo.3520963, Oct. 2019.

[33] Z. Yin and J. Shi, "GeoNet: Unsupervised learning of dense depth, optical flow and camera pose," in *Conf. Computer Vision and Pattern Recognition*, vol. 2, 2018.

[34] D. Xu, W. Wang, H. Tang, H. Liu, N. Sebe, and E. Ricci, "Structured attention guided convolutional neural fields for monocular depth estimation," in *Conf. Computer Vision and Pattern Recognition*, 2018, pp. 3917–3925.