Consistent Depth Estimation from Video

Libing Zeng
libingzeng@tamu.edu

Abstract

For the final project, we implemented a model, in PyTorch, to generate consistent depth estimation from video. Specifically, we first train a self-supervised Depth-Net [Dai et al. [2020]] to generate depth prediction for each frame of the video, and then we use the consistent video depth estimation [Luo et al. [2020]] to optimize the trained model from the first step to produce consistent depth estimation from video.

1 Related Work

In this section, we briefly review leaning-based depth estimation methods, including supervised depth estimation methods and self-supervised depth estimation methods.

Supervised Depth Estimation The depth estimation is modeled as a regression problem in most supervised approaches [Saxena et al. [2006]], minimizing the distance from the predicted depth and the ground truth. Deep learning based methods have been successfully used in single image depth estimation [Eigen et al. [2014], Eigen and Fergus [2015], Fu et al. [2018]]. However, training such models requires ground truth maps that are not easy to acquire.

Self-supervised Depth Estimation Due to the difficulty of the collection of training data, self-supervised learning methods have received much attention for their ability to learn a monocular depth estimation network directly from raw stereo pairs [Godard et al. [2017]] or monocular video [Zhou et al. [2017]]. The core idea is to apply differentiable warp and minimize photometric reprojection error.

2 Depth Estimation from Video

In this section, we will show how to build the depth prediction model, which losses are used, how to train it, and what the generated results look like.

2.1 Depth Estimation Network

I followed the depth network from [Dai et al. [2020]], which basically adopt the architecture in [Yin and Shi [2018]]. As shown in Fig. 1, An input RGB image is passed into the model and the synthesized image can be reconstructed from the depth generated by the model and the corresponding camera ego-motion parameters.

As to camera ego-motion estimations, there are several options. We can produce the camera ego-motion parameters using an pre-trained network, jointly trained model, the Libviso2 [Geiger et al.], or Colmap [Schönberger and Frahm [2016], Schönberger et al. [2016]]. In my implementation, we used Colmap.

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

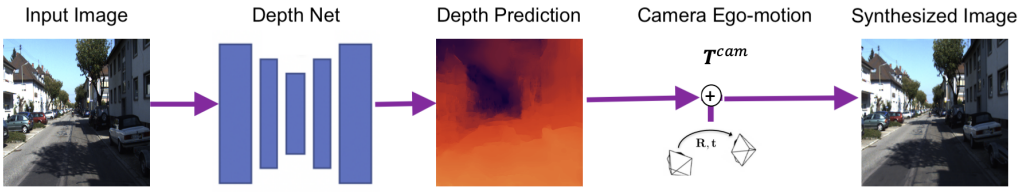

Figure 1: Depth Estimation Network. An input RGB image is passed into the model and the synthesized image can be reconstructed from the depth generated by the model and the corresponding camera ego-motion parameters.

2.2 Loss Functions

To train this Depth Estimation model, three loss terms are employed: photometric loss L_p, left-right photometric loss L_{lrp}, and disparity smoothness loss L_{disp}.

Photometric Loss L_p penalizes the photometric erros between the synthesized image $\hat{I}$ and the source input image I. $\hat{I}$ is reconstructed using the depth generated from the depth model and the corresponding camera ego-motion parameters. We use a robust image similarity measurement SSIM for L_p with $\alpha = 0.85$.

$$L_p = \alpha \frac{1 - SSIM(I, \hat{I})}{2} + (1 - \alpha) \left\| I - \hat{I} \right\|_1, \tag{1}$$

Left-right Photometric Loss L_{lrp} is imposed to address the scale ambiguity of the monocular depth prediction. The direct output from the depth network is actually the disparity. It can be used to synthesize the left image from its right counterpart, and vice versa. L_lrp penalizes the photometric errors of the synthesized stereo images.

Disparity Smoothness Loss L_{disp} is enforced to penalize a fluctuated disparity prediction. An edge-ware smoothness term is imposed.

$$L_{disp} = |\partial_x d| \, e^{\|-\partial_x I\|} + |\partial_y d| \, e^{\|-\partial_y I\|}, \tag{2}$$

2.3 Experiments

We use KITTI raw dataset [Geiger et al. [2012]] to train the depth prediction model. We resize all images into a fixed size 256×256.

The depth predictions generated by the depth network are show in Fig. 2. Here are a sequence of five images and their corresponding depth predictions. From the top row of depth predictions, we can see that the predictions are not temporally consistent, especially, the first three predictions. If we go through the predictions like playing a video, we can see serious flickering artifacts between neighboring depth predictions.

3 Consistent Depth Estimation from Video

In this section, we will use the method proposed by Luo et al. [2020] to optimize the trained depth estimation network.

3.1 Method Overview

Luo et al. [2020] present this method for reconstructing dense, geometrically consistent depth for all pixels in a monocular video. The key idea is to leverage a conventional structure-from-motion reconstruction to establish geometric constraints on pixels in the video. At test time, they fine-tune the trained depth network to satisfy the geometric constraints of a particular input video. The method

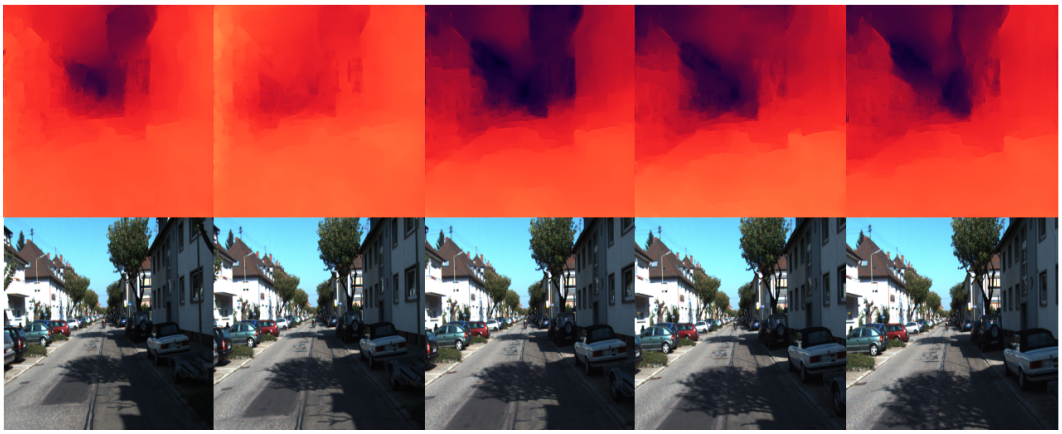

Figure 2: Depth Predictions. Here are a sequence of five images and their corresponding depth predictions. From the top row of depth predictions, we can see that the predictions are not temporally consistent, especially, the first three predictions.

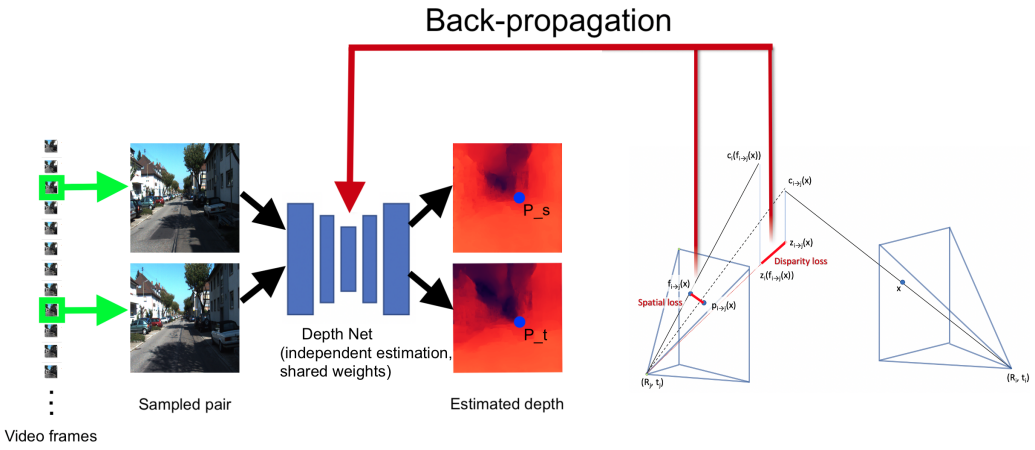

Figure 3: Method overview. With a monocular video as input, we sample a pair of (potentially distant) frames and estimate the depth using a pre-trained, single-image depth estimation model to obtain initial depth maps. From the pair of images, we establish correspondences using optical flow with forward-backward consistency check. We then use these correspondences and the camera poses to extract geometric constraints in 3D. We decompose the 3D geometric constraints into two losses: 1) spatial loss and 2) disparity loss and use them to fine-tune the weight of the depth estimation network via standard backpropagation. This test-time training enforces the network to minimize the geometric inconsistency error across multiple frames for this particular video. After the fine-tuning stage, our final depth estimation results from the video is computed from the fine-tuned model.

overview is shown in Fig. 3. With a monocular video as input, we sample a pair of (potentially distant) frames and estimate the depth using a pre-trained, single-image depth estimation model to obtain initial depth maps. From the pair of images, we establish correspondences using optical flow with forward-backward consistency check. We then use these correspondences and the camera poses to extract geometric constraints in 3D. We decompose the 3D geometric constraints into two losses: 1) spatial loss and 2) disparity loss and use them to fine-tune the weight of the depth estimation network via standard backpropagation. This test-time training enforces the network to minimize the geometric inconsistency error across multiple frames for this particular video. After the fine-tuning stage, our final depth estimation results from the video is computed from the fine-tuned model.

Figure 4: Consistent Depth Predictions. Here are a sequence of five images and their corresponding depth predictions. From the top row of depth predictions, we can see that the predictions, compared to Fig. 2, are much more temporally consistent.

3.2 Loss Functions

As mentioned before, a novel geometric constraint loss is the key idea of the method. Test time fine-tuning the pre-trained single-image depth prediction model with the geometric loss could help it to produce more consistent depth for a particular input video.

Geometric Loss. For a give frame pair $(i, j) \in S$, the optical flow field $F_{i\to j}$ describes which pixel pairs show the same scene point. We can use the flow to test the geometric consistency of our current depth estimates: if the flow is correct and a flow-displaced point $f_{i\to j}(x)$ is identical to the depth-reprojected point $p_{i\to j}(x)$, then the depth must be consistent. The idea of the method is that we can turn this into a geometric loss $L_{i\to j}$ and back-propagate any consistency errors through the network, so as to coerce it into to producing depth that is more consistent than before. $L_{i\to j}$ comprises two terms, an image-space loss $L_{i\to j}^{spatial}$, and a disparity loss $L_{i\to j}^{disparity}$. For more details, please check the original paper Luo et al. [2020].

3.3 Experiments

This is a test-time fine-tuning method for some particular input video. I use the same test video as in the test stage of the pre-trained depth estimation model, so that we can do the comparisons between results generated using the model without fine-tuning and with fine-tuning.

The depth predictions generated by the fine-tuned depth network are show in Fig. 4. Here are a sequence of five images and their corresponding depth predictions. From the top row of depth predictions, we can see that the predictions, compared to Fig. 2, are much more temporally consistent.

4 Conclusion

In this project, we implement the depth estimation network proposed by Dai et al. [2020], and then use the test time fine-tuning method introduced by Luo et al. [2020] to optimize the pre-trained depth estimation network to generate temporally consistent depth predictions from video.

References

Qi Dai, Vaishakh Patil, Simon Hecker, Dengxin Dai, Luc Van Gool, and Konrad Schindler. Self-supervised object motion and depth estimation from video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2020.

D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 2650–2658, 2015. doi: 10.1109/ICCV.2015.304.

David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. 3, 06 2014.

Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361, 2012. doi: 10.1109/CVPR.2012.6248074.

Andreas Geiger, Julius Ziegler, and Christoph Stiller. Stereoscan: Dense 3d reconstruction in real-time. In *in IEEE Intelligent Vehicles Symposium, 2011*, pages 963–968.

Clément Godard, Oisin Mac Aodha, and Gabriel J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *CVPR*, 2017.

Xuan Luo, Jia-Bin Huang, Richard Szeliski, Kevin Matzen, and Johannes Kopf. Consistent video depth estimation. 39(4), 2020.

Ashutosh Saxena, Sung Chung, and Andrew Ng. Learning depth from single monocular images. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems*, volume 18, pages 1161–1168. MIT Press, 2006. URL `https://proceedings.neurips.cc/paper/2005/file/17d8da815fa21c57af9829fb0a869602-Paper.pdf`.

Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016.

Zhichao Yin and Jianping Shi. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In *CVPR*, 2018.

Tinghui Zhou, Matthew Brown, Noah Snavely, and David G. Lowe. Unsupervised learning of depth and ego-motion from video. In *CVPR*, 2017.