Robustness Meets Deep Learning: An End-to-End Hybrid Pipeline for Unsupervised Learning of Egomotion

Alex Zihao Zhu*, Wenxin Liu*, Ziyun Wang, Vijay Kumar, Kostas Daniilidis
University of Pennsylvania
{alexzhu, wenxinl, ziyunw, kumar, kostas}@seas.upenn.edu

Abstract

In this work, we propose a method that combines unsupervised deep learning predictions for optical flow and monocular disparity with a model based optimization procedure for camera pose. Given the flow and disparity predictions from the network, we apply a RANSAC outlier rejection scheme to find an inlier set of flows and disparities, which we use to solve for the camera pose in a least squares fashion. We show that this pipeline is fully differentiable, allowing us to combine the pose with the network outputs as an additional unsupervised training loss to further refine the predicted flows and disparities. This method not only allows us to directly regress pose from the network outputs, but also automatically segments away pixels that do not fit the rigid scene assumptions that many unsupervised structure from motion methods apply, such as on independently moving objects. We evaluate our method on the KITTI dataset, and demonstrate state of the art results, even in the presence of challenging independently moving objects.

1. Introduction

The advent of unsupervised methods for neural networks has resulted in the rapid growth of works that train networks that predict the camera pose and depth of a scene, such as [29], without any labeled training data. By utilizing the ability of neural networks to learn a prior for the scale of a scene, these networks are able to predict accurate 6-dof poses and depths without the scale ambiguity problem that optimization based structure from motion methods face for monocular inputs, and without the need for explicit feature extraction. However, these methods abstract away the function between the image and the pose within the network weights, and so cannot provide any guarantees on robustness or safety. Furthermore, outliers such as independently moving objects remain a challenging problem, as incorporating outlier rejection schemes into the pose when it is directly regressed from the network has proven difficult.

Most state-of-the-art learning frameworks for pose estimation have a network structure which estimates 6-dof pose from 2D images using convolutions. It is well established that CNN's are exceptionally strong when operating on spatially structured data such as 2D images, and we have seen highly successful examples in problems such as depth [9] and flow [19] prediction, where network based methods are now state of the art. However, whether a network with these architectures can extract 3D translations and rotations from the 2D image plane is not obvious, and learning based methods for camera pose have yet to beat traditional methods.

On the other hand, geometric optimization methods have seen immense success in this field with feature correspondences and outlier rejection using epipolar constraints [1]. Given accurate correspondences between images and depths, optimization methods such as Random Sample Concensus (RANSAC) [5] are able to recover extremely robust and accurate estimates of the camera pose and the surrounding scene. Direct visual odometry methods, for example, directly optimize on the photometric errors generated by warping each pixel by the estimated pose and depth. These methods have extremely high accuracy, and can make use of full amount of information from the image.

In this work, we decompose the classical direct visual odometry pipeline and introduce a learning-based front end. We combine the best of both worlds by leveraging the learning ability of neural networks to predict optical flow correspondences and disparities from a set of images, and applying a robust optimization backend using RANSAC to estimate pose from the network outputs.

By applying RANSAC for pose estimation, we are able to apply outlier rejection at a pixel level to extract only the set of predicted flow and disparity values that best fit the camera pose model. In particular, most SFM methods assume a static scene, and so are easily corrupted by independently moving objects. There have been a number of works that try to filter out these objects, for example by directly predicting a mask for the valid pixels in the scene

*These authors contributed equally to this work.

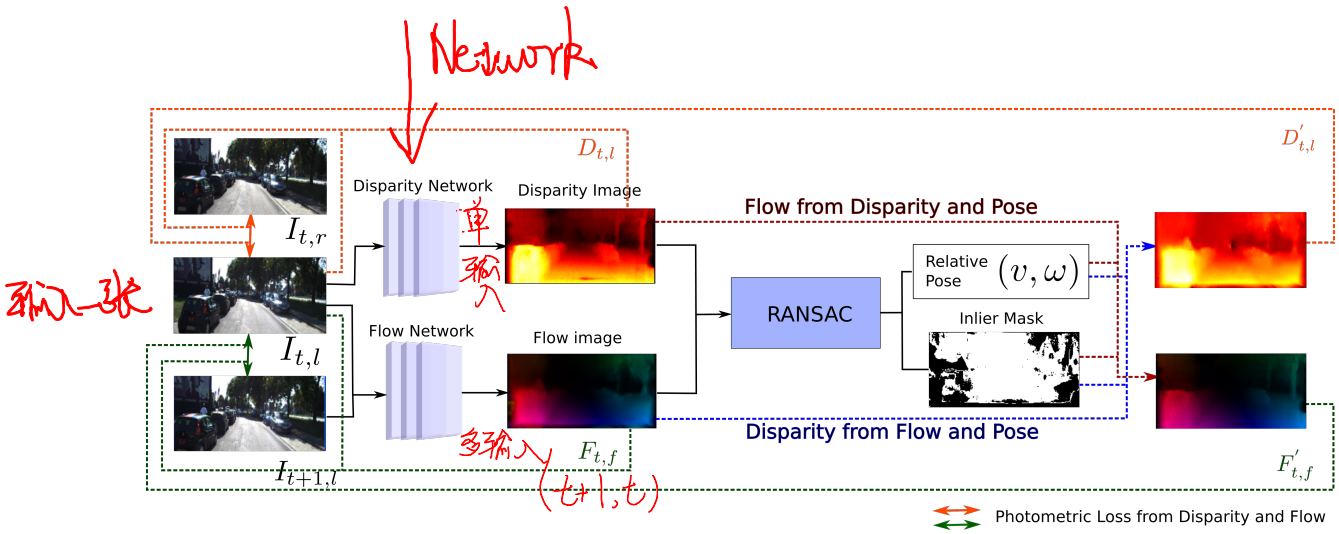

Figure 1. The pipeline given three images as inputs. Only forward and backward images are used as inputs, but a third image from the stereo is used for the disparity network and end-to-end training with RANSAC.

from the network [29, 20], or by detecting mismatches between the predicted optical flow and the disparity and pose [18, 26]. Our method does this through a principled geometric matching, without the need to separately learn these objects.

Our contributions can be summarized as:

- A novel end-to-end unsupervised structure from motion pipeline which uses two fully convolutional networks to predict optical flow and disparity from a pair of images, and a RANSAC outlier rejection scheme to robustly estimate camera pose.
- A novel set of photometric losses that uses the camera pose to estimate disparity from flow and vice versa, allowing for additional supervision of each modality from another image.
- Evaluation on the KITTI datasets, where we show robustness to independently moving objects, with ablations demonstrating the improvements of our RANSAC method.

2. Related Work

There have been a number of recent methods that leverage the principles of photoconsistency and image warping [11] to perform unsupervised learning of image motion. Garg et al. [6] and Godard et al. [9] showed that metric depth can be learned from a monocular camera by warping stereo images onto one another. Similarly, Yu et al. [12] and Meister et al. [15] use a similar transformation to learn optical flow from a pair of images. Zhou et al. [29] also showed that 3D camera motion can also be learned from this regime, by jointly predicting the egomotion of the camera, the depth of the scene, and combining the two to warp each pixel in the image. Since then, a number of methods have improved upon this scheme. Zhan et al [27] add a feature reconstruction loss to resolve ambiguities seen with a photometric loss, such as in textureless regions. Li et al. [13] use the egomotion to align the point clouds associated with the predicted depths, while Wang et al. [22] introduce a recurrent neural network to replace the feed-forward CNN.

However, these methods rely on a rigid scene assumption, and so are corrupted by independently moving objects in the scene. A number of works have tried to remove these objects from the loss function, including the works by Zhou et al. [29] and Vijayanarasimhan et al. [20], which predict masks to remove invalid pixels. Moreover, the works by Ranjan et al. [18], Luo et al. [14] and Yang et al. [26] use the mismatches between egomotion and depth and optical flow predictions to generate these masks.

In a similar vein to our work, Wang et al. [21] train a depth network by applying direct visual odometry optimization using the predicted depths to minimize the image reprojection error, and Yang et al. [24] incorporate a separately trained depth network into the Direct Sparse Odometry [3] framework, providing a monocular odometry pipeline that is able to estimate metric trajectories comparable to a stereo setup.

Our work, on the other hand, uses an interpretable geometric, model-based backend to jointly detect outliers and regress pose, while leveraging the strength of the network in 2D to predict both correspondences and depths.

3. Method

Our pipeline consists of two neural networks, one which predicts optical flow from a pair of images, and one which predicts disparity from a single image. Both networks are based on the FlowNet-S [4] architecture, which consists of an encoder-decoder pipeline with skip connections and a multi-scale loss. However, we have significantly reduced the size of the network, which we discuss in Sec. 3.8.

Each network can be trained separately, with a combi-

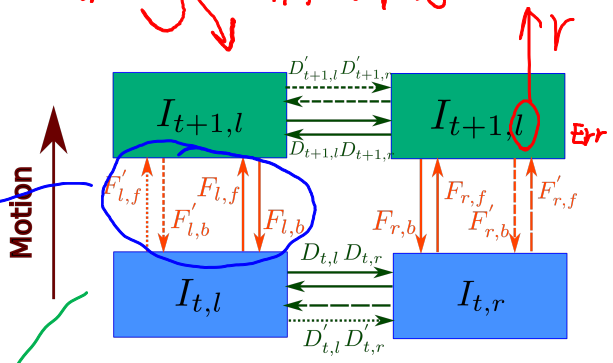

Figure 2. Losses from 2 pairs of stereo images given by forward/backward flow and left/right disparity. The inner circle of losses are from the flow and disparity network outputs, and the outer circle losses are from the calculated flow and disparity given the relative pose estimate from RANSAC. Arrows of the same style (type of dash) corresponds to the same pose estimate.

nation of photometric, smoothness and geometric losses, which we describe in Sec. 3.1-3.5. At training time, we sample four images randomly from the dataset, consisting of two stereo pairs from times t_0 and t_1. We then pass these images into both networks to predict a forward, $F_{t_0 \to t_1}(\vec{x})$, and backward, $F_{t_1 \to t_0}(\vec{x})$, flow for each image in the stereo pairs, and a disparity for each image, $D(\vec{x})$, resulting in four flow and disparity predictions respectively for every two stereo image pairs. The inner circle on Fig. 2 illustrates the 4-way loss pattern we adopted to maximize the usage of training data within a pair.

As both flow and disparity can both be modeled by a general image warping function that warps a pixel, $\vec{x}$, to another location: $W(\vec{x}_i) \to \vec{x}_j$, we will describe our loss functions in terms of this general warping, which can then be substituted by either optical flow or disparity.

We use RANSAC, Sec. 3.6, for relative pose estimation and outlier rejection, taking the flow and disparity from the networks as inputs. Then, we calculate the flow from pose and disparity, and the disparity from pose and flow. The *refinement* losses, Sec. 3.7, from the calculated flow and disparity implicitly enforces the epipolar constraint between the flow and disparity outputs from the networks, making three images visible when calculating the gradient. This pipeline is depicted in Fig. 1, and the additional losses are illustrated as the outer circle on Fig. 2. As the inlier mask from RANSAC removes any points that do not fit the rigid motion model, any moving objects are also automatically segmented. As a result, we only back-propagate through the set of inliers, and the whole pipeline is trainable end-to-end.

3.1. Appearance Loss

Given a pair of images, I_i, I_j, and a warping between them, $W_{i \to j}$, we apply a photoconsistency assumption, which assumes that the correct warp should warp pixels from I_i to pixels at I_j with the same intensity. We therefore apply the following photometric loss to enforce the constraint:

$$\mathcal{L}^{W_{i\to j}}_{\text{photo}}(\vec{x}) = \rho(I_i(\vec{x}) - I_j(W_{i\to j}(\vec{x}))) \quad (1)$$

$$\rho(x) = \sqrt{x^2 + \epsilon^2} \quad (2)$$

where $\rho(x)$ is the robust Charbonnier loss function.

We combine this photometric loss with a structural similarity (SSIM) loss [23] to form our appearance loss:

$$\mathcal{L}^{W_{i\to j}}_{\text{appearance}}(\vec{x}) = (1-\alpha)\text{SSIM}(\vec{x}) + \alpha\mathcal{L}_{\text{photo}}(\vec{x}) \quad (3)$$

3.2. Geometric Consistency

Several works, such as [9, 15], have proposed additional losses to constrain geometric consistency between the forward and backward (or left and right) estimates of a warp. That is, the backward warp, warped to the previous image, should be equivalent to the negative of the forward warp. We apply this constraint to both the disparity and flow:

$$\mathcal{L}^{W_{i\to j}}_{\text{consistency}}(\vec{x}) = \rho(W_{i\to j}(\vec{x}) + W_{j\to i}(W_{i\to j}(\vec{x}))) \quad (4)$$

3.3. Smoothness Regularization

We further apply a constraint for the warp to be locally smooth. This is applied with an edge aware smoothness loss, which weighs the loss lower at pixels with high image gradient:

$$\mathcal{L}^{W_{i\to j}}_{\text{smooth}}(\vec{x}) = \rho\left(\delta_x W_{i\to j}(\vec{x}) e^{-|\delta_x I(\vec{x})|}\right) + \rho\left(\delta_y W_{i\to j}(\vec{x}) e^{-|\delta_y I(\vec{x})|}\right) \quad (5)$$

3.4. Motion Occlusion Estimation

For a given warp function, there may be pixels that are warped out of the image. Applying a photometric or geometric consistency loss at these pixels would introduce errors into the model, as we cannot sample from points from outside the image. To resolve this issue, we generate a mask, $M^{W_{i\to j}(\vec{x})}_{\text{occ}}$, which is 0 for pixels that are warped out of the image, and 1 otherwise, and is used for the photometric and geometric consistency losses.

3.5. Total Loss

For a single warp, $W_{i\to j}(\vec{x})$, the total loss is:

$$\mathcal{L}^{W_{i\to j}}_{\text{total}} = \sum_{\vec{x}} M^{W_{i\to j}}_{occ}(\vec{x}) \left(\mathcal{L}^{W_{i\to j}}_{\text{photo}}(\vec{x}) + \lambda_1 \mathcal{L}^{W_{i\to j}}_{\text{consistency}}(\vec{x})\right) + \lambda_2 \mathcal{L}^{W_{i\to j}}_{\text{smoothness}}(\vec{x}) \quad (6)$$

The final loss, then, is the sum of $\mathcal{L}_{\text{total}}$ for each of the flow and disparity predictions:

$$\mathcal{L}_{\text{total}} = \sum_{c\in\{l,r\}} \mathcal{L}_{\text{total}}^{F_{0\to 1,c}} + \mathcal{L}_{\text{total}}^{F_{1\to 0,c}} + \sum_{t\in\{0,1\}} \mathcal{L}_{\text{total}}^{D_{l\to r,t}} + \mathcal{L}_{\text{total}}^{D_{r\to l,t}} \quad (7)$$

3.6. RANSAC Outlier Rejection

Given the flow and disparity predictions, F, D, we can use the Random Sample Consensus (RANSAC) algorithm [5] to estimate the best set of inliers to be used to estimate the pose of the cameras. RANSAC is a robust inlier selection scheme that allows us to filter out outliers such as independently moving objects from the optimization to estimate pose. The motion field equation that constrains pose, depth and optical flow is:

$$F(\vec{x}) = \frac{1}{Z(\vec{x})} Av + B\omega = \frac{1}{Z(\vec{x})} \begin{bmatrix} -1 & 0 & x \\ 0 & -1 & y \end{bmatrix} v + \begin{bmatrix} xy & -(1+x^2) & y \\ 1+y^2 & -xy & -x \end{bmatrix} \omega \quad (8)$$

$$Z(\vec{x}) = \frac{fb}{D(\vec{x})} \quad (9)$$

where x and y represent the coordinates in the normalized camera frame of the corresponding pixel, v and ω are the linear and angular velocity of the camera in the camera frame, $Z(\vec{x})$ is the depth in the camera frame, and a_v and a_ω are the 2-dimensional vectors corresponding to linear and angular velocity terms. f is the focal length of the camera and b is the baseline between the cameras. Note that, in this work, we estimate displacements rather than velocity, although we will keep the same notation for brevity. I.e. $v \times t$ and $\omega \times t$ instead of v and ω.

Using $F(\vec{x})$ and $D(\vec{x})$ from the network, this is a least squares problem for v and ω. We perform 3-point RANSAC by randomly sampling 3 points from $F(\vec{x})$ and $D(\vec{x})$, and solving for v and ω. The solution is then used to find the set of inlier points that best fit the estimated model. The threshold for inliers is set to be the absolute difference between the network and RANSAC flow estimates, in terms of pixels. Because the motion field equation above assumes a static scene, the computed inlier mask will automatically filter out independently moving (non-static) objects.

After performing RANSAC, we have an estimate of the camera velocity, computed from the inlier set of points. While the function to compute the inlier set is non differentiable, we can treat the inliers as a mask, and backpropagate through the computed velocity to the inliers from the flow and depth predictions from the network.

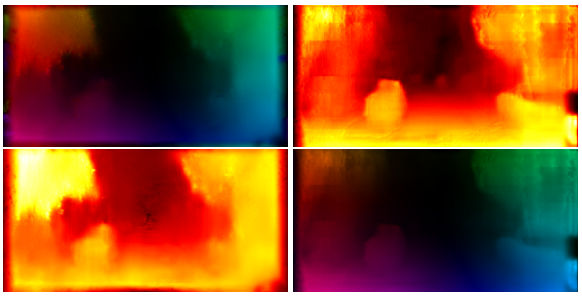

Figure 3. Using the RANSAC pose, we can use the predicted flow (top left) to estimate disparity (bottom left), and disparity (top right) to estimate flow (bottom right).

3.7. Refinement Loss

Using the pose estimate from RANSAC and the motion field equation in (8), we can estimate the optical flow from the disparities and pose using the forwards equation, as in Zhou et al. [29], but also the disparities using the flow and pose using the backwards equation derived from (8). To estimate disparity from flow and pose, we would like to find the disparities, $\hat{D}(\vec{x})$, that minimize $L(\vec{x})$:

$$L(\vec{x}) = F(\vec{x}) - \frac{\hat{D}(\vec{x})}{fb} Av - B\omega \quad (10)$$

This is equivalent to finding the scale between the vector v_1 and the projection of v_2 onto v_1:

$$\hat{D}(\vec{x}) = \frac{v_2^T v_1}{v_1^T v_1} \quad (11)$$

$$v_1 = A_v/fb, v_2 = F(\vec{x}) - B_\omega \quad (12)$$

Using these estimated flows and disparities, we can then compute the appearance loss, (1), for the other image sequence. That is, we can use the disparities estimated from flow to compute the appearance loss on the stereo pair, and the flow estimates from disparity on the temporal pair. These losses allow each network to learn from an additional pair of images. An example of these estimates can be found in Fig. 3

3.8. Network Architecture

One disadvantage of our method is that it requires dense per-pixel optical flow and disparity predictions from two networks in order to estimate pose, as compared to other networks that directly regress pose using a, potentially smaller, CNN. In order to compensate for this, we use a much smaller pair of networks in our pipeline than the standard works, such as [9, 15]. Our network is based on the Flownet-S architecture [4], but with a significantly smaller model. In summary, we remove the final three convolution layers from the encoder and the first two convolution

Sequence	00		01		02		03		04		05	
	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}
Before refinement	14.95	5.92	78.78	2.89	11.16	3.89	8.06	4.33	3.25	2.52	10.25	4.47
After refinement	5.12	1.87	72.98	2.55	6.01	1.91	7.61	5.19	2.70	0.76	6.02	2.29
Final model	4.56	2.46	78.98	3.03	5.89	2.16	6.84	2.42	9.12	1.42	3.93	2.09
DVSO [25]	0.71	0.24	1.18	0.11	0.84	0.22	0.77	0.18	0.35	0.06	0.58	0.22
SfMLearner [28]	66.35	6.13	35.17	2.74	58.75	3.58	10.78	3.92	4.49	5.24	18.67	4.10
UnDeepVO [13]	4.41	1.92	69.07	1.60	5.58	2.44	5.00	6.17	4.49	2.13	3.40	1.50
Zhan et al. [27]	-	-	-	-	-	-	-	-	-	-	-	-
Luo et al. [14]	-	-	-	-	-	-	-	-	-	-	-	-

	06		07		08		09		10	
	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}
Before refinement	11.58	2.52	14.68	8.44	10.26	4.53	12.32	5.00	11.37	4.10
After refinement	3.15	1.60	5.12	3.28	5.17	1.86	4.19	1.34	4.94	1.95
Final model	7.48	3.76	3.13	2.25	4.81	2.24	8.84	2.92	6.65	3.89
DVSO [25]	0.71	0.20	0.73	0.35	1.03	0.25	0.83	0.21	0.74	0.21
SfMLearner [28]	25.88	4.80	21.33	6.65	21.90	2.91	18.77	3.21	14.33	3.30
UnDeepVO [13]	6.20	1.98	3.15	2.48	4.08	1.79	7.01	3.61	10.63	4.65
Zhan et al. [27]	-	-	-	-	-	-	11.92	3.60	12.62	3.43
Luo et al. [14]	-	-	-	-	-	-	3.72	1.60	6.06	2.22

Table 1. Translation $t_{rel}(\%)$ and rotation $r_{rel}(^\circ/100\,\mathrm{m})$ RMSE on the 11 KITTI Odometry training sequences. We present our results from our ablation study (before and after refinement), trained on KITTI raw, as well as our final test model, trained only on the KITTI odometry sequences 00-08.

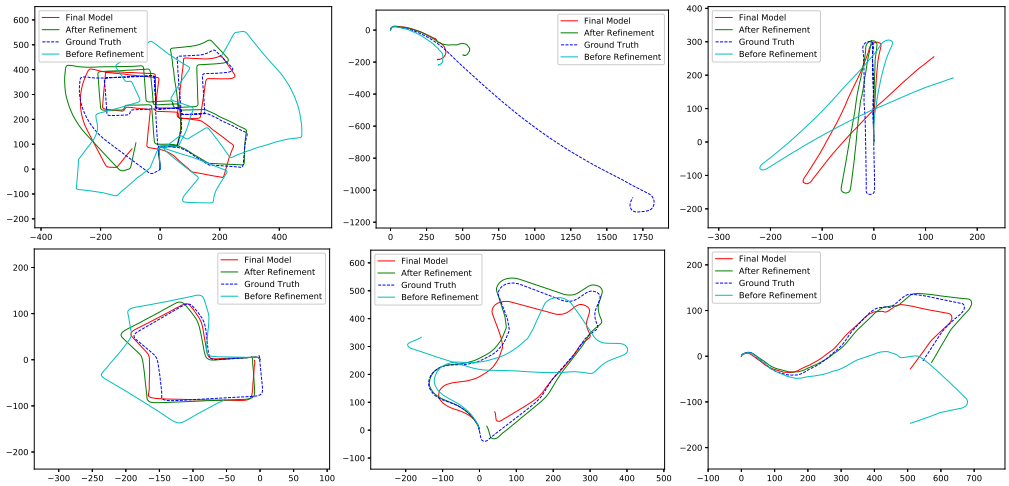

Figure 4. Selected trajectories of the models trained on KITTI raw before and after refinement, as well as our final model, against ground truth. Sequence numbers by from top left to bottom right are 00, 001, 06, 07, 09, 10.

layers from the decoder, and halve the number of output channels at each layer. In our experiments, our network reduces an input image with resolution 128x448 to activations with resolution 8x28 and 256 channels at the bottleneck. In Tab. 2, we compare the number of parameters in our network with several methods, including Monodepth [9], SFM-Learner [29], FlowNet [5], as well as the Resnet50 [10] variant from Monodepth. Our network is at least 10% of the size of these networks.

Method	# Parameters
Ours	2,943,496
Monodepth	31,596,936
SFM-Learner	33,187,568
FlowNet-S	35,885,068
Monodepth Resnet50	58,445,736

Table 2. Comparison of model size between our network and competing flow and depth architectures.

4. Experiments

We evaluate our structure on the KITTI Dataset [7] and make comparison to other state-of-the-art networks for depth, flow and pose estimation. In our ablation studies, we quantitatively compare the models before and after training with the proposed refinement loss terms in Sec. 3.7, and show that the refinement significantly improves the accuracy of the final pose estimates. We also show qualitative examples where the outlier mask is able to correctly segment independently moving objects as outliers.

4.1. Evaluation Details

For testing, we trained our final model on KITTI Odometry sequences 00 to 08 to observe the same splits used in other similar methods such as [29]. We trained the flow and depth networks for 15 epochs and started the refinement process on the full pipeline with RANSAC for another 15 epochs. We evaluate the final model's pose performance on the full training set of KITTI Odometry, depth on the KITTI Stereo 2015 Dataset and also the KITTI Raw with splits defined by Eigen et al. [2], and flow on KITTI Flow 2015. Note that in all comparisons our model is only trained on KITTI Odometry.

4.1.1 Depth Prediction

We evaluate our depth network prediction based on the metrics used in Eigen et al. [2]. We test on the same set of images according to the test split specified by [2], however our model is not trained on the training split but a split based on the KITTI Odometry Dataset. The Eigen split test set has around 670 images in total selected from KITTI Raw sequences. In addition, we also test our model on the Stereo 2015 set using the same metrics. We report both results and comparison to other methods evaluated according to Eigen splits and report the values as in Table 3. Note that UnDeepVO [13] results are from the model trained only from the KITTI Odometry Dataset and Luo et al. [14] is trained on their full stereo model. We can see that our stereo model performs well in general comparing to other state-of-the-art methods despite its size.

4.1.2 Flow Prediction

Our optical flow evaluation is performed on KITTI Flow 2015 training dataset. We have the ground truth from 200 images in the training set and we calculate the average End-point Error (EPE) as an error metric. We then calculate the percentage of outliers on the non-occluded(noc) area and occluded(occ) areas defined by KITTI Dataset [7]. We compare to the state-of-the-art pose network by Luo et al. [14] which incorporates both flow and depth networks into the pipeline. We see that our flow estimate is much worse than theirs. However, this is typically due to outliers where our network fails due to its small size, and are masked out by the RANSAC.

4.1.3 Pose Estimates

We compare our pose estimates results against ground truth on KITTI. DVSO [25] performs much better than us because they use a keyframe based windowed optimization structure and preserves the VIO framework, while our method is only doing pairwise image SFM and concatenate the results which produces large drift. The SfM-Learner [28], UnDeepVO [13], Zhan et al. [27] are methods with CNN pose network structure and they do not use flow, and we showed that our structure performs much better than theirs. Luo et al. [14] has a joint depth and flow networks like ours, and they produce a segmentation mask which functions similarly to RANSAC inlier mask and boosts their network performance. However our model is much smaller in size compared to theirs.

4.2. Ablation Studies

In this section we perform an ablation study of our model with and without the proposed fine-tuning, trained on the complete City, Residential, and Road sections of the KITTI Raw dataset [7]. For evaluation, we evaluate on the KITTI Odometry [8] and Stereo/Flow datasets [17, 16]. We first train the flow and disparity networks for 30 epochs until the training losses stabilize, before adding the proposed RANSAC optimization on top of the graph. We use the estimated pose to fine tune the network and train for another 20 epochs. We show that the fine-tuning process has a significant improvement on the odometry estimates, and small improvements in disparity and flow predictions.

4.2.1 KITTI Odometry Benchmark

Table 1 shows the odometry results of our pipeline with the network models trained before and after refinement on the KITTI Odometry training set. There are 11 training sets in KITTI Odometry with ground truth 6dof pose in total, and we evaluate the RMSE in translation and rotation, following [8]. The refinement process improves the disparity and flow

	RMSE	RMSE(log)	Abs Rel	Sq Rel	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Before refinement ablation (stereo2015)	5.490	0.171	0.108	1.919	0.897	0.966	0.987
After refinement ablation (stereo2015)	5.261	0.170	0.108	1.546	0.899	0.968	0.988
After refinement ablation (Eigen split)	3.663	0.184	0.117	0.696	0.877	0.959	0.983
Final model (stereo2015)	5.645	0.224	0.140	1.489	0.819	0.930	0.974
Final model (Eigen)	3.663	0.184	0.117	0.696	0.877	0.959	0.983
DVSO [25]	3.390	0.177	0.092	0.547	0.898	0.962	0.982
Garg et al. [6]	5.104	0.273	0.169	1.080	0.740	0.904	0.962
SfMLearner [28]	5.181	0.264	0.201	1.391	0.696	0.900	0.966
UnDeepVO [13]	6.57	0.268	0.183	1.73	-	-	-
Zhan et al. [27]	4.204	0.216	0.128	0.815	0.835	0.941	0.975
Luo et al. [14]	5.011	0.209	0.128	0.935	0.831	0.945	0.979

Table 3. Evaluation results on depth prediction with comparisons to other state of the art methods evaluated on the Eigen train/test split. Note, however, that our methods are trained on KITTI odometry and KITTI raw, rather than the usual Eigen train/test split. We also include results on KITTI stereo 2015.

Figure 5. An example of a failure case from sequence 01. From left to right the images concatenated are flow, inlier mask and disparity respectively. The network consistently underestimates the flow over the textureless road section, causing the RANSAC to select a large part of the sky in the inlier set.

	EPE		Error Perc	
	noc	occ	noc	occ
Before refinement	9.28	18.42	42.30%	50.10%
After refinement	9.72	19.02	41.11%	49.14%
Final model	11.75	21.98	45.20%	52.68%
Luo et al. [14]	3.86	14.78	-	-

Table 4. Evaluation results on KITTI Flow 2015 dataset on flow prediction.

outputs of the networks and clearly improves the odometry estimates of the pipeline by a large margin.

Fig. 4 shows the trajectories generated by the two models against ground truth on 6 of the 11 sequences. Note that sequence 01 has a much higher translation error than the other datasets. This is because this is the only sequence on a highway in a very open space without many other structures, as can be seen in Fig. 5. An example of a failure case from sequence 01. From left to right the images concatenated are flow, inlier mask and disparity respectively. The network consistently underestimates the flow over the textureless road section, causing the RANSAC to select a large part of the sky in the inlier set, and produce an underestimate of the translation.

4.2.2 Depth and Flow Prediction

We evaluate the effect of refinement on our disparity network through the accuracy of depth prediction on the metrics used in [2]. The results between the two models can be found in Tab. 3, where it can be observed that there is very little difference before and after the refinement. Note that, for the refinement comparison, our models are trained on KITTI Raw Dataset instead of the split specified by Eigen et al. [2]. We provide two results evaluated on KITTI Stereo 2015 using the same metrics, and the other of the refined model evaluated on the same test set of Eigen split, though trained on KITTI Raw.

Our flow results after the refinement is slightly worse than the original in terms of EPE and slightly better in the percentage of error pixels. We observed that the refinement improved the flow predictions within the inlier region, but did not improve pixels considered outliers, resulting in slightly worse results for those pixels.

4.3. Independently Moving Objects

Our RANSAC procedure automatically segments independently moving objects in the image. We show some examples of this segmentation in Fig. 6. As the outlier rejection is purely geometric, it does not rely on the network to learn that these objects have independent motion. In addition, the outlier masks show that RANSAC also segments parts of the scene where the network tends to struggle, such as strong shadows (second row), the sky (fifth row), thin structures (sixth row) and vegetation (seventh row).

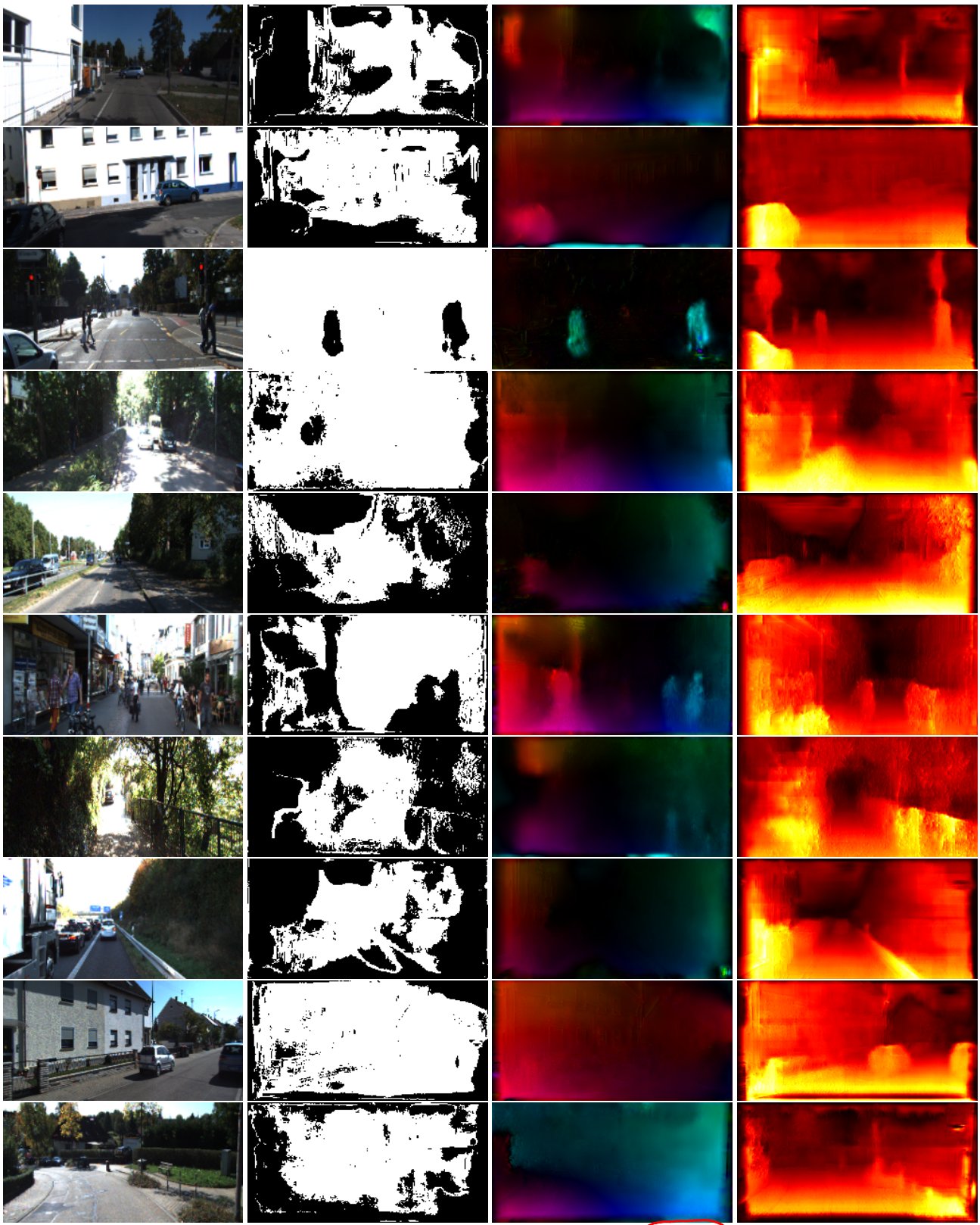

Figure 6. Interesting qualitative results from our pipeline. Left to right: Grayscale image, inlier mask, predicted flow, predicted disparity. The inlier mask demonstrates RANSAC's ability to remove areas over which the network has high uncertainty, such as vegetation and textureless regions such as the sky. In addition, it allows us to automatically remove independently moving objects from the scene. Note that in the fourth image, there is a person slight left of the center of the image. Best viewed in color.

黑色：不可信区
after RANSAC (sky, 行人)

	EPE		Error Perc.	
	noc	occ	noc	occ
Before refinement	5.17	7.49	**11.92**	7.74
After refinement	**3.79**	**4.92**	12.15	**4.75**

Table 5. Flow errors on the KITTI Flow 2015 sequences, before and after refinement with RANSAC. Errors are computed only over pixels considered inliers by RANSAC, which corresponds to roughly 64% of pixels.

	RMSE	RMSE (log)	Abs Rel	Sq Rel
Before ref.	6.77	**0.18**	**0.12**	3.11
After ref.	**6.62**	**0.18**	**0.12**	**2.77**

Table 6. Depth errors on the KITTI Stereo 2015 sequences, before and after refinement with RANSAC. Errors are computed only over pixels considered inliers by RANSAC, which corresponds to roughly 64% of pixels.

5. Discussion

From our experiments, we can see that our pose prediction results are competitive with most state of the art methods. Overall, DVSO clearly beats the other methods, but relies on a joint optimization scheme that allows for optimization over multiple long term keyframes. On the other hand, our method's pose predictions are generated by concatenating pose estimates between each pair of frames in each sequence, and so could see significant improvement with a similar optimization scheme. Amongst the other methods, we outperform the monocular only SFMLearner and the work by Zhan et al., are quite close to UnDeepVO, but do not achieve the same accuracy as Luo et al. However, this work takes as input images of size 256x832, and so receive a significantly larger amount of information to process.

In our ablations, we saw that our pose results improve significantly given the additional refinement, but we did not see a similar improvement in the flow and depth errors. To resolve these two conflicting sets of results, we additionally computed the flow and depth error metrics only over points considered inliers by RANSAC, which we show in Tables 5 and 6. From these results, we can see that our inlier flow errors improve significantly after refinement, with a more modest improvement in the depth errors. This follows from the formulation of our loss, as it only allows for backpropagation through the inlier points. However, while the errors over these inlier points have improved, the outlier points experienced worse errors, resulting in a lower than expected improvement in global errors. This is not an issue when the goal is pose estimation, and only the inliers are used to estimate pose, but in the future, we plan to explore better ways to combine these losses to improve the global errors further.

6. Conclusion

In this work, we demonstrate a novel unsupervised egomotion pipeline that combines unsupervised learning with geometric optimization. We show that our method is able to compete with state of the art methods which directly predict pose from a network, while providing extra guarantees about robustness and automatically detecting independently moving objects. We also show that this pipeline is able to achieve state of the art accuracy with a significantly reduced model, and hope that it can spur future work in bringing robustness and safety to learning for egomotion.

7. Acknowledgements

We gratefully appreciate support through the following grants: NSF-DGE-0966142 (IGERT), NSF-IIP-1439681 (I/UCRC), NSF-IIS-1426840, NSF-IIS-1703319, NSF MRI 1626008, ARL RCTA W911NF-10-2-0016, ONR N00014-17-1-2093, the Honda Research Institute and the DARPA FLA program.

References

[1] J. Delmerico and D. Scaramuzza. A benchmark comparison of monocular visual-inertial odometry algorithms for flying robots. *Memory*, 10:20, 2018. 1

[2] 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*, pages 2366–2374, 2014. 6, 7

[3] J. Engel, V. Koltun, and D. Cremers. Direct sparse odometry. *IEEE transactions on pattern analysis and machine intelligence*, 40(3):611–625, 2018. 2

[4] P. Fischer, A. Dosovitskiy, E. Ilg, P. Häusser, C. Hazırbaş, V. Golkov, P. Van der Smagt, D. Cremers, and T. Brox. Flownet: Learning optical flow with convolutional networks. *arXiv preprint arXiv:1504.06852*, 2015. 2, 4

[5] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 1, 4, 5

[6] R. Garg, V. K. BG, G. Carneiro, and I. Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *European Conference on Computer Vision*, pages 740–756. Springer, 2016. 2, 7

[7] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 6

[8] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 6

[9] C. Godard, O. Mac Aodha, and G. J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *CVPR*, volume 2, page 7, 2017. 1, 2, 3, 4, 5

[10] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 5

[11] M. Jaderberg, K. Simonyan, A. Zisserman, et al. Spatial transformer networks. In *Advances in neural information processing systems*, pages 2017–2025, 2015. 2

[12] J. Y. Jason, A. W. Harley, and K. G. Derpanis. Back to basics: Unsupervised learning of optical flow via brightness constancy and motion smoothness. In *European Conference on Computer Vision*, pages 3–10. Springer, 2016. 2

[13] R. Li, S. Wang, Z. Long, and D. Gu. Undeepvo: Monocular visual odometry through unsupervised deep learning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7286–7291. IEEE, 2018. 2, 5, 6, 7

[14] C. Luo, Z. Yang, P. Wang, Y. Wang, W. Xu, R. Nevatia, and A. Yuille. Every pixel counts++: Joint learning of geometry and motion with 3d holistic understanding. *arXiv preprint arXiv:1810.06125*, 2018. 2, 5, 6, 7

[15] S. Meister, J. Hur, and S. Roth. Unflow: Unsupervised learning of optical flow with a bidirectional census loss. *arXiv preprint arXiv:1711.07837*, 2017. 2, 3, 4

[16] M. Menze, C. Heipke, and A. Geiger. Joint 3d estimation of vehicles and scene flow. In *ISPRS Workshop on Image Sequence Analysis (ISA)*, 2015. 6

[17] M. Menze, C. Heipke, and A. Geiger. Object scene flow. *ISPRS Journal of Photogrammetry and Remote Sensing (JPRS)*, 2018. 6

[18] A. Ranjan, V. Jampani, K. Kim, D. Sun, J. Wulff, and M. J. Black. Adversarial collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation. *arXiv preprint arXiv:1805.09806*, 2018. 2

[19] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8934–8943, 2018. 1

[20] 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. 2

[21] C. Wang, J. Miguel Buenaposada, R. Zhu, and S. Lucey. Learning depth from monocular videos using direct methods. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2

[22] R. Wang, J.-M. Frahm, and S. M. Pizer. Recurrent neural network for learning densedepth and ego-motion from video. *arXiv preprint arXiv:1805.06558*, 2018. 2

[23] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 3

[24] N. Yang, R. Wang, J. Stückler, and D. Cremers. Deep virtual stereo odometry: Leveraging deep depth prediction for monocular direct sparse odometry. In *European Conference on Computer Vision*, pages 835–852. Springer, 2018. 2

[25] N. Yang, R. Wang, J. Stückler, and D. Cremers. Deep virtual stereo odometry: Leveraging deep depth prediction for monocular direct sparse odometry. *CoRR*, abs/1807.02570, 2018. 5, 6, 7

[26] Z. Yang, P. Wang, Y. Wang, W. Xu, and R. Nevatia. Every pixel counts: Unsupervised geometry learning with holistic 3d motion understanding. *arXiv preprint arXiv:1806.10556*, 2018. 2

[27] H. Zhan, R. Garg, C. S. Weerasekera, K. Li, H. Agarwal, and I. Reid. Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 340–349, 2018. 2, 5, 6, 7

[28] Y. Zhong, Y. Dai, and H. Li. Self-supervised learning for stereo matching with self-improving ability. *arXiv preprint arXiv:1709.00930*, 2017. 5, 6, 7

[29] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe. Unsupervised learning of depth and ego-motion from video. In *CVPR*, volume 2, page 7, 2017. 1, 2, 4, 5, 6