Fall 2021

Looking into the Future: Predicting Future Video Frames Using Monocular Depth Estimation and Egomotion

Meera Kumar
San Jose State University

LOOKING INTO THE FUTURE: PREDICTING FUTURE VIDEO FRAMES USING MONOCULAR DEPTH ESTIMATION AND EGOMOTION

A Thesis

Presented to

The Faculty of the Department of Computer Engineering

San José State University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

by

Meera Kumar

December 2021

The Designated Thesis Committee Approves the Thesis Titled

LOOKING INTO THE FUTURE: PREDICTING FUTURE VIDEO FRAMES USING MONOCULAR DEPTH ESTIMATION AND EGOMOTION

by

Meera Kumar

APPROVED FOR THE DEPARTMENT OF COMPUTER ENGINEERING

SAN JOSÉ STATE UNIVERSITY

December 2021

Mahima Agumbe Suresh, Ph.D.	Department of Computer Engineering
Magdalini Eirinaki, Ph.D.	Department of Computer Engineering
Wencen Wu, Ph.D.	Department of Computer Engineering

ABSTRACT

LOOKING INTO THE FUTURE: PREDICTING FUTURE VIDEO FRAMES USING MONOCULAR DEPTH ESTIMATION AND EGOMOTION

by Meera Kumar

Micro-mobility has become a growing market that has altered transportation within cities. While helping people reach their destination efficiently while using fewer fossil fuels and resources, e-scooters lack tailored safety protocol. Using depth and ego-motion machine learning estimation through a live video stream, we hope to identify possible oncoming hazard for e-scooter users. To approach this problem, we tested methods that removed the pose network with a scaling transformation which was derived via linear regression. Our intuition was that training and inference will be faster with the removal of the pose network and was validated by the results. We also found that forward warping has good accuracy using the transformed ground truth egomotion over the relative egomotion from the pose network. This discovery can be used to predict if an object within the scene has a probability of colliding with the e-scooter user.

ACKNOWLEDGMENTS

I would like to thank my advisor, Professor Mahima Suresh, for believing in me and guiding me through this thesis and masters program.

I am also grateful to the other professors whose classes I've taken at SJSU for helping me become a better machine learning engineer. Lastly, I would like to thank my family for supporting me and helping me pursue my goals.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 INTRODUCTION

Cities are seeing a growing trend with newer modes of transportation: micro-mobility alternatives that are lighter and intended for short trips. A visible manifestation of this trend is the introduction of electric scooters to the streets of major Bay Area cities. Micro-mobility is important in changing the way we move from one place to another while finding alternatives to cars. Not only do they harm the environment through the consumption of gasoline, cars cause many injuries and deaths through accidents. Streets with a lot of cars discourage people from going places on foot or other modes of transport. With micro-mobility people can move quickly within neighborhoods without having to use a car. Eventually street traffic could be changed in cities to accommodate pedestrians, bikers, and scooters, which are less likely to cause accidents and are better for the environment. Also, problems with the poor connectivity of public transport can be solved with micro-mobility options from a bus or train stop to people's final destination.

As a relatively new form of transportation to join the urban street space there are safety considerations to make for e-scooters. People may not know how fast they are going or have a good view of potential dangers. With the increasing number of e-scooter related accidents [1], a helmet equipped with cameras and inertial motion units or an app can improve rider safety. However, generating insights from a live video stream is challenging on resource constrained deceives. Since safety warnings need to be provided immediately, computationally intensive programs must be offloaded to an edge server. This assumes the micro mobility user has good network while moving. Alternate solution is building a lightweight model that runs quickly on low resource devices.

In order to adapt a machine learning model to function on a phone with limited internet connectivity, we will focus reworking a depth estimation model to run faster at low resource levels. A popular approach to self-supervised depth estimation is training two neural networks, one for depth and one for pose. The pose estimation aids in verifying

the quality of the depth estimation. With ground truth egomotion data available, we scale it to imitate the relative egomotion outputted by the pose network. The motivation for our work is the conjecture that including ground truth egomotion when calculating the loss of a depth estimation model will allow for computational efficient performance. Many models focus on being unsupervised in their training which is impressive but loses sight of end use of the model. By adding data that is easy to collect from a phone, we proceed in developing a model usable by an e-scooter rider. Depth estimation is useful for determining the distance of various objects from the rider. With the depth of objects in an image, we can predict if the rider is moving too close to an object or if an object is moving quickly in the direction of the rider. Our main contributions are the following:

- We propose using ground truth 6-DoF egomotion data in lieu of the pose network.
- We analyze our findings and investigate ways to include collision detection as the next level for this model using forward warp.

2 LITERATURE REVIEW

At the basis of this project is object detection through unsupervised depth estimation. Humans have spatial awareness so we understand how far something is from us or within an image. It is a fundamental way we interact with the world which is why much work has gone into depth estimation. Depth estimation can be applied in various fields such as robotics, self-driving cars, medical imaging, VR, and 3D modeling and reconstruction. Unlike humans, a device with a camera is incapable of automatically determining the 3D distance of objects around it just through video footage. Monocular depth estimation (MDE) is the task of estimating depth within an RGB image. With only one camera source (monocular), creative methods have to be implemented in order to extract depth information.

2.1 Depth Estimation in Humans

Our first exposure to depth estimation is in our own ability to perceive depth. Studies on the human vision system provide valuable insights on depth estimation since many machine learning approaches are derived from it. Studies in psychology and vision show that humans rely on binocular disparity and motion parallax in order to understand the depth of objects in a scene [2], [3].Motion parallax is when objects moving at a constant velocity appear to move slower the further away they are from the viewer [4] as seen in Fig. 1.

Another kind of parallax field is created with motion [2]. When an observer moves through a 3D environment while looking straight ahead a global optical flow is generated. The visual field appears, expands, and flows out of the periphery of the observer. However motion parallax does not help with depth estimation for independently moving objects. Instead using dynamic occlusion, objects moving independently in the field cause motion discontinuities that provide relative depth between the moving object and the background as seen in.

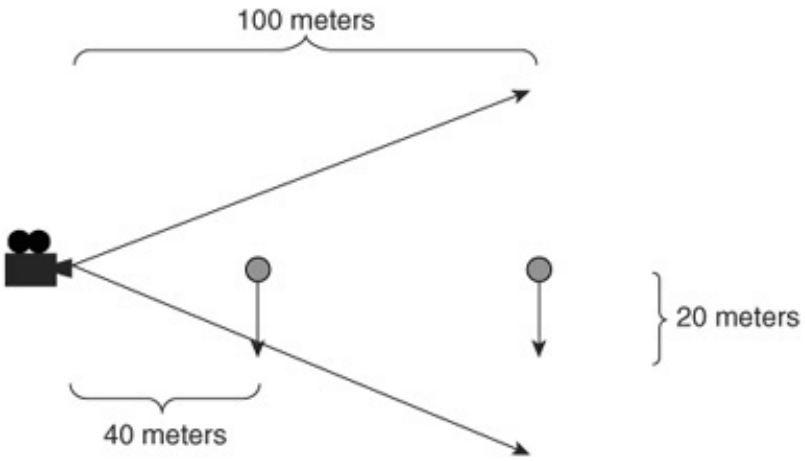


Fig. 1: Motion parallax diagram. The object at distance 40 m moves out of the viewer's field of view where as the object at 100 m moves only 25% across the field of view. This gives the illusion that the object at 40 m is moving faster than the object at 100 m even through they are moving at the same speed.

Binocular disparity is the positional difference between two cameras or retinal projects from a given point in space. Our visual cortex triangulates the difference between each retinal view to perceive depth. This was first discovered in the human vision system through Wheatstone's invention of the stereoscope in 1838. As shown in Fig. 2, object near will be in the distance between the eyes and the point of convergence. We can also sense an object is near through the contractions in extra-ocular muscles when focusing on objects close to us due to the increase in inward rotation.

Other monocular depth cues are static pictorial cues. Occlusion occurs when closer objects partially obscure objects further away. Perspective is the effect that an object becomes smaller as it gets closer to the viewpoint. Humans can perceive texture gradients where objects closer have more details (varying markings and projected shapes) than objects further away. For example when looking at sand, it looks smooth from a distance but up close you can see the different sands mixed together. Object identification and understanding the relative size of the object allows for depth perception through familiarity.

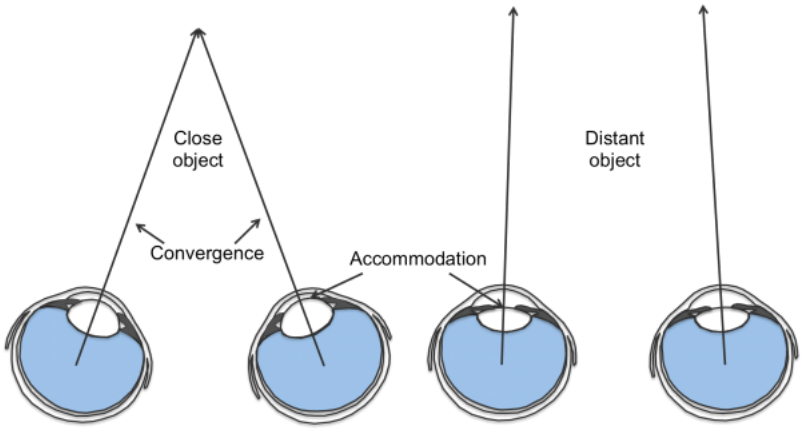


Fig. 2: Binocular disparity. On the left, the line of vision with each eye converges to aid in depth perception of an object near the viewer. Objects further away are outside the point of convergence of the eyes which makes precise depth estimation challenging.

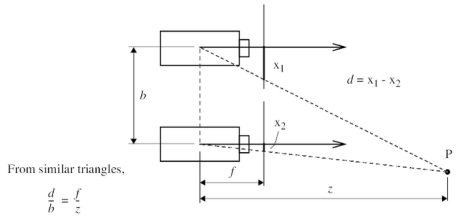
While perspective is visible even in static images, the kinetic depth effect is the phenomenon of perceiving 3D form from a 2D image when the object is moving. Kinetic depth is also known as structure from form [5].

The key points of human vision deployed in machine learning are binocular disparity in stereo approach and kinetic depth and motion parallax in monocular approaches.

2.2 Multi Stereo Depth Estimation

Classical depth estimations were made on multi view systems. This method is still used for reliable 3D modeling. A depth map can be computed from each viewpoint using binocular stereo similar to the human visual system. With epipolar geometry and stereo matching, images of the same scene can be rectified to imitate the way human eyes see only one view with both eyes. Analogous to binocular disparity as seen in Fig. 3, the disparity between the rectified images from two views can be used to calculate depth. Szeliski first proposed an algorithm for computing the depth estimations or motion maps for each image in a multi view system rather than a single depth map [6]. This method helped develop view interpolation where a model can estimate the depth or location of an object in occluded areas with the depth of the known views. Another application is

motion compensated frame interpolation which is based on motion parallax. A frame of the video feed can be predicted bidirectionally (from the previous and next video frame).



(a) Stereo geometry using binocular disparity by pixel matching.

(b) Disparity representation from Kitti.

Fig. 3: Binocular Disparity. [7]

2.3 Monocular Depth Estimation

Monocular depth estimation gained traction due to its lower level of requirements for data collection. Binocular vision remains more accurate for near range depth estimation, but monocular depth estimation relying on motion parallax outperforms for distant features [8]. Intuitively, small errors in the triangulation and angle estimation translate to large errors as the distance of the object from the camera increases. Also stereo data demands higher memory and processing requirements of the system. Using machine learning, monocular depth estimation can be regarded as a continuous regression problem between image pairs. The task is to learn the non-linear mapping of the space in the image to a depth map.

2.3.1 Supervised Single Image Depth Estimation

Supervised models rely on ground truth depth measurements for training. Some approaches to supervised models include using a MRF framework, boosting classifier, and CNNs.

Early approaches to supervised learning relied on Markov random field to infer depth cues and the relationship between features in order to recreate images as 3d models [9].

An non explicit approach was to use geometric cues as features in order to make 3D reconstructions from a single image [10]. There are also successful deep learning methods that regress depth from a single image using stacked neutral networks [11], DCNNs with spacing-increasing discretization [12], and deep convolutional neural fields [13]. However these supervised models need ground truth depth maps for hundreds of thousands of images which is challenging to acquire. To work around this, synthetic datasets have been designed for training. With proper image style transfer and adversarial training, a model trained on synthetic data can predict pixel level depth on single images [14].

2.3.2 Self-supervised Single Image Depth Estimation

Attaining ground-truth depth measurements is a costly task. Supervised methods need multiple observations of a scene or depth maps. Deep-Stereo, an image synthesis network, replicates multi-stereo data. The relative pose of multiple cameras is used to generate new scenes from neighboring frames [15]. Still this method requires some multi view images which is not suitable for monocular depth estimation. The lack of scalable data collection limits the practicality of supervised depth estimation.

In comparison, unsupervised models can run on video or static image sequences without extra labeling with comparable results (Fig. 4). Data can be collected through a calibrated camera rig to provide some ground truth. Calibrated rigs provide ground truth for pose. Another method uses unconstrained camera movement. Camera movement provides more dynamic movement between frames where the model can use the change in position to determine the depth of objects in the frame.

Inspired by autoencoders, Garg introduced a semi supervised framework that uses pairs of images from a calibrated rig [16]. The resulting stereo image pairs have known transformations which can be used to calculate depth. This was furthered with the development of a fully differentiable, unsupervised CNN with better loss functions [17]. In these methods, depth estimation is viewed as an image reconstruction problem from

sequential frames. A disparity map is estimated through constructing rectified images using the left RGB, I_l. The disparity map is used to synthesize one image by applying an inverse warp on the other image. The difference between the synthesized and real image is used to constrain the training process.

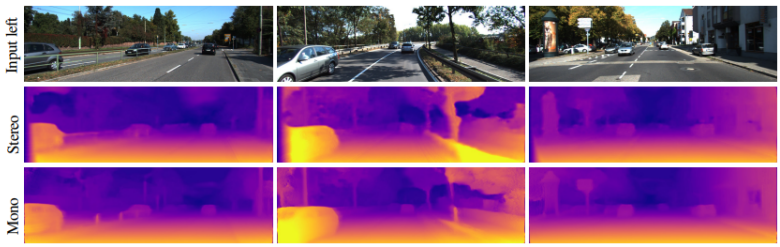


Fig. 4: Comparison between stereo and monocular depth prediction. [17]

2.3.3 Multi-frame Monocular Depth Estimation

Calculating depth from monocular video brings further challenge. Previously mentioned methods fail to handle object movements in dynamic scenes. To deal with moving objects, initial works relied on separately trained optical flow models [18], [19]. Better results have been found by eliminating learned flow and using motion segmentation based on scene geometry and egomotion [20].

DeMon builds on unsupervised depth estimation by using the concept of stereo pairs on two unconstrained video frames [21]. Through a chain of encoder-decoder networks, they estimate depth and egomotion jointly. This model has partial supervision through optical flow data. Stereo cameras are not always feasible in real world application, but treating adjacent frames in a video feed as stereo pairs of the same scene similar geometric calculations can be done.

By applying left-right video frame consistency [17] the accuracy between the estimated depth and pose can be verified by the real frame. To deal with motion, an explainability mask is predicted with pose and depth in order to discount regions that are

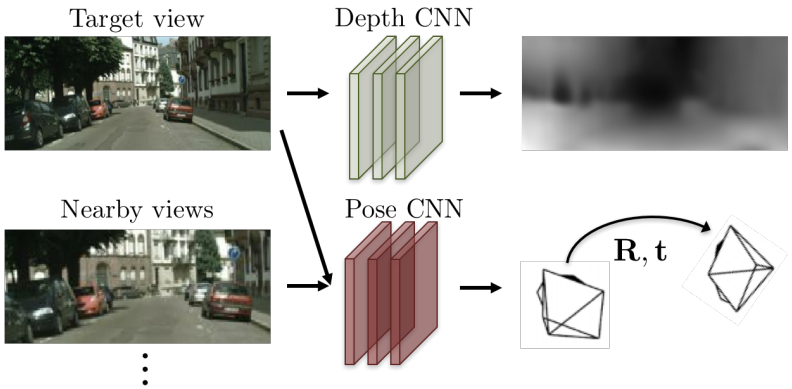
not static. This method suffers from per-frame scale ambiguity and isn't consistent over long videos. SfM-Net uses geometry to segment frames and predict depth and rotation of objects [22]. This model also allows for supervision of the nets through egomotion (camera motion) and depth (RCBD sensors) data.

SFMLearner is a fully unsupervised network that predicts depth and egomotion estimation from monocular video input [23]. They added an egomotion network to existing depth networks in order to predict pose between consecutive video frames. In Fig. 5, the pose CNN is shown to calculate the disparity between the target view and adjacent views very much like in human vision and stereo views. The dual networks generate a novel view from a video sequence. Given source views, I_{t-1} and I_{t+1}, at different time steps, a learnt transformation from the pose network is applied to the images to warp them to the target view, I_t. Using a spacial transformer network, the warped view synthesis is compared to the known image at I_t in order to self-supervise the model.

vid2depth builds on SfMLearner by altering the loss functions from 2D to 3D. This scales the geometry between frames appropriately allowing for more consistent depth over longer videos [24]. This is achieved through addition of Iterative Closest Point (ICP) loss shown in Fig. 6. ICP loss predicts the rigid transformation between the point clouds between pairwise frames thereby trying to match the objects between frames. Newer works still use a combination of depth and pose networks with slightly modified frameworks or loss functions [25], [26].

Taking depth estimation models to mobile devices requires a balance of speed and accuracy given the computation, memory, and power limitations. Encoder-decoder models based on mobilenet-v3 [27] and a teacher student network based on MobileNet and ResNetSt-101 [28] have shown great promise. Combining sensors with various lightweight models has also shown good results for image segmentation and classification [29].

(a) Training: unlabeled video clips.



(b) Testing: single-view depth and multi-view pose estimation.

Fig. 5: SFMlearner overview. [23]

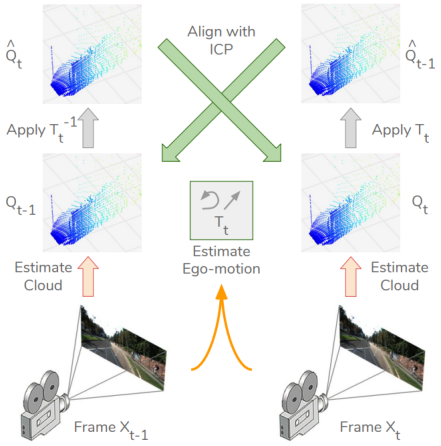


Fig. 6: vid2depth 3D loss. [24]

Our work builds upon the work initiated in vid2depth to show that some supervision using data easily extracted from a phone can improve training and inference times. The

original model has some set up to extract raw pose data so we complete that feature and added loss functions.

3 PROBLEM FORMULATION

Unsupervised depth estimation training time and performance can be improved by incorporating orthogonal sources of data such as inertial navigation data within the network.

3.1 Pose Network

In the original work of vid2depth, dual networks as shown in Fig. 5 are used. The depth network takes just the target view ,I_t, as input while the pose network takes the target view and the adjacent source views at time steps, $t+1$ and $t-1$. The results of the two networks are used to reconstruct one of the frames to check the accuracy of the depth and egomotion estimations. Despite being jointly trained, the depth and the pose estimation model are disconnected. They can be used independently during test-time inference. This structure allows for edits to either of the networks without changing the other. The pose network consists of 7 stride 2 convolutions followed by a 1x1 convolution which results in 6D relative egomotion vectors between the target view and all the source views.

For a low resource device such as a phone, running one neural network can be resource consuming. With our end goal of using depth estimation for crash collision detection for a user on the go, we view the pose network to be a cumbersome portion of the model. It would require memory space for 7 layers of parameters, outputs, and the network structure. Using a linear regression eliminates the need to store sequences of video frames and convolution layers. We only need the ground truth 6D egomotion and the coefficients from the linear regression to transform the input into the relative egomotion between frames. The other advantage to this approach is removing the need for the frame at the next time step when making depth and forward warp predictions.

3.2 Ground Truth Egomotion Transform

When a frame is defined by metadata, the coordinates are in world space. The IMU provides the exact speed, GPS coordinates, and rotational movements of the vehicle at time t. The target view inputted into the model is the camera's coordinate system. Fig. 7 shows the coordinate system of the OXT (IMU device) and the camera's.

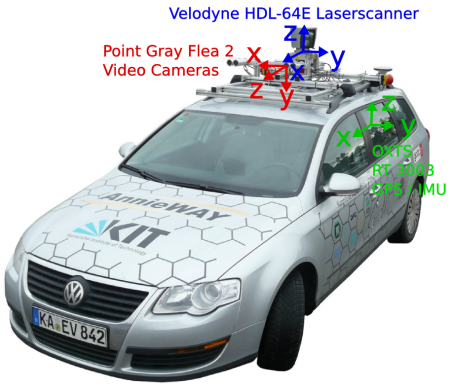


Fig. 7: Sensor set up used for Kitti Dataset. Velodyne and IMU are in world space whereas the camera is in camera space. [30]

To get the data for the matrix, the rotational values, roll, pitch, yaw have to be expanded into their orthogonal unit vectors. This expansion is done using the Euler and Tait-Bryan rotations, given the angle or rotation from the 3 axis. In Section 4.1, we discuss how the homogeneous camera matrix transform to go from the world space to the camera space. This is an affine projection which preserves the lines in the image but not necessarily the angles and the perspective. So even with these precise rotational calculations of the pose, there is no guarantee the angles are precisely translated into the camera space.

To complete the projection the homogeneous matrix is multiplied by the camera's intrinsic matrix. The intrinsic matrix contains information of the focal length, image sensor formation, and principal point. With this ground truth egomotion now projected into the camera space, we need the relative pose. The relative pose is the change in pose

between the target frames and the source frames. It is calculated through the dot product of the inverse pose at the source view and the pose at the target view. In our approach the source frames are the frames adjacent to the target view. Once again, we run into the need for the frame in the next time step if we use the matrix rotations to get the relative egomotion between frames.

In general, projecting 3D views from 2D images is an ill posed problem. As discussed above the camera and inertial navigation device are on different planes. Within a camera, there are variables that affect the target view such as focal length. The focal length proportionally scale the points on the image plane. Therefore it will always be challenging to retrieve the scaling of an image without ground truth data on the egomotion or continuous focal length. As shown in Appendix A, different basis for the ground truth result in variations in the relative egomotion. As there is no concrete method to achieve this, we hope to test using a linear regression to quickly calculate relative egomotion.

While linear regression takes some time to set up and train prior to training the model, in implementation we remove the need for multiple images as an input and decrease the number of matrix operations at run time.

4 METHODOLOGY

Structure from motion (SfM) is a standard in self supervised depth estimation. Many models are based on the joint unsupervised networks of depth and egomotion [23], [24]. The slight change in pose between video frames allows for depth understanding using disparity. The pose network outputs the relative pose between two images that is verified by geometric constraints on the image pairs. The predicted depth is reprojected onto a target frame after being warped by the predicted relative pose. The difference between the predicted and actual frame is measured to correct the depth network.

4.1 Ground Truth Egomotion

The pose network predicts egomotion as 6-DoF transformation matrices which combine the translational and rotational motion of the scene. To imitate extraction of pose information from a cell phone's gyroscope, we utilize the meta data for each frame recorded by the inertial navigation system from the Kitti dataset. The IMU provides position, orientation, and velocity data as a ground truth reference. There are two things to consider in using the ground truth egomotion. Firstly the camera is looking along the positive Z axis and the Y axis is up. By transforming with the vector $[0,0,1]$ you can get the camera's viewing vector in the world space which is relevant to the e-scooter rider.

Secondly the pose calculated by the pose network is not the absolute pose in the scene but the relative pose between adjacent frames. To calculate the relative pose using ground truth data, the roll, pitch, and yaw are expanded using the basic rotation transforms. The dot product of the rotations around the 3 axis results in the rotation matrix for the specific frame. This matrix is combined to form the 4×4 homogeneous pose matrix:

$$T_t = \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{1}$$

The relative pose between the target frame I_t and the previous frame I_{t-1} is then $T_{t-1}^{-1} \cdot T_t$. In order to speed up calculations during training and inference, we hypothesized that we can find an approximate transformation using linear regression which will scale the ground truth egomotion to match the relative egomotion.

4.1.1 Scale Egomotion

Experiments show that dual pose depth networks tend to underestimate shifts in rotational motion [31]. The true shift in the camera is not fully reflected in the estimated depth map. Furthermore, a scaling factor was added to the pose estimate for better training of the depth network. In order to have a comparable study between removal of the egomotion network on the depth network, we scaled the egomotion meta data to match the predicted egomotion as modeled in Fig. 8.

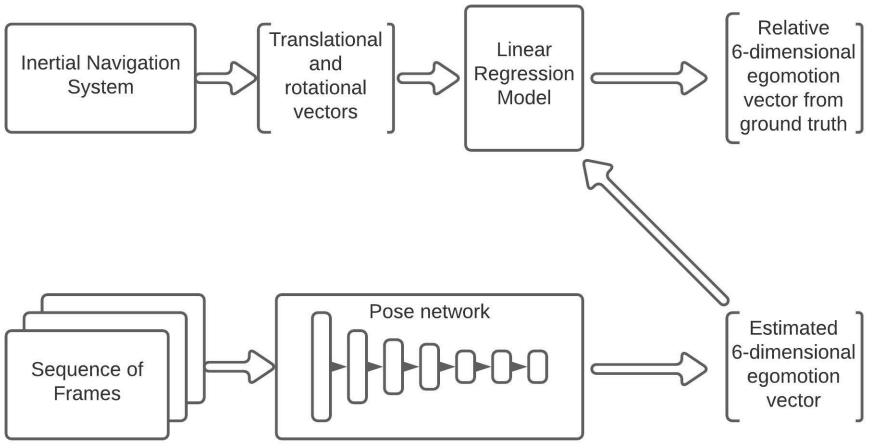


Fig. 8: Linear Regression training using vid2depth relative egomotion.

Using the inference model from the original paper and 462 samples, we gathered a dataset of predicted egomotion. The inference provides the relative egomotion forwards between I_t and I_{t+1} and backwards between I_t and I_{t-1} from the target frame. Fig. 14 and Fig. 15 in Appendix B show why forward and backwards time step were treated as

separate calculations rather than negating the forward time step to get the backwards time step.

A separate scaling matrix was derived using linear regression for the the forward and previous relative pose. The linear regression did transform the ground truth rotations close to the relative rotation as seen in Fig. 9.

4.1.2 Incorporate Egomotion into Model

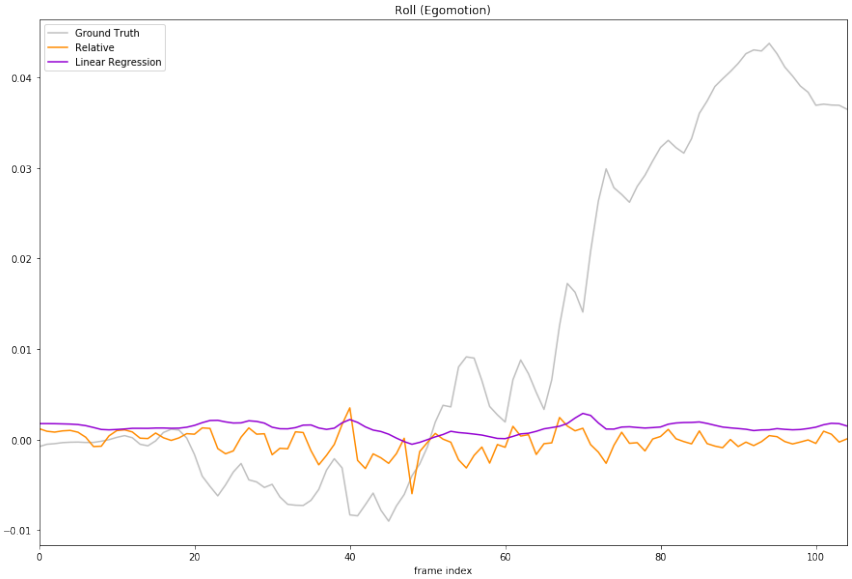
The strength of vid2depth is the depth model and the pose estimations model can be used independently during inference. Even during training the pose network and depth network feed forward separately but their results are combined for novel view generation. Fig. 10 shows how we deal with pose and depth prediction. The true 6D egomotion is scaled via the linear regression transformation and read directly into the loss functions discussed in Section 4.2.

4.2 Model Overview

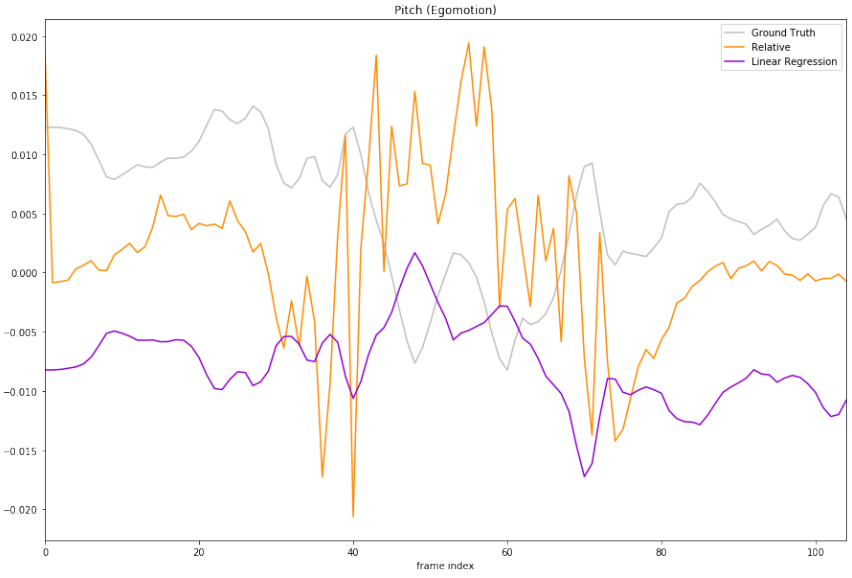
The basic method of vid2depth is to learn depth and egomotion through training on a single monocular video stream and self supervise through . As discovered [17], [21], frames close to each other from a free moving video feed can be used as stereo pairs. Given a pair of consecutive frames I_{t-1} and I_t, the model estimates depth at D_{t-1} and D_t and the egomotion at T_t. The egomotion at T_t represents the camera's movement from time $t-1$ to t. In order to apply 3D loss, the pixel level depth at t is projected into a point cloud Q_t. This transformation requires the camera's intrinsic matrix, K, and the egomotion, T_t.

$$Q_t^{ij} = D_t^{ij} \cdot K^{-1}[i, j, 1]^T \tag{2}$$

To generate a novel view, $\hat{X}_t$, pixel coordinates need to be mapped between X_t and X_{t-1}. Since the camera is moving between frames, some of the pixels will be moved out



(a) Roll



(b) Pitch

Fig. 9: Comparison between ground truth pose, relative pose, and relative pose derived from linear regression.

of view while new features will enter the frame. vid2depth has consideration to the removal of these pixels during train because experiments show that keeping points that leave the frame during forward egomotion, degrades training over time. A mask for the

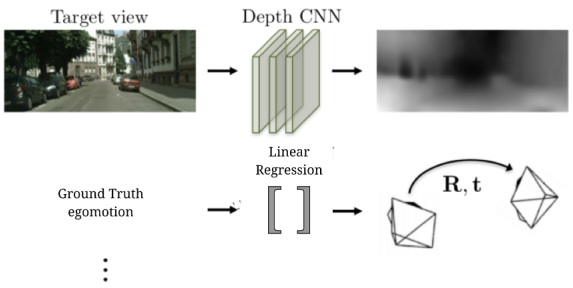


Fig. 10: Reworked architecture for single view depth and multi-view egomotion estimation.

changing border of a scene can be determined computationally using the relative depth and egomotion from the networks. This leads for a robust 3D comparison between the generated view and the ground truth view.

4.2.1 Loss Functions

The generated novel views are compared to the original frames to produce a differential reconstruction loss.

$$L_{rec} = \sum_{ij} \| (I_t^{ij} - \hat{I}_t^{ij}) M_t^{ij} \| \tag{3}$$

M_t is the derived principle mask, to eliminate pixels that are in I_{t-1} but move off the screen at I_t. Regardless this loss function isn't enough because frame generation can't account for dramatic changes in lighting. It is hard for this loss function alone to determine the quality of the depth estimation without resulting in artifacts. A structure similarity and deep smoothness loss are added [17]. Structure similarity index is a common metric used in image prediction by applying simple pooling to compute the means and standard deviations on the original and projected image.

$$L_{SSIM} = \sum_{ij} [1 - SSIM(I_t^{ij}, \hat{I}_t^{ij})] M_t^{ij} \tag{4}$$

The smoothness loss is used to regularize the depth estimates. Using the gradients of the depth map, this loss functions helps the depth network factor sharp changes in the depth at the pixel level.

Finally, 3D loss is factored in by comparing the point clouds $\hat{Q}_t$ to Q_t. The iterative closest point computes the transformation that minimizes the point to point distance between corresponding pixels. Given the estimated depth at a time step, a point cloud Q_t can be calculated as shown in equation 2. The point cloud projection is notably dependent on the estimated egomotion. The iterative closest point (ICP) will then show the shortest distance between the $\hat{Q}_t$ and Q_t. That ICP transform is used to adjust the egomotion estimation. Any residual errors after adjusting the egomotion are errors with the depth estimation. The impact of depth errors on egomotion and vice versa is ignored which allows for easier replacement of the pose network with the linear regression.

4.3 Forward Warp

Given the relative camera movement from t to $t+1$, as the estimated egomotion at time step t, the point cloud at that time step can be transformed to get an estimate for the next frame: $\hat{Q}_{t+1} = T_t Q_t$. This method is based on vid2depth's inverse warp to calculate ICP loss. This transformation combined with Eq. 2 allows for generation of the view at I_{t+1}. The mapping allows for the reconstruction of I_{t+1} by warping I_t based on the estimated depth and egomotion:

$$\hat{I}_{t+1}^{ij} = I_t^{ij}[i,j,1]^T = KT_{t+1}(D_t^{ij} \cdot K^{-1}[i,j,1]^T) \tag{5}$$

As a result current frame information can be used to predict the next frame in a sequence with the monocular video feed and some ground truth or inference data.

5 EXPERIMENTS

5.1 Dataset

We used the KITTI dataset [30] for training and evaluation. Many depth prediction models use this dataset as a benchmark which makes it convenient for comparisons. The dataset comes with monocular video stream with various metadata such as 3D point clouds, sceneflow, and odometry. While collecting monocular video footage, they used an intertial measurement unit to track ground truth movement of the vehicle. From this we have translational and rotational data to infer ground truth egomotion. We used 462 frames for training the transformation from the linear regression and 375 frames for testing the forward warp. Inference tests were run on one Intel Xeon E5-2660 v4 compute core. Training was done on a NVIDIA P-100 GPU.

5.2 Evaluation of Linear Regression

Transforming world space egomotion into the camera space seems to be effectively achieved with linear regression. Table 1 shows the mean absolute error between the egomotion predicted by the pose network and the ground truth pose scaled by the transformation derived through a linear regression. Ideally, the egomotion between frame I_{t+1} to I_t is the negative of I_t to I_{t+1}. But the results show that there are some variation between the relative pose going to the previous frame vs the relative pose going to the next frame. This could be errors from the vid2depth pose estimation.

Table 1
MAE of egomotion predicted by linear regression

6D egomotion	towards I_{t-1}	towards I_{t+1}
X	0.0006	0.0005
Y	0.0003	0.0002
Z	0.0084	0.0079
roll	0.0009	0.0014
pitch	0.0045	0.0048
yaw	0.0014	0.0015

A major part of removing the pose network was to scale ground truth egomotion to match the model's estimated egomotion. Using the linear regression relative egomotion then decreased the time it took to train and run inference. Table 2 shows that using linear regression to train and infer the depth significantly reduces time. In all the trial runs, training with the scaled egomotion had 27% lower run time compared to training with both neural networks. The inference also shows 34% decrease in time.

Table 2
Time comparison between pose network and linear regression method

Method	Training	Inference
with pose network	18.4 hrs	18.86 ms/frame
with linear regression	13.5 hrs	12.47 ms/frame

5.3 Evaluation of Forward Warp

Table 3 compares the run time and accuracy metrics between the forward warp results from using the relative egomotion from the vid2depth pose network and the relative egomotion from the linear regression transformation. This was tested over using the depth estimated using vid2depth inference over a sequence of 375 frames. Time of forward warp is in milliseconds per frame. The input image size was 128 x 416 and trained on 1 Intel Xeon E5-2660 v4 compute core (codename Broadwell, 2.0GHz, 35M Cache, 128 GB RAM). In order to find results close to a low resource device, GPUs were not used. Even so the amount of ram in a single compute node is much higher than even high end smart phones that have about 6-8 GB RAM.

Table 3
Forward warp evaluation metrics

Egomotion Source	Dataset	Time	Abs Rel	Sq Rel	RMSE	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
vid2depth inference	Kitti	43	7.6988	**2.1803**	**7.2653**	0.5633	0.7510	0.8352
linear regression	Kitti	29.83	**7.0965**	2.1881	7.4814	**0.5030**	**0.7163**	**0.8120**

There is a 30.6% decrease in forward warp time per frame with the linear regression. The total time shown includes the time for inference of the egomotion and pose to reflect

a real time situation. Noting that the time between frames is about 1 ms, processing even at 29 ms is much too slow for real time forward warping.

A visual comparison between the pose network based forward warp and the linear regression forward warp (Fig 11) shows that the linear regression forward warp has more definition when the camera is moving and still. It can also anticipate some movement from objects within the scene. The top row is the input frame at I_t that was warped into the results in the last row. The target frame, I_{t+1} is shown in the middle.

(a) Using pose network relative egomotion

(b) Using linear regression relative egomotion

Fig. 11: Comparison between ground truth pose, relative pose, and relative pose derived from linear regression.

6 DISCUSSIONS AND CONCLUSIONS

In this work, our goal was to test the hypothesis that ground truth egomotion can (i) improve the training and inference time of monocular depth estimation; (ii) project the current frame to the next and obtain a sneak-peek of the future scene. For our application, we need a fast and efficient way to do it. The following were some observations and directions for future work we came up with.

Linear regression is a simple and suitable model to transform the ground truth motion vectors from the world space into the coordinate system of the camera and the image. However, based on our observation, we hypothesize that the errors from the vid2depth model can propagate as our linear regression training is based on the original pose network's output. One issue with the vid2depth method is the lack of a consistent scaling for the pose estimation which causes errors over long video sequences. Even in our own graphs of the ground truth and relative egomotion, we could see the pose network's prediction drift from the ground truth egomotion. This makes gathering enough datapoints with low error for the regression challenging. It is up to future work to test this hypothesis.

Given the success of the regression model when applied to forward warping, we propose for the future to perform the regression from the pose network result to the real motion vectors. This could be useful to project the true motion to an e-scooter rider and provide situational awareness. How we project this is also a potential future work as we would have to handle the natural ambiguity of scenes with motion.

Looking into the future can be applied in other domains as well. e.g., filling gaps in surveillance videos when their views are blocked by obstacles as the stillness of the camera is more forgiving to slower run times. Also moving towards adding more metadata such as camera pose at the start of a scene can provide better depth estimation in variable situations.

In conclusion, the work uncovers several potential directions to improve rider safety by providing situational awareness on the streets. Further improvements to the linear regression architecture via different dataset and slight changes to the loss function could bring good results at low speeds.

Works Cited

[1] A. Y. Bresler, C. Hanba, P. Svider, M. A. Carron, W. D. Hsueh, and B. Paskhover, "Craniofacial injuries related to motorized scooter use: A rising epidemic," *American Journal of Otolaryngology*, vol. 40, pp. 662–666, Sept. 2019.

[2] E. J. Gibson, J. J. Gibson, O. W. Smith, and H. Flock, "Motion parallax as a determinant of perceived depth," *Journal of Experimental Psychology*, vol. 58, pp. 40–51, July 1959.

[3] M. F. Bradshaw and B. J. Rogers, "The interaction of binocular disparity and motion parallax in the computation of depth," *Vision Research*, vol. 36, pp. 3457–3468, Nov. 1996.

[4] H. von Helmholtz, *Helmholtz's treatise on physiological optics*, vol. 1, trans from the 3rd German ed.,. Optical Society of America, 1924.

[5] H. Wallach and D. N. O'Connell, "The kinetic depth effect," *Journal of Experimental Psychology*, vol. 45, pp. 205–217, Apr 1953.

[6] R. Szeliski and S. B. Kang, "Shape ambiguities in structure from motion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 506–512, May 1997.

[7] S. Arjomand Bigdeli, G. Budweiser, and M. Zwicker, "Temporally coherent disparity maps using crfs with fast 4d filtering," *IPSJ Transactions on Computer Vision and Applications*, vol. 8, Dec 2016.

[8] M. Mansour, P. Davidson, O. Stepanov, and R. Piché, "Relative importance of binocular disparity and motion parallax for depth estimation: A computer vision approach," *Remote Sensing*, vol. 11, no. 17, 2019.

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

[10] D. Hoiem, A. A. Efros, and M. Hebert, "Geometric context from a single image," in *10th IEEE International Conference on Computer Vision*, pp. 654–661, 2005.

[11] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in *28th Annual Conference on Neural Information Processing Systems 2014*, vol. 3, pp. 2366–2374, NIPS, Jan 2014.

[12] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao, "Deep ordinal regression network for monocular depth estimation," in *2018 Conference on Computer Vision and Pattern Recognition*, pp. 2002–2011, IEEE/CVF, June 2018.

[13] F. Liu, C. Shen, G. Lin, and I. Reid, "Learning depth from single monocular images using deep convolutional neural fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, p. 2024–2039, Oct 2016.

[14] A. Atapour-Abarghouei and T. P. Breckon, "Real-time monocular depth estimation using synthetic data with domain adaptation via image style transfer," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2800–2810, IEEE/CVF, 2018.

[15] J. Flynn, I. Neulander, J. Philbin, and N. Snavely, "Deepstereo: Learning to predict new views from the world's imagery," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5515–5524, June 2016.

[16] R. Garg, V. Kumar, G. Carneiro, and I. Reid, "Unsupervised CNN for single view depth estimation: Geometry to the rescue," in *European Conference on Computer Vision*, Lecture notes in computer science, pp. 740–756, Cham: Springer International Publishing, 2016.

[17] C. Godard, O. M. Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, CVPR 2017, (Honolulu, HI), July 2017.

[18] Z. Yin and J. Shi, "Geonet: Unsupervised learning of dense depth, optical flow and camera pose," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1983–1992, IEEE/CVF, 2018.

[19] Z. Yang, P. Wang, Y. Wang, W. Xu, and R. Nevatia, "Lego: Learning edge with geometry all at once by watching videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 225–234, IEEE/CVF, 2018.

[20] V. Casser, S. Pirk, R. Mahjourian, and A. Angelova, "Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos," vol. 33, pp. 8001–8008, AAAI Press, July 2019.

[21] B. Ummenhofer, H. Zhou, J. Uhrig, N. Mayer, E. Ilg, A. Dosovitskiy, and T. Brox, "Demon: Depth and motion network for learning monocular stereo," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5622–5631, IEEE/CVF, 2017.

[22] S. Vijayanarasimhan, S. Ricco, C. Schmid, R. Sukthankar, and K. Fragkiadaki, "Sfm-net: Learning of structure and motion from video," *ArXiv*, vol. abs/1704.07804, 2017.

[23] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised learning of depth and ego-motion from video," in *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pp. 6612–6619, IEEE/CVF, 2017.

[24] R. Mahjourian, M. Wicke, and A. Angelova, "Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints," in *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pp. 5667–5675, IEEE/CVF, 2018.

[25] J. Wang, G. Zhang, Z. Wu, X. Li, and L. Liu, "Self-supervised joint learning framework of depth estimation via implicit cues," *ArXiv*, vol. abs/2006.09876, 2020.

[26] J. Watson, O. M. Aodha, V. Prisacariu, G. J. Brostow, and M. Firman, "The temporal opportunist: Self-supervised multi-frame monocular depth," in *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pp. 1164–1174, IEEE/CVF, 2021.

[27] Z. Zhang, Y. Wang, Z. Huang, G. Luo, G. Yu, and B. Fu, "A simple baseline for fast and accurate depth estimation on mobile devices," in *Conference on Computer Vision and Pattern Recognition Workshop*, pp. 2466–2471, IEEE/CVF, 2021.

[28] Y. Wang, X. Li, M. Shi, K. Xian, and Z. Cao, "Knowledge distillation for fast and accurate monocular depth estimation on mobile devices," in *Conference on Computer Vision and Pattern Recognition Workshop*, pp. 2457–2465, IEEE/CVF, 2021.

[29] J. K. Mahendran, D. T. Barry, A. K. Nivedha, and S. M. Bhandarkar, "Computer vision-based assistance system for the visually impaired using mobile edge artificial

intelligence," in *Conference on Computer Vision and Pattern Recognition Workshop*, pp. 2418–2427, IEEE/CVF, 2021.

[30] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *International Journal of Robotics Research*, 2013.

[31] T. Van Dijk and G. De Croon, "How do neural networks see depth in single images?," in *Proceedings of International Conference on Computer Vision*, ICCV, pp. 2183–2191, IEEE, 2019.

Appendix A

COMPARISON OF GROUND TRUTH EGOMOTION

We calculated the exact displacement of the camera between frames using the longitude, latitude, and altitude from the metadata. As seen in Fig. 12, the units for distance is very large and attempts to scale it down to match the relative translational movement with a linear regression were unsuccessful. Fig. 13 had values that transformed easily to match the relative egomotion.

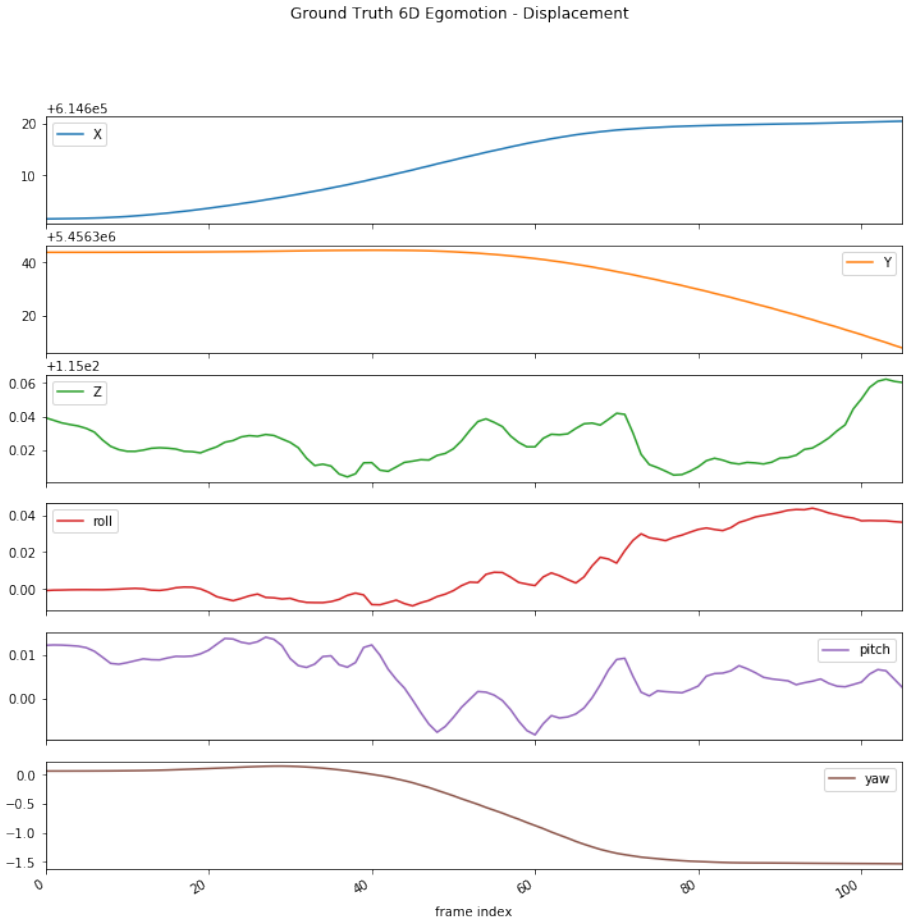


Fig. 12: 6D Egomotion with distance for translational values.

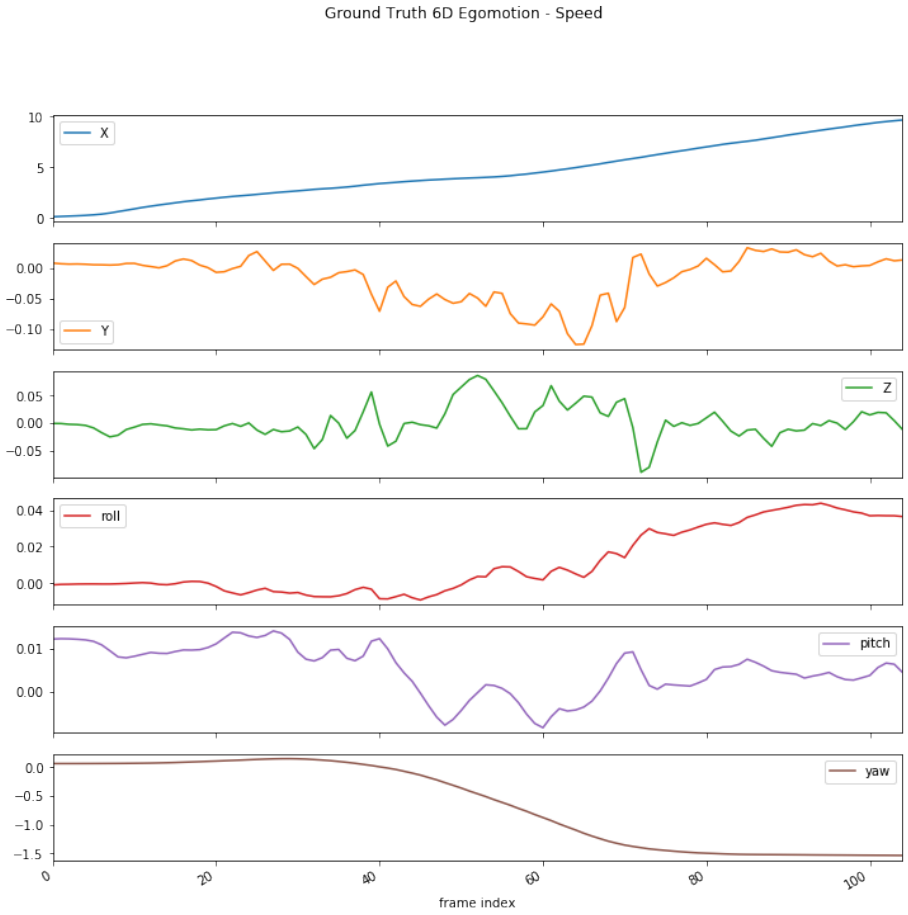


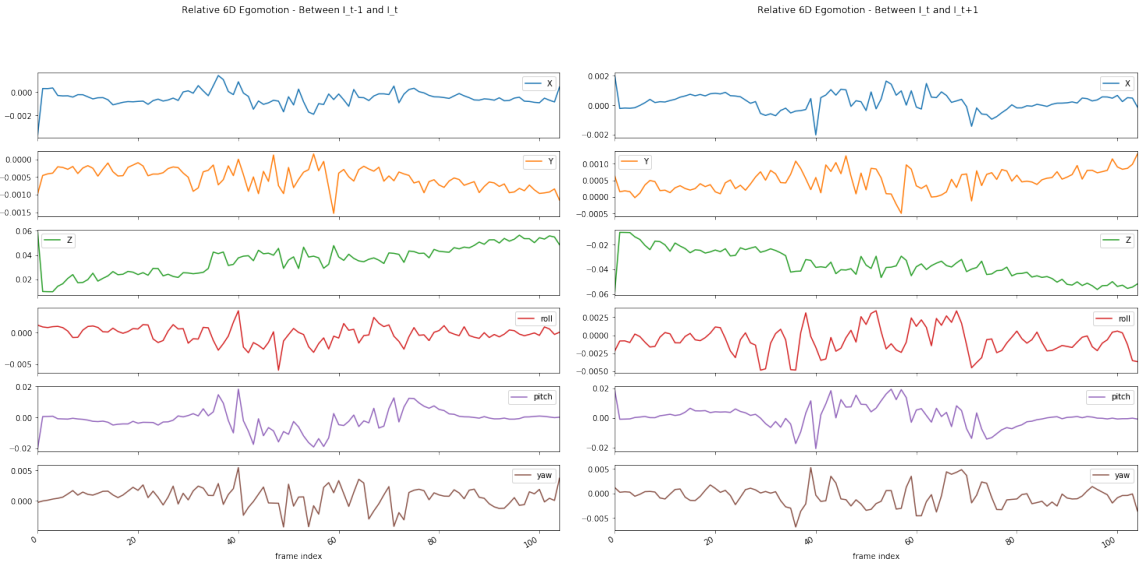
Fig. 13: 6D Egomotion with speed for translational values.

Appendix B

COMPARISON OF RELATIVE 6-DOF EGOMOTION

Realistically the 6D egomotion to the previous time step would be the negative of the current time step's egomotion. When the positive and negative time step are graphed next to each other in Fig. 14 and Fig. 15, it is apparent that there are slight differences in the forward and backwards relative egomotion.

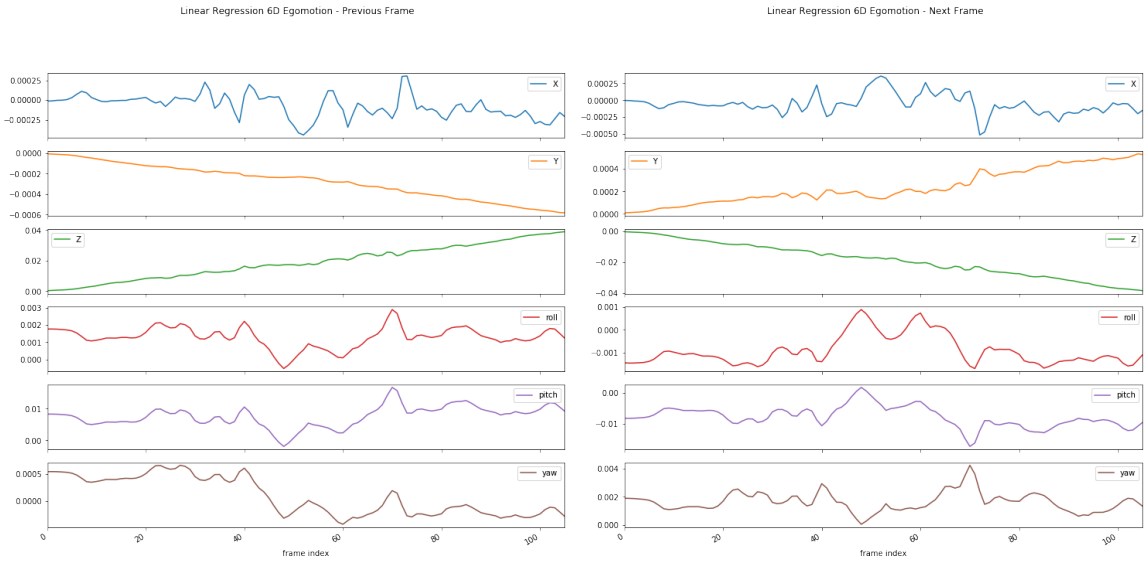
While the relative egomotion from the liner regression transformation is smoother than that from the pose network, the overall shape of the dimension over time match.



(a) 6D Egomotion in the negative timestep.

(b) 6D Egomotion in the positive timestep.

Fig. 14: From Pose Network



(a) 6D Egomotion in the negative timestep.

(b) 6D Egomotion in the positive timestep.

Fig. 15: From Linear Regression