KFNet - Point Cloud estimation for camera localization

Gokul Hari
Masters of Engineering, Robotics
University of Maryland, College Park
Maryland, USA
hgokul@umd.edu

Sameer Pusegaonkar
Masters of Engineering, Robotics
University of Maryland, College Park
Maryland, USA
sameer@umd.edu

***Abstract*—Visual Odometry is an important computer vision problem which deals with estimating the camera pose with respect to each & every frame in the video feed. We refer the paper KFNet, which proposes a neural network architecture in the context of Bayesian learning by embedding a structure based camera pose estimation technique into the Kalman filter within a deep learning framework. Specifically, the paper focuses on estimating 3D point clouds and solves for the camera poses. We provide a comprehensive explanation of all the stages of the problem with more importance to the controls aspect(state-estimation), the constraints & derive all the equations utilized. We verify & compared the results provided in the paper. In addition, to reject outlier estimated states, we also replaced a static threshold with an auto-thresholding mechanism, and we utilised this mechanism for better visualization of the states. We then conclude, by providing how results were obtained and instructions to replicate them.**

***Index Terms*—Computer Vision, Camera localization**

I. Introduction

Camera Relocalization is the process of estimating a camera pose in the form of a 6x1 vector. This vector is essentially consists of x, y, z, roll, pitch & yaw for localizing the camera in the world coordinate frame. This problem, approached when only using images is termed as visual odometry.

The visual odometry problem has been addressed in deep learning in three broad categories.

- Absolute Pose Regression: This was one of the earliest methods [1] [2] to model the camera pose estimation problem. However, this has been later realised to be not very accurate or generalizable, since modelled as an image retrieval proble,
- Relative Pose Regression: This approach involves regressing relative poses, ie, linear and angular velocities. Though this approach was shown to be more generalizable, it involves a image retrieval procedure that needs to match query images from a database, hence it is slower.
- Structure based approaches: This method involves 3D geometry, by transforming 2D image pixels to 3D points in world coordinate frame, with the help of additional information like depth/scene coordinates, and helps in solving for the poses using PnP algorithm [3]. This method is known to be highly accurate and has been the go-to approach in solving pose estimation problem, which will be considered in this study.

The best performing structure based approaches involves the major challenge of estimating intermediate variables like *depth* and *optical flow* [5]. Inspired by how human visual cortex processes depth information with using visual cues from the environment, deep learning frameworks are usually involved *in estimating depth/scene coordinates*.

Optical flow (pixel to pixel matching between images I_t and I_{t+1}) estimation involves brightness and smoothness assumptions (called as the correspondence problem). Recent deep learning frameworks learn to estimate optical flow very accurately and are capable of generalizing very well. However, these systems involves static scene assumption, and are likely to fail in the presence of dynamic objects in the scene.

Both scene coordinate regression (SCoRe) and optical flow regression problems involve approximations and assumptions that can be violated, resulting uncertain predictions. To address this KFNet formulates a Bayesian learning approach to regress uncertainty maps along with these scene coordinates and optical flow predictions, and are used in a probabilistic, Kalman filter based loss function.

The structure of this paper can be visualized as 3 subsystems below:

The Measurement System: Consists of a network called SCoordNet to directly derive a maximum likelihood estimations of the scene coordinates for a single image.

The Process System: Consists of a network called OFlowNet that models optical flow based on transitions for image pixels across time steps which is used to obtain prior predictions of scene coordinates. Here we obtain the aforementioned 2D-3D geometric transformation called the scene coordinates. To elaborate, given a 2D image, scene coordinate is a discriminative prediction for where the point imaged at pixel *i* lives in the 3D scene co-ordinate frame.

The Filtering System: This combines both the predictions & gives us a maximum posteriori estimations of the scene coordinates.

The above structure is resembles the methodology of a typical Kalman Filter. In the next section we describe the general background Bayesian intuition on approaching this problem.

II. Background

Existing research in camera pose estimation has been employing deterministic approaches. KFNet utilises a probabilistic structure based approach of Bayesian deep learning. In this section we will derive the Bayesian intuition behind state estimation.

When we are estimating a state x_t, we formulate the belief of the current state at time t as,

$$Be(x_t) = P(x_t \mid u_1, z_1, ...u_t, z_t) \tag{1}$$

given all the measurements $(z_1, z_2..z_t)$ and inputs $(u_1, u_2..u_t)$ upto time t. Using the Bayes rule, in equation 31 in Appendix IX-A, we can arrive at the relationship,

$$\begin{aligned} &P(x_t \mid u_1, z_1, ...u_t, z_t) \\ &= \eta P(z_t \mid x_t, u_1, z_1, ...u_t) P(x_t \mid u_1, z_1, ...u_t) \end{aligned} \tag{2}$$

where, η is a normalization constant. Based on the Markovian assumption in equation 34 explained in Appendix section IX-B, we get,

$$\begin{aligned} &P(x_t \mid u_1, z_1, ...u_t, z_t) \\ &= \eta P(z_t \mid x_t) P(x_t \mid u_1, z_1, ...u_t) \end{aligned} \tag{3}$$

In $P(x_t \mid u_1, z_1, ...u_t)$, using equation 33 in IX-A, we can introduce x_{t-1} term as,

$$\begin{aligned} &P(x_t \mid u_1, z_1, ...u_t) = \\ &\int \eta P(x_t \mid u_1, z_1, ...u_t, x_{t-1}) P(x_{t-1} \mid u_1, z_1, ...u_t) dx_{t-1} \end{aligned} \tag{4}$$

which means that the probability of the state x_t, given all prior measurements and inputs is equal to the integral with respect to prior state, for an integrand. The integrand is a product of the probability of state x_t, given the previous state at *t-1*, measurements and inputs upto time *t*, and the probability of previous state x_{t-1}, given all measurements and inputs upto time *t*.

By applying the Markovian assumption in 35 in appendix IX-B, we can write,

$$\begin{aligned} &P(x_t \mid u_1, z_1, ...u_t) = \\ &\int \eta P(x_t \mid u_t, x_{t-1}) P(x_{t-1} \mid u_1, z_1, ...u_t) dx_{t-1} \end{aligned} \tag{5}$$

Also, the state at x_{t-1} does not depend on the input u_t in the Markov model, hence

$$\begin{aligned} &P(x_t \mid u_1, z_1, ...u_t) = \\ &\int \eta P(x_t \mid u_t, x_{t-1}) P(x_{t-1} \mid u_1, z_1, ...z_{t-1}) dx_{t-1} \end{aligned} \tag{6}$$

The term $P(x_{t-1} \mid u_1, z_1, ...z_{t-1})$ is nothing but the prior belief of the state x_{t-1}.

$$\begin{aligned} &P(x_t \mid u_1, z_1, ...u_t) = \\ &\int P(x_t \mid u_t, x_{t-1}) Be(x_{t-1}) dx_{t-1} \end{aligned} \tag{7}$$

Substituting equation 7 in equation 3, we get

$$\begin{aligned} &P(x_t \mid u_1, z_1, ...u_t, z_t) = \\ &\eta P(z_t \mid x_t) \int P(x_t \mid u_t, x_{t-1}) Be(x_{t-1}) dx_{t-1} \end{aligned} \tag{8}$$

which can be given as,

$$Be(x_t) = \eta P(z_t \mid x_t) \int P(x_t \mid u_t, x_{t-1}) Be(x_{t-1}) dx_{t-1} \tag{9}$$

This derivation gives the intuition behind kalman filters.

III. Bayesian Formulation

Given a stream of RGB images upto time *t*, $\mathcal{I}_t = I_1, I_2, ...I_t$, we want to predict the latent state called the scene coordinate map, denoted as $\theta_t \in \mathbb{R}^{N \times 3}$, where N is the number of pixels. We can model this as a Gaussian variable

$$\theta_t^+ = (\theta_t \mid \mathcal{I}_t) \sim \mathcal{N}(\hat{\theta}_t, \Sigma_t) \tag{10}$$

where $\hat{\theta}_t$ is expectation and Σ_t is covariance. By bayes rule, the posterior probability θ_t is given as

$$P(\theta_t \mid \mathcal{I}_t) \propto P(\theta_t \mid \mathcal{I}_{t-1}) P(\mathcal{I}_t \mid \theta_t, \mathcal{I}_{t-1}) \tag{11}$$

This is formulated based on the intuition from equation 9.
1) $P(\theta_t \mid \mathcal{I}_{t-1})$ is the prior belief about the scene coordinate map, computed based on the optical flow estimated from the previous sequence of images $\mathcal{I}_{t-1}$. This prior belief is prone to uncertainties introduced by dynamic object in the scenes that violate the static assumption made in this process. Therefore we model the scene coordinate map θ_t along with a Gaussian noise component, w_t. This is given in the process equation below.

$$\theta_t = \mathbf{G_t} \theta_{t-1} + w_t \tag{12}$$

Here, $\mathbf{G_t} \in \mathbb{R}^{N \times N}$ is the state transition matrix, and it is a function of optical flow correspondences between images $I_{t-1} \rightarrow I_t$. The process noise modelled as a Gaussian $w_t \sim \mathcal{N}(0, W_t)$, has co-variance given by $W_t \in \mathbb{S}_{++}^N$. Note that $\mathbb{S}_{++}^N$ is an N- dimensional positive definite matrix space.
The prior estimated scene coordinate map from the image sequence I_{t-1} is denoted as,

$$\theta_t^- = (\theta_t \mid \mathcal{I}_{t-1}) \sim \mathcal{N}(\hat{\theta}_t^-, R_t) \tag{13}$$

where $\hat{\theta}_t^- = G_t \hat{\theta}_{t-1}$, and $R_t = G_t \Sigma_t G_t^T + W_t$ are the expectation and covariance of the Gaussian θ_t^-.

2) $P(I_t \mid \theta_t, \mathcal{I}_{t-1})$ models the likelihood of I_t computed from θ_t using a nonlinear function $h(\cdot)$ and this relationship is given as $I_t = h(\theta_t)$. This means, that the 3D scene

coordinates θ_t can be warped and projected on the image plane to obtain I_t, using the function $h(\cdot)$. This is done using the θ_t measurements obtained from the scene coordinate regressor. However, the regressed measurements will contain uncertainties which is modelled as a Gaussian noise v_t. We formulate a measurement equation given as

$$z_t = \theta_t + v_t \tag{14}$$

where $v_t \sim \mathcal{N}(0, V_t)$, with measurement noise co-variance $V_t \in \mathbb{S}^N_{++}$. Here, z_t is nothing but the noisy scene coordinate measurements. We can rewrite this likelihood as $P(z_t \mid \theta_t, \mathcal{I}_{t-1})$.

3) Reformulating likelihood measurement with noise :

Let e_t be the error between the prior belief from process system and the measurements of the state θ_t, such that $e_t = z_t - \hat{\theta}_t^-$ and can be expanded as,

$$e_t = z_t - \hat{\theta}_{t-1} \tag{15}$$

Since G_t (state transition matrix using optical flow) and $\hat{\theta}_{t-1}$ (belief at t-1) are known, observing z_t can be considered equivalent to observing e_t. So the likelihood $P(z_t \mid \theta_t, \mathcal{I}_{t-1})$ can be reformulated as $P(e_t \mid \theta_t, \mathcal{I}_{t-1})$. Using 14 and 15, thus we can write

$$e_t = \theta_t - \hat{\theta_t}^- - v_t \tag{16}$$

Note that equation 16 is a sum of Gaussians and hence, the error is also a Gaussian variable given as,

$$(e_t \mid \theta_t, \mathcal{I}_{t-1}) \sim \mathcal{N}(\theta_t - \hat{\theta_t}^-, V_t) \tag{17}$$

such that the expectation $(\theta_t - \hat{\theta_t}^-)$ is the difference between the state t, or the posterior probability and prior belief. The variance (V_t) is the same as the measurement noise's variance. Hence, equation 11 is modified to represent the posterior probability distribution as,

$$P(\theta_t \mid \mathcal{I}_t) \propto P(\theta_t \mid \mathcal{I}_{t-1}) P(e_t \mid \theta_t, \mathcal{I}_{t-1}) \tag{18}$$

4) Computing Kalman Gain :

The posterior probability is a Gaussian resulted by fusing the prior and likelihood Gaussians. Lets denote this bi-variate Gaussian distribution as

$$\left[\begin{pmatrix} \theta_t \\ e_t \end{pmatrix} \mid \mathcal{I}_{t-1} \right] = \mathcal{N} \left[\begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix} \right] \tag{19}$$

where μ_1 and μ_2 are the expectation while, $\begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix}$ is the co-variance matrix.

From equations 54 and 56 in appendix, we obtained the expectations $\mu_1 = \hat{\theta}_t^-$, $\mu_2 = 0$ and variances $\Sigma_{11} = R_t$, $\Sigma_{12} = \Sigma_{21} = R_t$, $\Sigma_{22} = V_t + R_t$. If the error e_t is made as the conditioning variable, then we can arrive at the posterior distribution as

$$(\theta_t \mid e_t, \mathcal{I}_{t-1}) \sim \mathcal{N}(\hat{\theta}_t, \Sigma_t) \sim \mathcal{N}(\hat{\theta}_t^- + K_t e_t, R_t(I - K_t)) \tag{20}$$

where $K_t = \frac{R_t}{V_t + R_t}$ is the kalman gain. Equation 10 is same as the posterior distribution θ_t^+ Check Appendix IX for the derivation.

Thus, $(e_t \mid \theta_t, \mathcal{I}_{t-1})$ that is, the error distribution given the state θ_t and image sequence $\mathcal{I}_{t-1}$. An overall summary of the variables and notations used in the bayesian formulation is described in image 1. The block diagram of our deep learning framework under study is shown in 2

Module	inputs	outputs	
The process system	$\hat{\theta}_{t-1}$ Σ_{t-1} I_{t-1} I_t	G_t W_t $\hat{\theta}_t^- = G_t \hat{\theta}_{t-1}$ $R_t = G_t \Sigma_{t-1} G_t^T + W_t$	- transition matrix - process noise covariance - prior state mean - prior state covariance
The measurement system	I_t	z_t V_t	- state observations - measurement noise covariance
The filtering system	$\hat{\theta}_t^-$ z_t R_t V_t	$e_t = z_t - \hat{\theta}_t^-$ $K_t = \frac{R_t}{V_t + R_t}$ $\hat{\theta}_t = \hat{\theta}_t^- + K_t e_t$ $\Sigma_t = R_t(I - K_t)$	- innovation - Kalman gain - posterior state mean - posterior state covariance

Fig. 1. Variables and notations in bayesian formulation

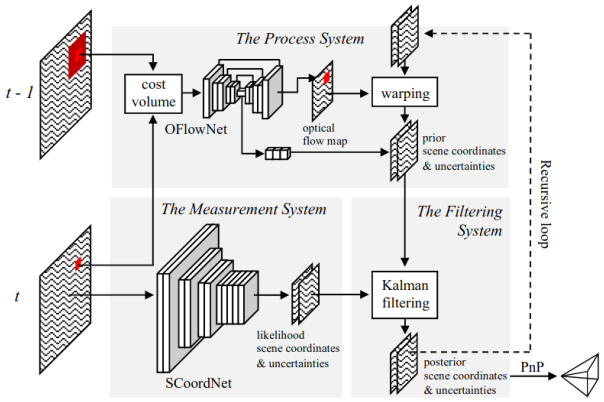

Fig. 2. Overall architecture of the KFNet frame work with the process, measurement and the filtering system

IV. Measurement System

The measurement system is similar to the process in the SCoRe task. It consists of how the observations are generated from the latent scene coordinates. The architecture of SCoordNet is described below along with uncertainties to model the measurement noise. After this a probabilistic loss is defined based on the likelihood of the measurement system.

A. Architecture

SCoordNet is primarily a fully convolutional network. The parameters of the network are lightweight. Specifically, it has

1/8 the parameter size as original SCoordNet. SCoordNet consists of 12 convolutional layers, Each neuron in the layers has a ReLU activation function except the last one. A stride of 2 is used to downsize the input image by a factor of 8. The output results in a channel of 4, where 3 include the 3-d scene coordinates and 1 includes a uncertainty measurement.

B. Loss

The latent scene coordinates $\theta_{(i)}$ for a given pixel p_i should follow the distribution $\mathcal{N}(z_{(i)}, {v_{(i)}}^2\mathbb{I}_3)$. A negative logarithm of the density function has been taken which leads to the maximum likelihood (ML) estimation for each pixel.

$$\mathcal{L}_{likelihood} = \sum_{i=1}^{N} \left(3logv(i) + \frac{\left\| z_{(i)} - y_{(i)} \right\|_2^2}{2v(i)^2} \right) \tag{21}$$

Here y is the ground-truth label for $\theta_{(i)}$. To add uncertainty measurements we use: $s_{(i)} = log{v_{(i)}}^2$ This uncertainty loss is added to quantify the prediction errors arising from the defined model. This is typically prominent at the boundary with depth discontinuity which is hard to model. SCoordNet suffers from this issue and hence is it evident to add a uncertainty measurement to reduce such errors during training.

V. Process System

The process systems consists of utilizing the pixel transition from time *t-1* to *t*. OFlowNet is used to predict the optical flows and process noise co-variance jointly for each pixel.

A. Architecture

OFlowNet mainly consists of a cost volume constructor and a flow estimator. First the features are extracted from 2 input images. The output feature maps have a spatial size of 1/8 of the input size. A cost volume is built for each pixel of the feature map so that:

$$C_i(o) = \left| \frac{F_t(p_i)}{\left\| F_t(p_i) \right\|_2} - \frac{F_{t-1}(p_i + o)}{\left\| F_{t-1}(p_i + o) \right\|_2} \right| \tag{22}$$

here F is the feature map. A L2 normalization to the feature map has been applied along the channel dimension before differentiation.

The flow estimator works over the cost volumes for flow inference. A U-Net architecture has been used with skip connections. The output of this network is a confidence map for each pixel. This map has been passed through a differentiable spatial softmax operator to compute the optical flow.

Therefore:

$$\hat{o} = E(o) = \sum softmax(f_o).o \tag{23}$$

where f_o is the confidence. A fully connected layer is added after the bottleneck of the UNet to regress the logarithmic variance.

B. Loss

After producing the optical flows, the state transition matrix G_t is obtained. The coordinate map and uncertainty map from time t-1 to t have been warped through bilinear warping. The prior coordinates of pixel p_i denoted by θ_i^- will follow the distribution:

$$\theta_{(i)}^- \sim \mathcal{N}(\hat{\theta}_{(i)}^-, {r_{(i)}}^2\mathbb{I}_3) \tag{24}$$

A negative logarithm of the PDF has been taken to model the process loss as:

$$\mathcal{L}_{prior} = \sum_{i=1}^{N} \left(3logr_{(i)} + \frac{\left\| \hat{\theta}_{(i)}^- - y_{(i)} \right\|_2^2}{2r_{(i)}^2} \right) \tag{25}$$

The loss definition uses the prior distribution of the scene coordinates to provide a weak supervision for training the OFlowNet.

VI. Filtering System

The filtering system fuses both the measurement and process systems together to get the posterior estimation. Note that the measurement system derives the likelihood estimation and the process system derives the prior estimation of the scene coordinates θ_t.

A. Loss

We know that, at time t and for a particular pixel p_i, the likelihood estimation is given by $\mathcal{N}(z_{(i)}, {v_{(i)}}^2\mathbb{I}_3)$ whereas the prior distribution is given by $\mathcal{N}(\hat{\theta}_{(i)}^-, {r_{(i)}}^2\mathbb{I}_3)$. From above we get the innovation and Kalman gain as:

$$e_{(i)} = z_{()i)} - \hat{\theta}_{(i)}^- \tag{26}$$

and

$$k_{(i)} = \frac{{r_{(i)}}^2}{{v_{(i)}}^2 + {r_{(i)}}^2} \tag{27}$$

A Gaussian postulate of the Kalman filter is imposed. This results in the fused scene coordinates of p_i with the least square error follow the posterior distribution as: $\theta_{(i)}^+ \sim \mathcal{N}(\hat{\theta}_{(i)}^+, {\sigma_{(i)}}^2\mathbb{I}_3)$ where $\hat{\theta}_{(i)}^+ = \hat{\theta}_{(i)}^- + k_{(i)}e_{(i)}$ and ${\sigma_{(i)}}^2 = {r_{(i)}}^2(1 - k_{(i)})$

Hence the Kalman filtering system is parameter free with the loss defined based on the posterior distribution:

$$\mathcal{L}_{posterior} = \sum_{i=1}^{N} \left(3log\sigma_{(i)} + \frac{\left\| \hat{\theta}_{(i)}^+ - y_{(i)} \right\|_2^2}{2\sigma_{(i)}^2} \right) \tag{28}$$

which is then added to the overall loss of the entire system.

$$\mathcal{L}_{posterior} = \tau_1\mathcal{L}_{likelihood} + \tau_2\mathcal{L}_{prior} + \tau_3\mathcal{L}_{posterior} \tag{29}$$

B. Consistency Examination

A statistical assessment tool called the Normalized Innovation Squared(NIS) is being added to filter any inconsistent predictions during inference. The inconsistencies occur due to the outlier estimations caused by the erratic scene coordinate regression or a failure of flow tracking. All of these issues can have an impact on the filtering.

We know that the innovation of Kalman gain follows a Gaussian distribution $\mathcal{N}(0, S_{(i)})$ where $S_{(i)} = (v^2_{(i)} + r^2_{(i)})\mathbb{I}_3$ The NIS test: $NIS = e^T_{(i)} S^{-1}_{(i)} e_{(i)}$ follows the Chi-squared distribution. In this test, it is determined if a pixel state lies outside or inside the $\chi^2(3)$ region. A critical value of 0.05 is used in the NIS test. It indicates that there is a requirement of 95% or more to set as a pixel state as negative. This also sets the uncertainty values of the failing pixels so large that it wont have any significant impact on the steps such as loss computation.

VII. Experiments and Results

A. Quantitative Results

1) Camera pose estimation: In this subsection, we will compare the translation and rotation errors by various visual odometry methods that predict the camera poses, involving Kalman Filters. Specifically, we compare PoseNet [1], LSTM-KF [6] and our KFNet methods. PoseNet is a fully supervised deep learning model that only relies on convolutional neural networks for regressing relative poses. Using PoseNet, temporal regularization using Kalman Filters were experimented in [6]. They use a standard Kalman filter and assumed constant velocity (PoseNet-KV) or constant acceleration (PoseNet-KA). Next, LSTM-KF [6] uses images to obtain image features, that are passed to the recurrent neural network stage involving LSTM (Long-Short Term Memory), which are capable of sequential modelling. These models are trained by direct supervision to regress relative poses. We compare these models to the ScoordNet and KFNet models, which are trained by structure based methods, with probabilistic loss functions. The scene coordinates regressed from ScoordNet is used to obtain the camera poses using PnP algorithm, instead of directly regressing relative camera poses. Table I compares these models with 4 sequences from the 7-scenes dataset. Due to the Bayesian formulation for the training and the structure based approach of estimating poses, we observe both ScoordNet and KFNet to perform much better than it's counter parts that directly regresses camera poses. Moreover, the improvement of KFNet from ScoordNet, which acts as a standalone measurement system, is attributed to the Kalman filtering process that takes place during test time.

2) Optical Flow: We also inferred the optical flow results of OFlowNet with respect to the groundtruth. We report the loss value, accuracy and the end point error (2-norm). These results are in table II

3) Scene Coordinate Regression: We inferred the process, measurement and filtered point cloud estimates from KFNet. These results are provided in table III. Next, We compared the results of KFNet with the state of the art and those results are shown in IV.

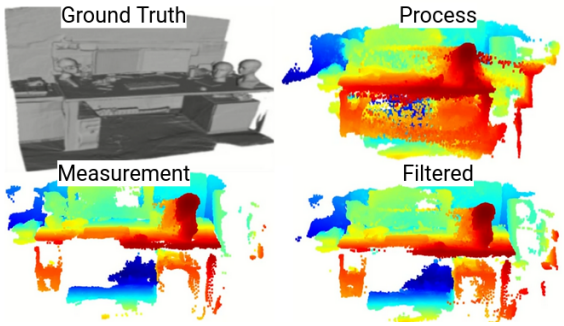

Fig. 3. Point cloud visualization of fire scene in 7 scenes dataset.

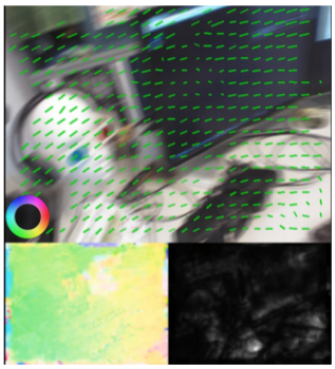

Fig. 4. Quiver plot, color wheel plot and the uncertainity map of optical flow from 7 Scenes-heads dataset.

B. Qualitative Results

1) Scene Coordinates: We performed 3D reconstruction of the scene using Open3D. We performed the 3D reconstruction using estimates from the process, measurement as well as the fitering system and those reconstruction results can be seen here. Figure 3 shows one such example reconstruction.

2) Optical Flow : We also visualized the Optical Flow and the uncertainity maps from the OFlowNet. Note that OFlowNet constitutes the process system and we qualitatively inferred it to be highly generalizable. Figure 4 shows the quiver plot, color wheel based visualization and the uncertainity map (in grayscale)

All the above process was performed in a Docker container. A tensorflow 1.10.1-gpu image was used from docker hub which supports CUDA & CUDNN. A python script for running KFNet has provided to verify the results obtained for the heads and the fire dataset. The python scripts can be found here. Results from KFNet have already been provided here. The above directory also consists of the docker images required to run the code. This docker image has to be converted into a container which then can be used to get the results from KFNet. Detailed ReadMe instructions to run the entire project can be found in the git repo.

TABLE I
MEDIAN TRANSLATION AND ROTATION ERRORS IN VARIOUS IMPLEMENTATIONS BASED ON KALMAN FILTERING

Datasets	PoseNet - KV		PoseNet - KA		LSTM-KF		ScoordNet		KFNet	
	trans m	rot $^\circ$	trans m	rot $^\circ$	trans m	rot $^\circ$	trans m	rot $^\circ$	trans m	rot $^\circ$
Fire	0.47	16.66	0.47	16.67	0.41	15.7	0.023	0.91	0.023	0.9
Heads	0.32	14.73	0.32	14.71	0.28	13.01	0.018	1.26	0.014	0.82
Office	0.48	8.64	0.48	8.62	0.43	7.65	0.026	0.73	0.025	0.69
Stairs	0.54	16.58	0.54	16.58	0.46	14.56	0.037	1.06	0.033	0.94
Average	0.45	14.15	0.45	14.14	0.395	12.73	0.026	0.99	0.095	0.83

TABLE II
LOSS, ACCURACY & END POINT ERROR (EPE) FOR OFLOWNET

scene	fire			heads			office			stairs		
	loss	acc	epe	loss	acc	epe	loss	acc	epe	loss	acc	epe
OFlowNet	-11.43	0.94	1.11	-14.34	0.97	0.67	-12.11	0.96	0.89	-10.53	0.92	1.31

TABLE III
DISTANCE ERRORS (IN CM) FOR THE SUBSYSTEMS IN KFNET INFERRED FROM 7 SCENES - HEADS AND FIRE DATA

	fire			heads		
Subsystems:	median	mean	std-dev	median	mean	std-dev
Measurement	1.91	7.72	10.09	2.70	9.72	13.09
Process	1.92	6.93	8.75	2.66	7.33	10.25
Filtered	1.88	6.81	8.69	2.16	7.21	9.79

TABLE IV
DISTANCE ERRORS (IN CM) COMPARED WITH STATE OF THE ART - 7 SCENES DATASET

Models:	mean	stddev
DSAC++	28.8	33.1
SCoordNet	16.8	23.3
KFNet	15.3	21.7

VIII. CONCLUSION

This work performed by the authors of the KFNet paper provide a neural network architecture named KFNet which addresses the camera localization problem in the context of Bayesian learning by embedding a structure based camera estimation technique into the Kalman filter approach. The results evidently show a significant improvement in camera position and orientation estimation as compared to other networks. The network also obtained a better accuracy metric among the state-of-the-art localization methods. Future work involves utilizing the same Kalman filter approach but for an array of other computer vision techniques such as image segmentation, object tracking & more.

IX. APPENDIX

A. Preliminary Probability Formulae

In this subsection we will define some important formulae for deriving the posterior probability/belief in section II. This was learned from [7]

- Total Probability Theorem:

$$P(A \mid B) = \frac{P(A,B)}{P(B)} \tag{30}$$

- Bayes Rule: Conditional probability of A, given B and C is,

$$P(A \mid B, C) = \frac{P(B \mid A, C)P(A \mid C)}{P(B \mid C)} \tag{31}$$

here $\eta = P(B \mid C)$, a fixed normalization term.

- Probability of event A and B, when involved with finite number of events C_i, can be given as

$$\begin{aligned} P(A,B) &= \sum_i P(A,B,C_i) \\ &= \sum_i P(A \mid B, C_i)P(C_i \mid B)P(B) \end{aligned} \tag{32}$$

- Using equation 32, we can define the probability of an event A, given B as

$$\begin{aligned} P(A \mid B) &= \frac{P(A,B)}{P(B)} \\ &= \frac{\sum_i P(A \mid B, C_i)P(C_i \mid B)P(B)}{P(B)} \\ &= \sum_i P(A \mid B, C_i)P(C_i \mid B) \end{aligned} \tag{33}$$

B. Markov Assumption

In markov's model, we make the following two assumptions.

- The measurements are only dependant on the current state.

$$P(z_t \mid x_{0:t}, z_{0:t-1}, u_{1:t}) = P(Z_t \mid x_t) \tag{34}$$

- The current state is dependant only on previous state and current input.

$$P(x_t \mid x_{1:t}, z_{1:t-1}, u_{1:t}) = P(x_t \mid x_{t-1}, u_t) \tag{35}$$

C. Derivation in Bayesian Formulation

The distribution with θ_t and e_t, given the image sequence $\mathcal{I}_{t-1}$ was denoted in equation 9 as

$$\left[\begin{pmatrix}\theta_t \\ e_t\end{pmatrix} \mid \mathcal{I}_{t-1}\right] = \mathcal{N}\left[\begin{pmatrix}\mu_1 \\ \mu_2\end{pmatrix}, \begin{pmatrix}\Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22}\end{pmatrix}\right] \tag{36}$$

This is a bivariate distribution of form, $X \sim \mathcal{N}(\mu, \Sigma)$ where $X = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$, $\mu = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}$, $\Sigma = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}$.

To find the expectation and covariances, we need to find the distribution of one variable (say x_1) conditioned on the other (x_2) and vice versa. Using the theorems of multivariate statistics, We can obtain the conditional distribution $(x1 \mid x2) \sim \mathcal{N}(\bar{\mu}, \bar{\Sigma})$, with the expectation

$$\bar{\mu} = \mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(x_2 - \mu_2) \tag{37}$$

and covariance

$$\bar{\Sigma} = \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21} \tag{38}$$

The proof for arriving at equations 37 and 38 is as follows:

We will consider a third variable z, that is a linear combination of the two variables x_1 and x_2 given as,

$$z = C_1 x_1 + C_2 x_2 \tag{39}$$

We also want z to be independant of x_2, inorder to utilise the property that, $P(z \mid x_2) = P(z)$, so that, the expectation $E(z \mid x_2) = E(z)$ and the variance $var(z \mid x_2) = var(z)$.

Using this above property, we can eventually derive $var(C_1 x_1 \mid x_2)$ and $E(C_1 x_1 \mid x_2)$. If we consider $C_1 = I$, we will derive $\bar{\mu} = var(x_1 \mid x_2)$ and $\bar{\Sigma} = E(x_1 \mid x_2)$.

Let the third variable $z = x_1 + C_2 x_2$, be independent of x_2. It's covariance is given as

$$\begin{aligned} cov(z, x_2) &= cov(x_1, x_2) + cov(C_2 x_2, x_2) \\ 0 &= cov(x_1, x_2) + C_2 var(x_2, x_2) \end{aligned} \tag{40}$$

Since $cov(x_1, x_2) = \Sigma_{12}$ and $var(x_2, x_2) = \Sigma_{22}$, we derive,

$$\begin{aligned} \Sigma_{12} + C_2\Sigma_{22} &= 0 \\ C_2 &= -\Sigma_{12}\Sigma_{22}^{-1} \end{aligned} \tag{41}$$

and hence we can derive that fully defines intermediate variable z. Using z, we will now find $\bar{\mu}$ and $\bar{\Sigma}$.

Computing $\bar{\mu}$*:* Having z be a linear combination of the two gaussians, the expectation of z is also a linear combination, $E(z) = \mu_1 + C_2\mu_2$. The expectation $E(x_1 \mid x_2)$ is expanded as,

$$\begin{aligned} E(x_1 \mid x_2) &= E(z - C_2 x_2 \mid x_2) \\ &= E(z \mid x_2) - E(C_2 x_2 \mid x_2) \\ &= E(z) - C_2 x_2 \\ &= \mu_1 + C_2(\mu_2 - x_2) \end{aligned} \tag{42}$$

Using equation 41, we obtain.

$$\bar{\mu} = E(x_1 \mid x_2) = \mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(x_2 - \mu_2) \tag{43}$$

Computing $\bar{\Sigma}$: The variance $var(x_1 \mid x_2)$ is given as,

$$var(x_1 \mid x_2) = var(z - C_2 x_2 \mid x_2) \tag{44}$$

The variance of sum of two distributions (say a and b) is equal to the sum of their variances and covariances, given as $var(a+b) = var(a) + var(b) + cov(a, b) + cov(b, a)$. By this property,

$$\begin{aligned} var(x_1 \mid x_2) = var(z \mid x_2) &+ var(C_2 x_2 \mid x_2) \\ &- C_2 cov(z, -x_2) - cov(z, -x_2)C_2' \end{aligned} \tag{45}$$

$$\begin{aligned} &= var(z \mid x_2) \\ &= var(z) \end{aligned} \tag{46}$$

Having known this relationship, deriving var(z) is same as deriving $\bar{\Sigma}$.

$$var(z) = var(x_1 + C_2 x_2) \tag{47}$$

$$\begin{aligned} var(z) = var(x_1) &+ C_2 var(x_2) C_2' \\ &+ C_2 cov(x_1, x_2) + cov(x_2, x_1) C_2' \end{aligned} \tag{48}$$

$$\begin{aligned} var(z) = \Sigma_{11} &+ \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{22}\Sigma_{22}^{-1}\Sigma_{21} \\ &- \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{12} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21} \end{aligned} \tag{49}$$

We know, $\Sigma_{12} = \Sigma_{21}, C_2 = -\Sigma_{12}\Sigma_{22}^{-1}$,and $C_2' = \Sigma_{22}^{-1}\Sigma_{21}$

$$var(z) = \Sigma_{11} + \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21} - 2\Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21} \tag{50}$$

$$\bar{\Sigma} = var(z) = \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{22} \tag{51}$$

Equation 43 is the expectation and equation 51 is the covariance matrix of the bivariate distribution. Thus, we can arrive at the conditional distribution,

$$\begin{aligned} &(x_1 \mid x_2) \sim \\ &\quad \mathcal{N}(\mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(x_2 - \mu_2), \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}) \end{aligned} \tag{52}$$

$$\begin{aligned} &(x_2 \mid x_1) \sim \\ &\quad \mathcal{N}(\mu_2 + \Sigma_{21}\Sigma_{22}^{-1}(x_1 - \mu_1), \Sigma_{22} - \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12}) \end{aligned} \tag{53}$$

Having derived the expectation and covariance for the conditional distribution of the variable, we proceed to derive the expectation and covariance of the bivariate distribution.

First, consider the univariate distribution $(\theta_t \mid \mathcal{I}_{t-1}) \sim \mathcal{N}(\mu_1, \Sigma_{11})$. Comparing this to Equation 13, we see that $\mathcal{N}(\mu_1, \Sigma_{11}) = \mathcal{N}(\hat{\theta}_t^-, R_t)$ and hence we can derive

$$\mu_1 = \hat{\theta}_t^-; \Sigma_{11} = R_t \tag{54}$$

Next, Using this relationship in equation 53, we can write the error distribution conditioned on the state as,

$$\begin{aligned} &(e_t \mid \theta_t, I_{t-1}) \sim \\ &\quad \mathcal{N}(\mu_2 + \Sigma_{21}\Sigma_{11}^{-1}(\theta_t - \mu_1), \Sigma_{22} - \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12}) \end{aligned} \tag{55}$$

Comparing equation 55 with 17, we can derive,

$$\begin{aligned} \mu_2 &= 0; \\ \Sigma_{12} = \Sigma_{21} &= R_t; \\ \Sigma_{22} &= V_t + R_t \end{aligned} \tag{56}$$

Thus the equations 54 and 56 define all the parameters to explain the bivariate distribution in 19.

Finally, we can derive $(\theta_t \mid e_t, \mathcal{I}_{t-1})$ using the equation 52 and substituting the variables in equations 54 and 56 to obtain,

$$(\theta_t \mid e_t, \mathcal{I}_{t-1}) \sim \mathcal{N}(\hat{\theta}_t^- + K_t e_t, R_t(I - K_t) \tag{57}$$

where $K_t = \frac{R_t}{V_t + R_t}$, the kalman gain. Equation 57 is the posterior distribution θ_t^+ that needs to be predicted.

REFERENCES

[1] A. Kendall, M. Grimes and R. Cipolla, "PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization," 2015 IEEE International Conference on Computer Vision (ICCV), 2015, pp. 2938-2946, doi: 10.1109/ICCV.2015.336.

[2] A. Kendall and R. Cipolla, "Geometric Loss Functions for Camera Pose Regression with Deep Learning," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 6555-6564, doi: 10.1109/CVPR.2017.694.

[3] Martin A. Fischler and Robert C. Bolles. 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM 24, 6 (June 1981), 381–395. DOI:https://doi.org/10.1145/358669.358692

[4] Zhou, K., Liu, Z., Qiao, Y., Xiang, T., & Loy, C. C. (2021). Domain generalization in vision: A survey. arXiv preprint arXiv:2103.02503.

[5] Zhan, H., Weerasekera, C. S., Bian, J. W., Garg, R., & Reid, I. (2021). DF-VO: What Should Be Learnt for Visual Odometry?. arXiv preprint arXiv:2103.00933.

[6] Coskun, Huseyin & Achilles, Felix & DiPietro, Robert & Navab, Nassir & Tombari, Federico. (2017). Long Short-Term Memory Kalman Filters:Recurrent Neural Estimators for Pose Regularization.

[7] https://www.youtube.com/watch?v=6uEgLv1Mr2s