Universität Hamburg

MIN-Fakultät
Department Informatik

End-to-end visual odometry through deep neural networks

Lyu Jianzhi

(Oberseminar TAMS, 15.05.2018)

Universität Hamburg
Fakultät für Mathematik, Informatik und Naturwissenschaften
Department Informatik

Technische Aspekte Multimodaler System

About thiswork

End-to-end, sequence-to-sequence probabilistic visual odometry

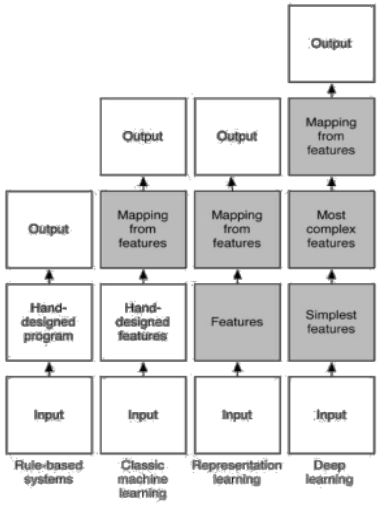
through deep neural networks

Sen Wang1,2, Ronald Clark2, Hongkai Wen2 and Niki Trigoni2
1.Edinburgh Centre for Robotics, Heriot-Watt University, UK
2.University of Oxford, UK

Download this paper: http://journals.sagepub.com/doi/pdf/10.1177/0278364917734298
Watch video: http://senwang.gitlab.io/DeepVO/#video

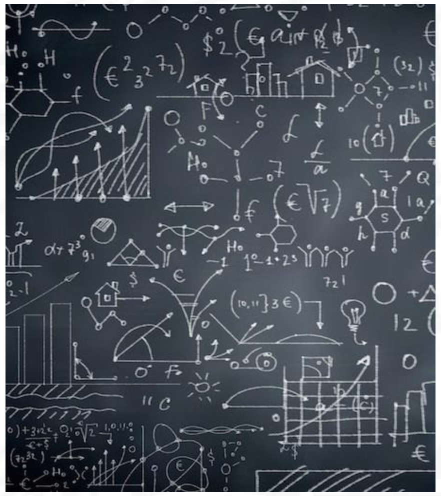
Contributions

1. **Proving that Monocular VO could be build by End-to- End training.**

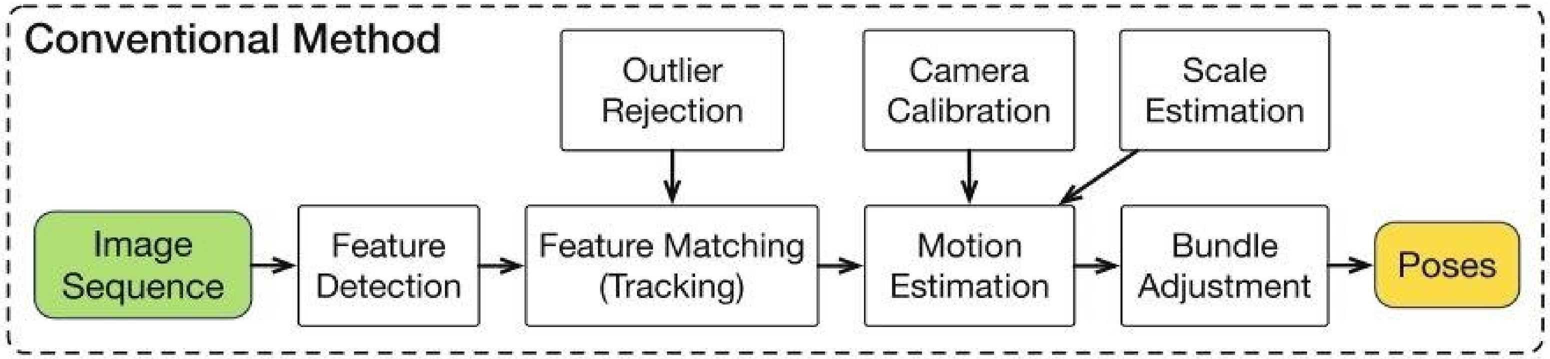


2. **RCNN architecture could generalized to unseen environment.**

3. **Complex movement could be modeled by RCNN.**



Conventional method

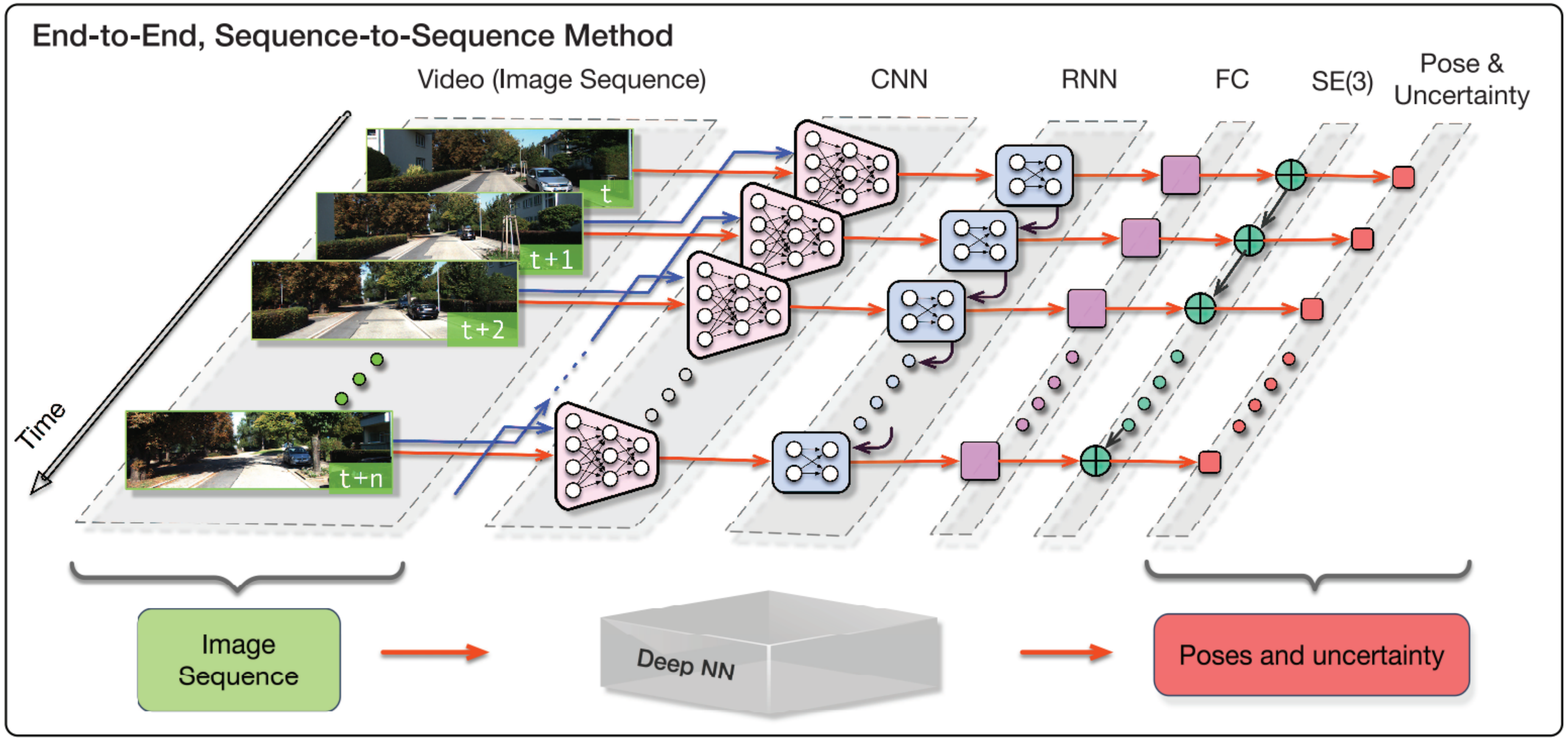


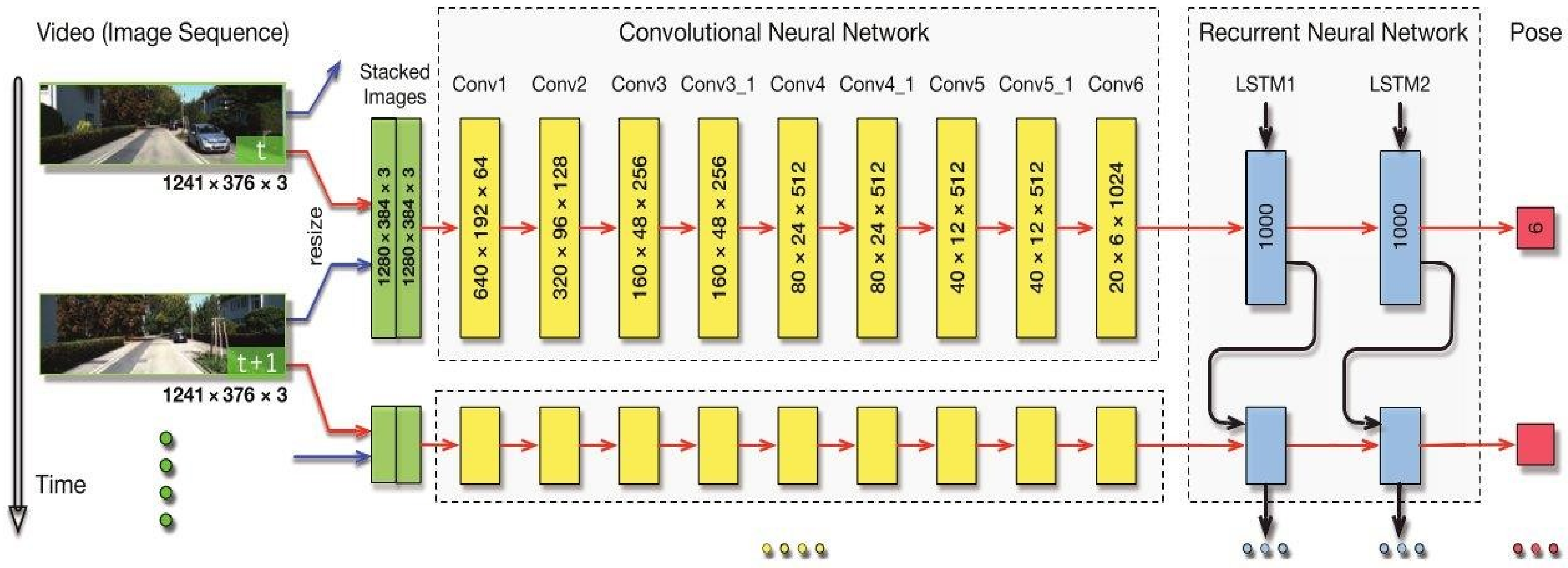
Network design

1. Traditional computer vision learn knowledge from appearance and image context
2. Visual odometry should learn from geometry.

This is what RCNN tried to address

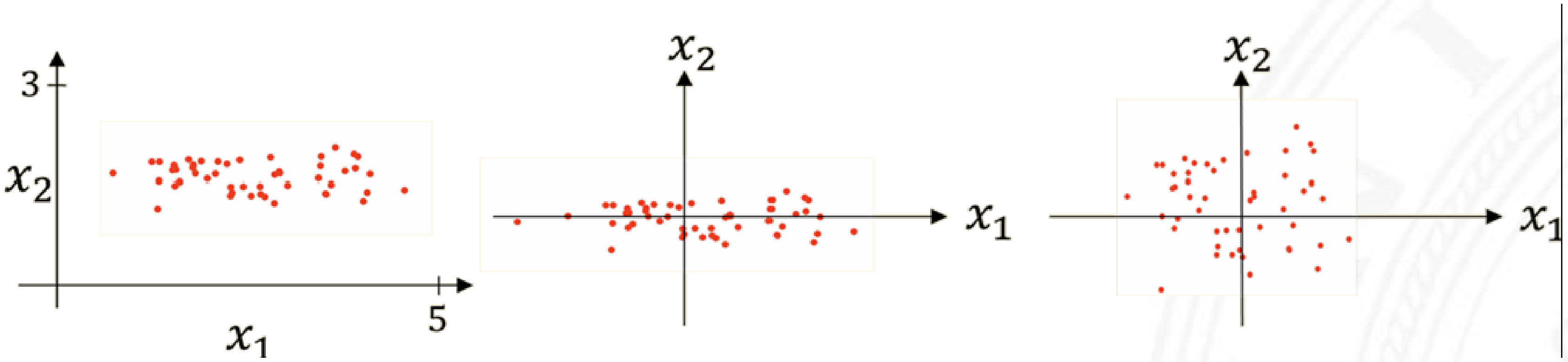
Network design




Video (Image Sequence)
Convolutional Neural Network
Recurrent Neural Network
Pose
Stacked Images
Conv1
Conv2
Conv3
Conv3_1
Conv4
Conv4_1
Conv5
Conv5_1
Conv6
LSTM1
LSTM2
t
1241 x 376 x 3
t+1
1241 x 376 x 3
resize
1280 x 384 x 3
1280 x 384 x 3
640 x 192 x 64
320 x 96 x 128
160 x 48 x 256
160 x 48 x 256
80 x 24 x 512
80 x 24 x 512
40 x 12 x 512
40 x 12 x 512
20 x 6 x 1024
1000
1000
6
Time

Preprocessing

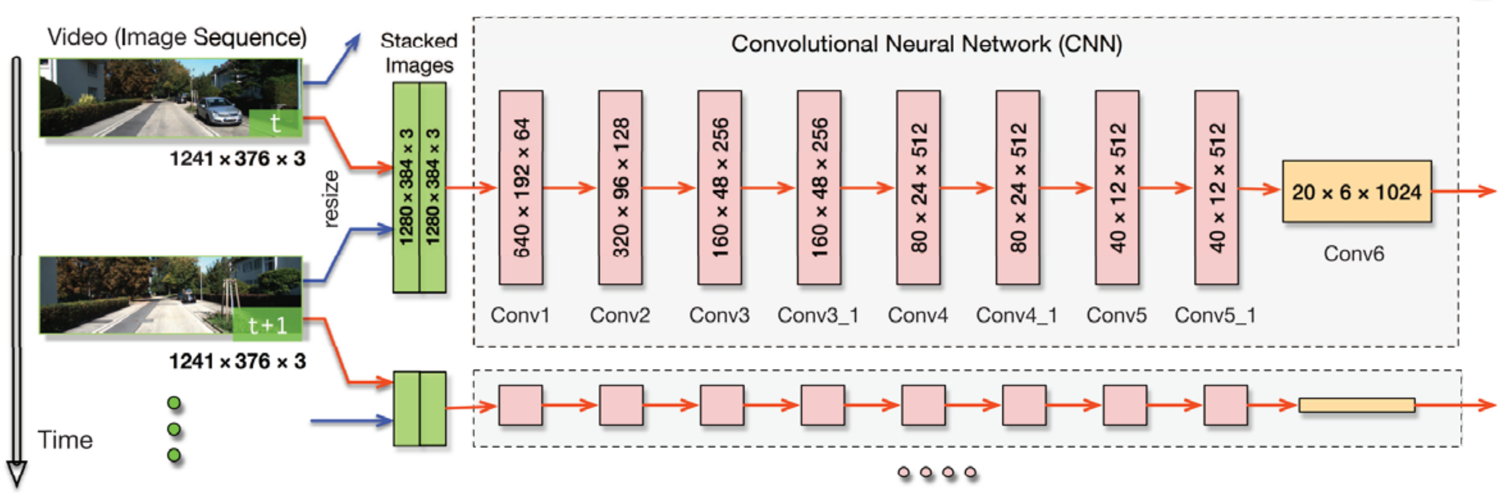
- Normalizing inputs (speed up training)
  => subtracting the mean RGB values of the training set



- Resize image to 64x
- Stack two images to form a tensor

CNN

- **What this research mean by learning "geometric" feature?**
  => They stacking two RGB images and feed it into CNN. Expecting the network to perform feature extraction on the concatenation of two consecutive monocular RGB images.

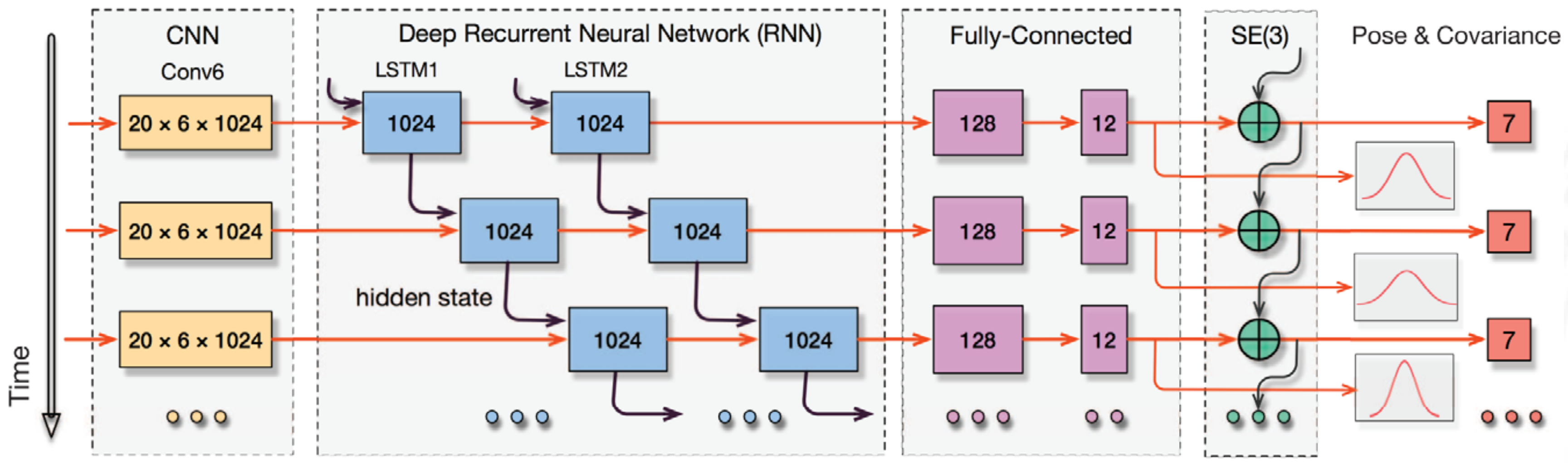


CONFIGURATION OF THE CNN

Layer	Receptive Field Size	Padding	Stride	Number of Channels
Conv1	7×7	3	2	64
Conv2	5×5	2	2	128
Conv3	5×5	2	2	256
Conv3_1	3×3	1	1	256
Conv4	3×3	1	2	512
Conv4_1	3×3	1	1	512
Conv5	3×3	1	2	512
Conv5_1	3×3	1	1	512
Conv6	3×3	1	2	1024

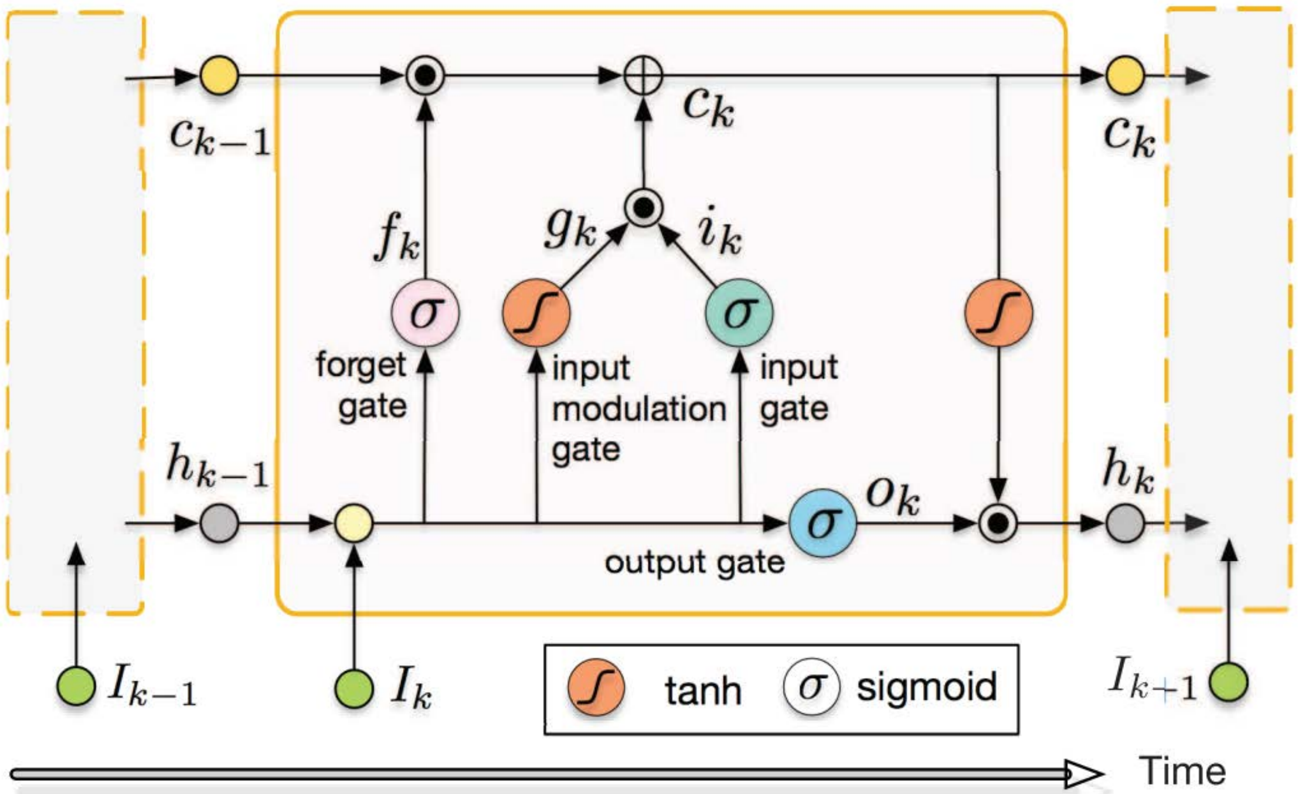
RNN

- RNN is not suitable to directly learn sequential representation from high-dimensional raw data, such as images.

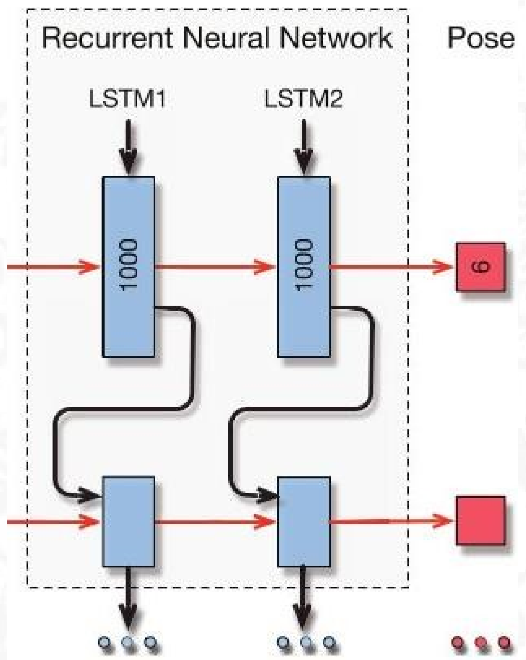


Vanishing gradient problem

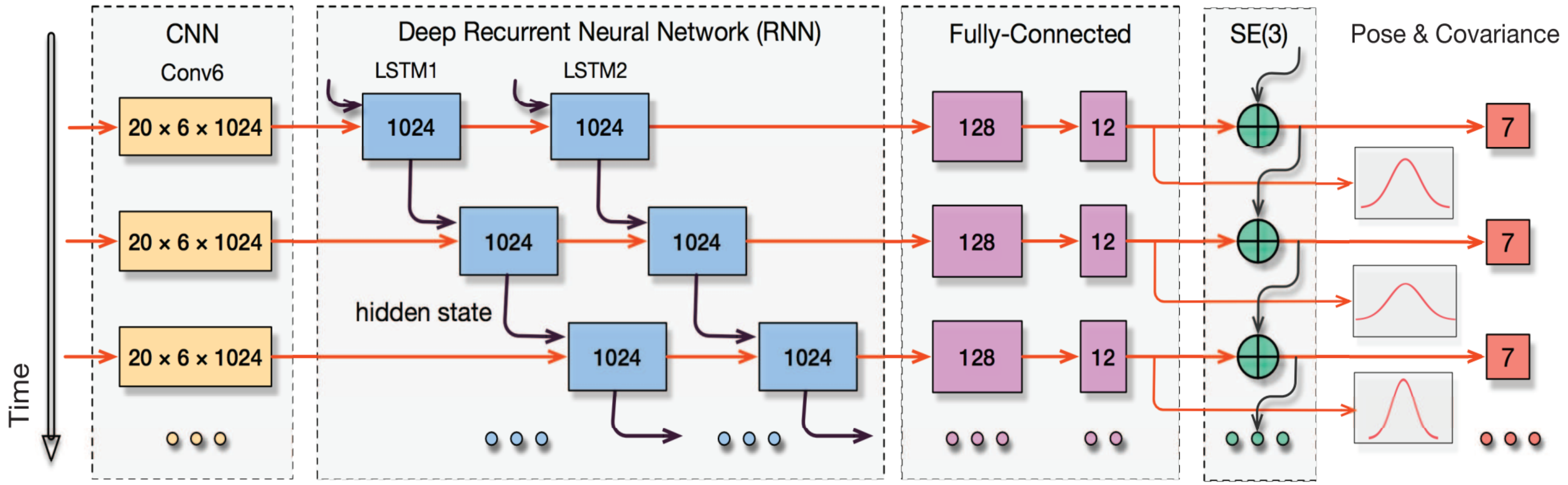
LSTM (Long short-term memory)



To get high level presentation

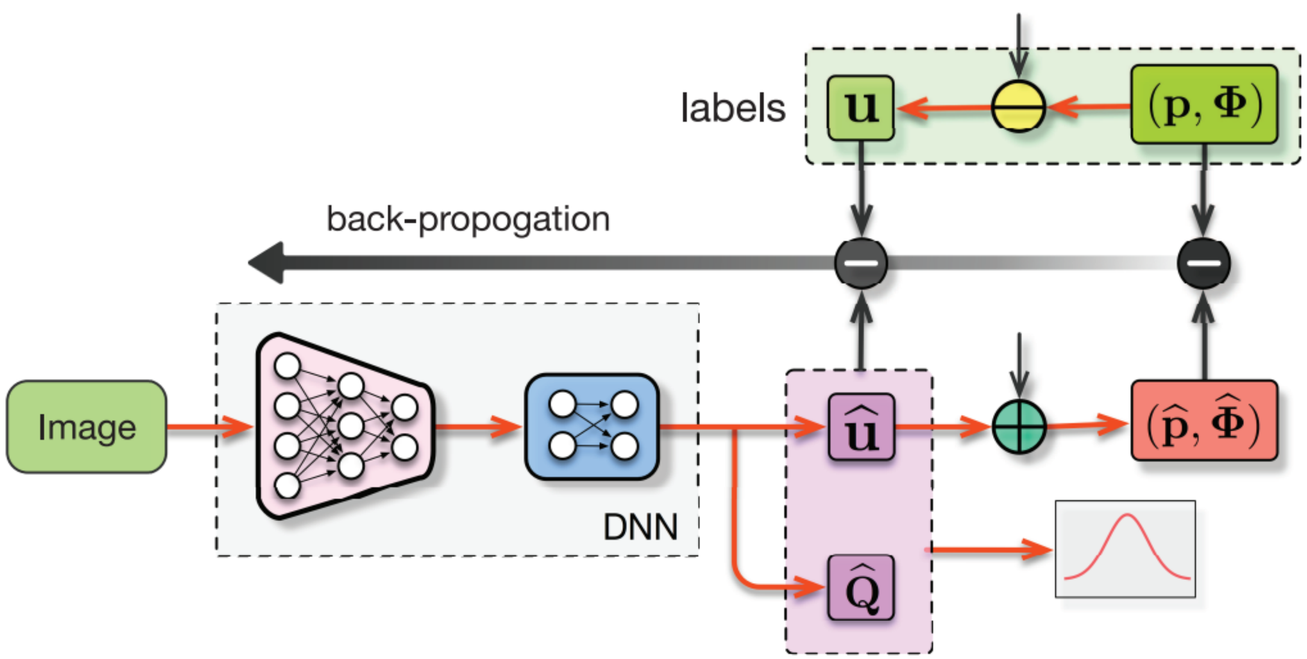


RCNN


CNN
Conv6
20 × 6 × 1024
20 × 6 × 1024
20 × 6 × 1024
Deep Recurrent Neural Network (RNN)
LSTM1
LSTM2
1024
hidden state
Fully-Connected
128
12
SE(3)
Pose & Covariance
7
Time

Cost function

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \frac{1}{t} \sum_{k=1}^{t} \| \widehat{\mathbf{p}}_k - \mathbf{p}_k \|_2^2 + \kappa \| \widehat{\mathbf{\Phi}}_k - \mathbf{\Phi}_k \|_2^2$$



$$\theta^* = \underset{\theta}{\operatorname{argmin}} \frac{1}{t} \sum_{k=1}^{t} \log | \widehat{\mathbf{Q}}_k | + (\widehat{\mathbf{u}}_k - \mathbf{u}_k)^T \widehat{\mathbf{Q}}_k^{-1} (\widehat{\mathbf{u}}_k - \mathbf{u}_k)$$

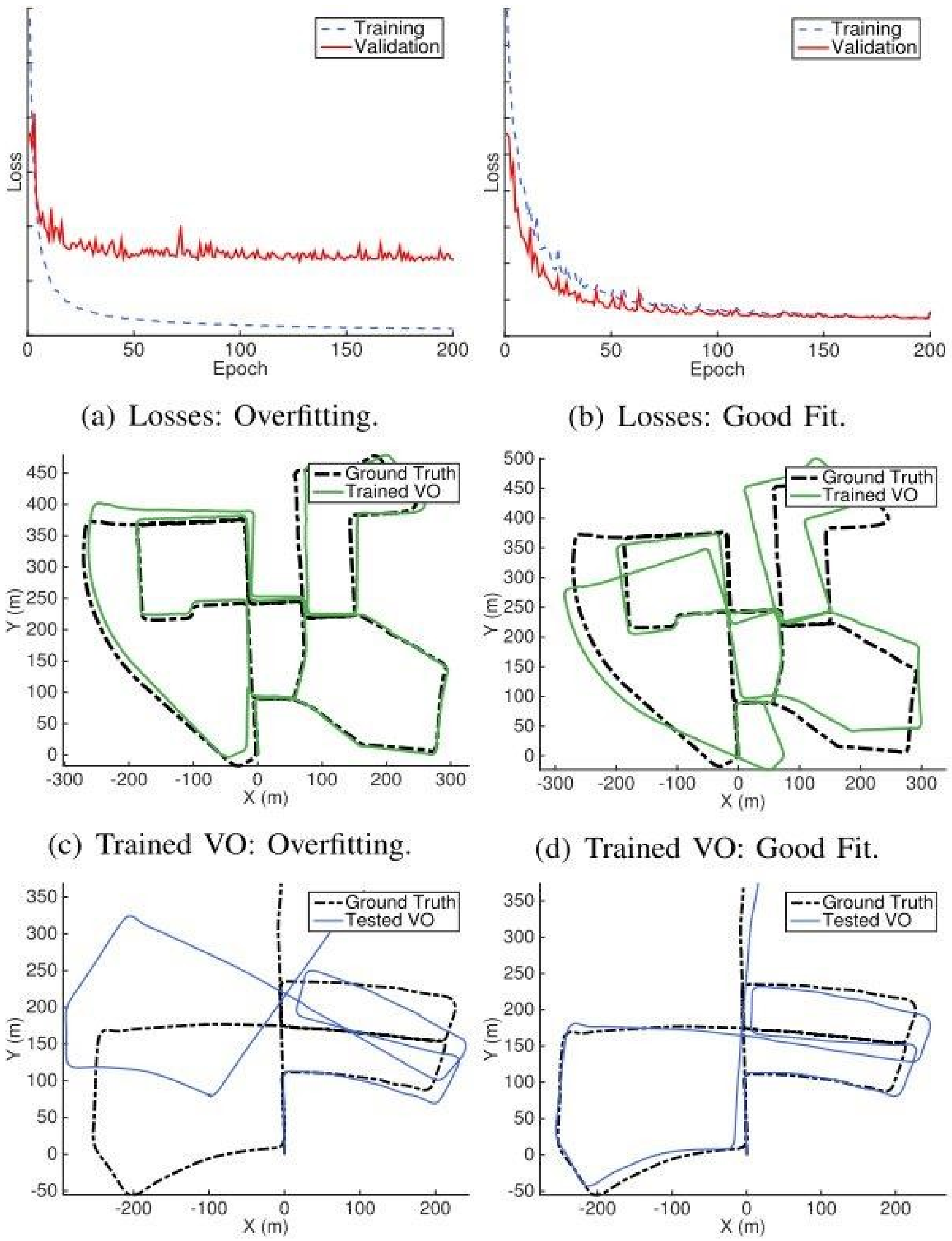
Experimental results

Training &testing

1. Dataset: KITTI VO / SLAM benchmark
   (22 sequences of images / 10fps / dynamic object)
2. 7410 training samples (image and trajectory pair)
3. Implemented based on Theano
4. Hardware: Nvidia Tesla K40 GPU
5. 200 epochs
6. Learning rate 0.001
7. Regularization: dropout / early stopping
8. CNN: transfer learning from FlowNet

Overfitting

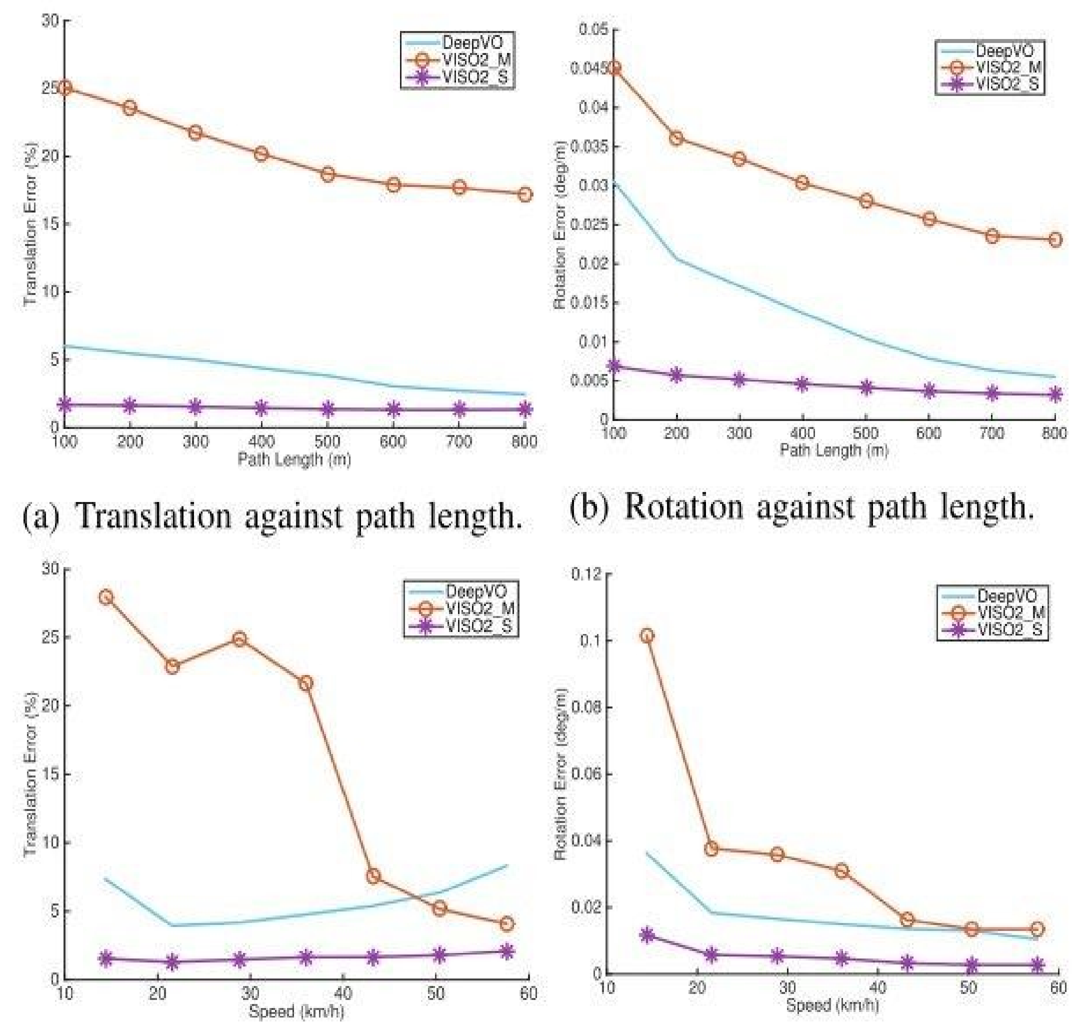
- **Orientation is more prone to overfitting**



(a) Losses: Overfitting.
(b) Losses: Good Fit.
(c) Trained VO: Overfitting.
(d) Trained VO: Good Fit.
(e) Tested VO: Overfitting.
(f) Tested VO: Good Fit.

Compare with traditional VO

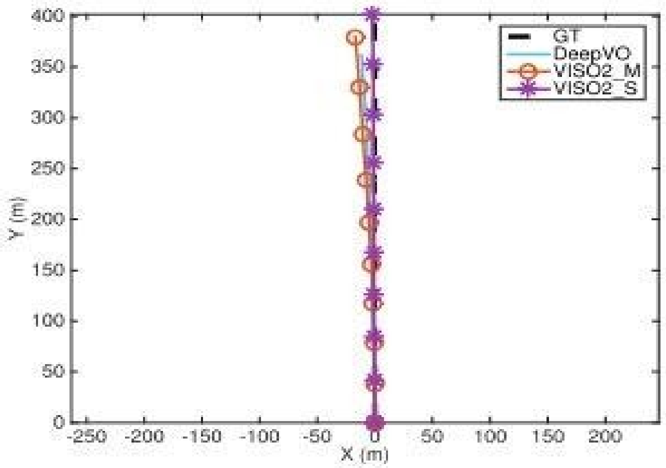
- Open-source VO library LIBVISO2
- Monocular / Stereo



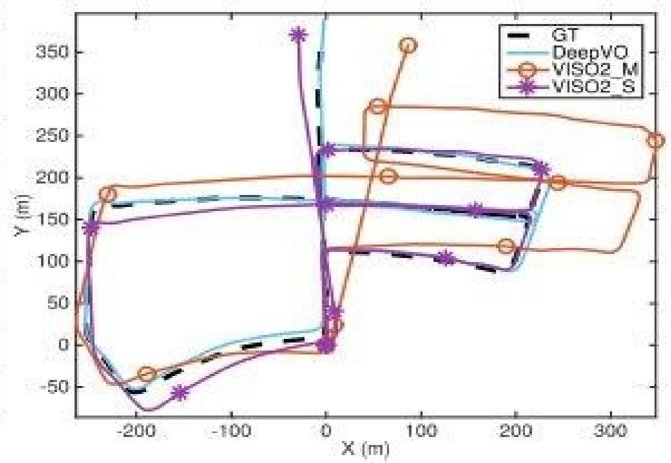
(a) Translation against path length. (b) Rotation against path length.
(c) Translation against speed. (d) Rotation against speed.

Fig. 5. Average errors on translation and rotation against different path lengths and speeds. The DeepVO model used is trained on Sequence 00, 02, 08 and 09.

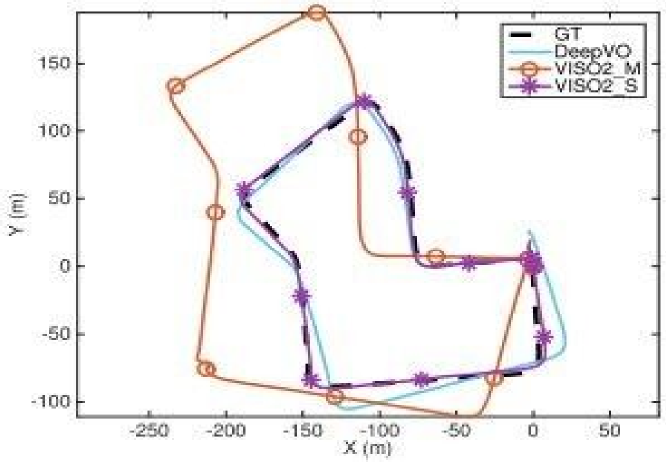
Trajectory (1/2)



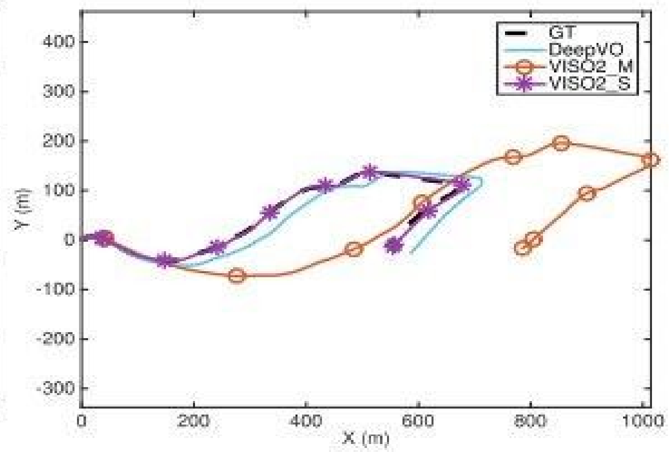
(a) Sequence 04.



(b) Sequence 05.



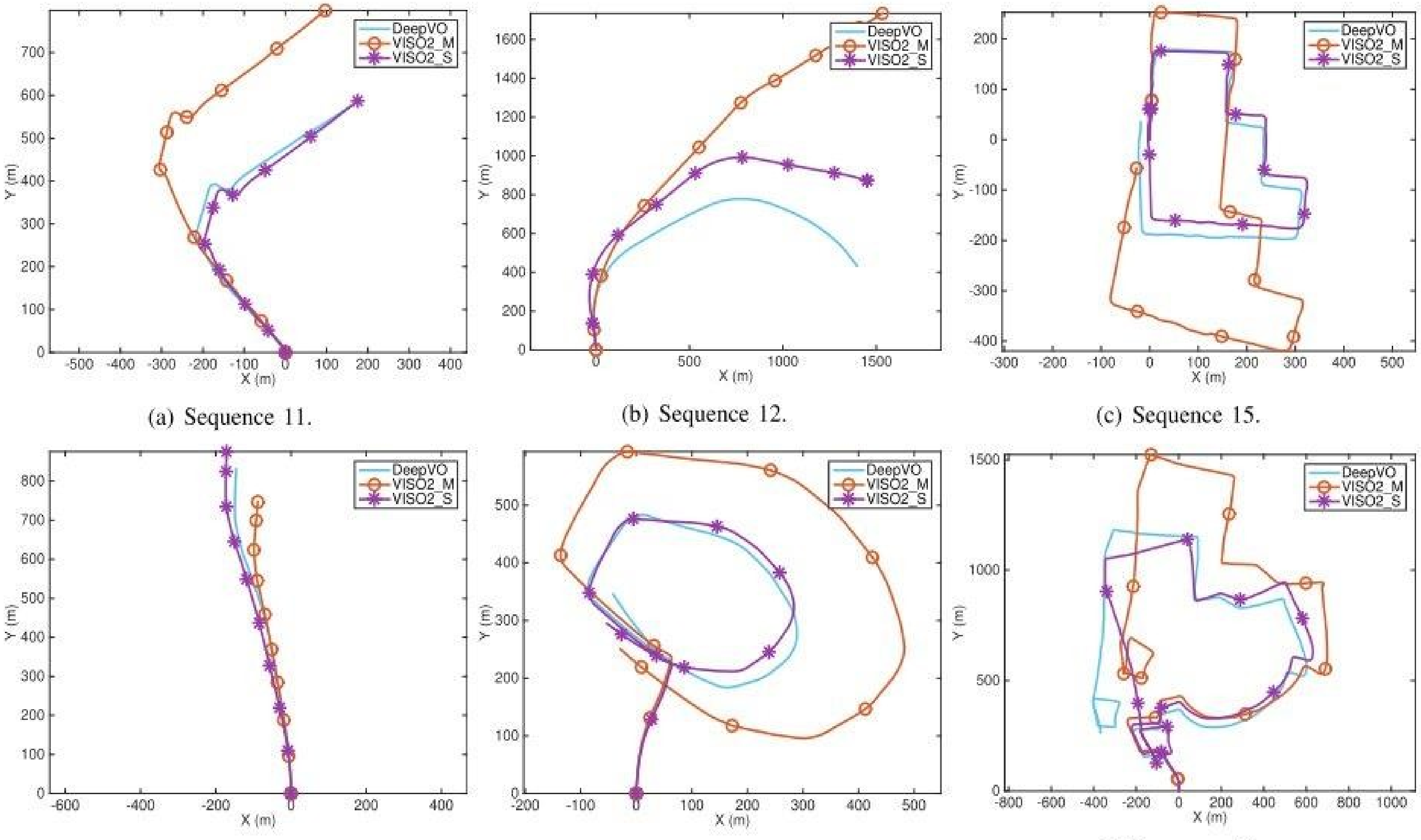
(c) Sequence 07.



(d) Sequence 10.

Trajectory (2/2)

- **No ground truth: Seq11~19**



(a) Sequence 11. (b) Sequence 12. (c) Sequence 15.
(d) Sequence 17. (e) Sequence 18. (f) Sequence 19.

TABLE II

RESULTS ON TESTING SEQUENCES.

Seq.	DeepVO		VISO2_M		VISO2_S	
	$t_{\text{rel}}(\%)$	$r_{\text{rel}}(^\circ)$	$t_{\text{rel}}(\%)$	$r_{\text{rel}}(^\circ)$	$t_{\text{rel}}(\%)$	$r_{\text{rel}}(^\circ)$
03	8.49	6.89	8.47	8.82	3.21	3.25
04	7.19	6.97	4.69	4.49	2.12	2.12
05	2.62	3.61	19.22	17.58	1.53	1.60
06	5.42	5.82	7.30	6.14	1.48	1.58
07	3.91	4.60	23.61	29.11	1.85	1.91
10	8.11	8.83	41.56	32.99	1.17	1.30
mean	5.96	6.12	17.48	16.52	1.89	1.96

- t_{rel}: average translational RMSE drift $(\%)$ on length of 100m-800m.
- r_{rel}: average rotational RMSE drift $(^\circ/100m)$ on length of 100m-800m.
- The DeepVO model used is trained on Sequence 00, 02, 08 and 09. Its performance is expected to improve when it is trained on more data.

Conclusion

- End-to-end monocular VO based on Deep learning
- Deep RCNN
- No need to carefully tune the parameters of the VO system
- It is not expected as a replacement to the classic geometry based approach