HYBRID DEEP LEARNING APPROACH FOR VEHICLE'S RELATIVE ATTITUDE ESTIMATION USING MONOCULAR CAMERA

Mehad Haggag[1,2*], Adel Moussa [1,3,] Naser El-Sheimy[1]

[1] Dept. of Geomatics Engineering, University of Calgary, Calgary, AB, Canada – (mehad.haggag, amelsaye, elsheimy)@ucalgary.ca
[2] Dept. of Geomatics Engineering, Benha University, Benha, Egypt – mohad.mohamed@feng.bu.edu.eg
[3] Department of Electrical and Computer Engineering, Port-Said University, Port-Said, Egypt.

KEY WORDS: Relative pose estimation, Monocular camera, Ego-motion, Optical flow map, KLT tracker, Deep Learning.

ABSTRACT:

Relative pose estimation using a monocular camera is one of the most common approaches for aiding vehicle's navigation. It involves determining the position and orientation of a vehicle relative to its surroundings using only a single camera. This can be achieved through four main steps: feature detection and matching, motion estimation, filtering and optimization, and scale estimation. Feature tracking involves detecting and tracking distinctive visual features in the environment, such as corners or edges, and using their relative motion to estimate the camera's movement. This approach can be prone to errors due to feature detection and tracking difficulties, as well as issues with moving objects, occlusions, and changes in lighting conditions. These typical computer vision approaches are computationally intensive and may require significant processing power as well, which limits their real time application. This paper proposes a hybrid deep neural network approach for estimating the relative attitude of a vehicle using a monocular camera to aid in vehicle navigation. The proposed neural network adopts a relatively shallow architecture to minimize the computational cost and to meet the real-time requirements of low-cost processing systems. The network is trained using the KITTI dataset and can estimate the relative attitude of the vehicle with a RMSE of relative orientation of 0.017 degrees per frame. The processing time of the proposed approach is around 28 ms per frame including both the tracking and network prediction steps, which is significantly faster than the typical estimation pipelines. The results show that the proposed approach is a viable alternative to conventional computer vision methods and can significantly reduce computational costs, deal with the confusing scenarios of the moving objects while maintaining a good accuracy in estimating ego-motion.

1. INTRODUCTION

In recent years, vision-aided techniques for autonomous vehicle navigation have emerged as a promising alternative for pose estimation. Compared to other navigation systems, vision-aided navigation systems can be relatively cost-effective, as they can use off-the-shelf hardware and software components.

Vision-based techniques are not affected by wheel slip in uneven terrain, making them more robust in challenging environments. Moreover, localization with imagery can be compatible with other computer vision tasks, such as classification and object detection, enabling the development of more complex systems for better situational awareness and informed decision making.

The typical pipeline of computer vision approaches for pose estimation involves four main steps: feature detection and matching, motion estimation, filtering and optimization, and scale estimation. Feature detection and matching involves identifying and tracking distinctive features in the images. Motion estimation involves computing the camera's motion by analysing the changes in the positions of the tracked features between frames. Filtering and optimization involve smoothing the estimated motion to reduce errors and improve accuracy. Finally, scale estimation involves determining the scale of the estimated motion using external sources of information, such as GNSS, IMU sensors or odometer.

Recently, deep neural networks (DNNs) have shown promising results in various computer vision and image processing applications, outperforming traditional techniques. Also, the deep learning neural networks have been recently introduced in the field of localization and pose estimation. This research paper proposes a hybrid approach combining deep learning and computer vision algorithms for reduced computational cost, and enhanced adaptability to various environmental conditions.

By using a monocular camera and a regression-based deep neural network, we aim to approximate the changes in consecutive frames and the relative vehicle pose. Additionally, an analysis of processing times was conducted for both the hybrid and traditional computer vision techniques to compare the efficiency of each approach. The proposed method demonstrates promising results and holds potential for further research and application in real-world scenarios.

2. RELATED WORK

Relative pose estimation techniques can be divided into two main categories: computer vision-based algorithms and more recently, deep learning-based algorithms. Computer vision algorithms have a superior performance in terms of accuracy and robustness. The typical workflow of computer vision algorithms is extracting and matching a set of image features, constructing multiview geometry then attempting to estimate motion from a sequence of images.

These methods typically apply feature detection methods like FAST (Rosten et al. 2006), SURF (Bay et al. 2008), ORB (Rublee et al. 2011), SIFT (Lowe, D. G. ,2004) and BRIEF (Calonder et al. 2010) to extract points of interest. The Kanade– Lucas–Tomasi (KLT) (Tomasi et al. 1991) feature tracker is one of the most common feature point tracking methods to track points in the sequential frames.

As the matching algorithms essentially rely on the image intensity values, some points could share the same appearance however they do not belong to the same object point. That is why it is

https://doi.org/10.5194/isprs-annals-X-1-W1-2023-649-2023 | © Author(s) 2023. CC BY 4.0 License.

essential to remove outliers from matched points after matching algorithm. The most popular method for outlier removal is RANSAC (Fischler et al. 1981). Some other robust estimation methods are M-estimation (Torr et al. 1997), case deletion, and explicitly fitting and removing outliers (Sim et al.2006).
However, even though the computer vision methods have a robust accuracy, but they are computationally expensive and still challenging in real-time applications. To overcome the large computational requirements of these methods, some approaches (Scaramuzza et al.2009), (Engel et al.2013) and (Forster et al.2014) were proposed to try to simplify the computations and enhance the performance of monocular relative pose estimation.
Besides, the computer vision algorithms make use of geometry reconstruction from matched image points to estimate the camera pose state; that is why, the geometry-based methods cannot work well on some conditions like: i) image frame without obvious texture information or strong interest points, (ii) environment with light intensity changing abruptly, or (iii) when the vehicle is moving at a high speed as the consecutive frames should have sufficient overlap for matching process. Moreover, the real scale estimation and camera calibration parameters are very essential parts for pose estimation process.

Recently, some deep learning-based VO methods have been developed without explicitly using any geometric reconstruction. The deep learning methods have achieved promising results in estimating optical flow or estimating 6 DoF poses. The network called PoseNet (Kendall et al. 2015) uses CNNs to learn mapping from images to estimate absolute six-DoF poses. FlowNet (Dosovitskiy et al. 2015) makes use of optical flow between images. GeoNet, (Yin et al. 2018) is an unsupervised learning framework for optical flow, monocular depth, and ego-motion estimation from videos. VLocNet, (Valada et al. 2018) is another CNN architecture for six-DoF pose regression and odometry estimation from consecutive monocular images.

3. METHODOLOGY

The conventional computer vision approaches for pose estimation begin with feature extraction such as corners or edges from images. This may be simple corner detector as Harris, Förstner, and Moravec or more sophisticated feature extraction algorithms were tested SIFT, FAST, SURF, ORB, and BRISK.

SIFT is designed to identify and describe local features in an image that are invariant to scale, rotation, and translation. While, FAST is a fast and efficient algorithm that operates by analysing a small set of contiguous pixels in a circular pattern around each candidate corner location, allowing it to rapidly detect corners. The SURF algorithm identifies interest points in an image using a scale-space extrema detection method and uses a unique representation of image patches called a "Haar wavelet response" to efficiently extract feature descriptors. BRIEF is a fast algorithm that selects pairs of pixels at random within a small patch around a given point to compute binary codes for feature descriptors. ORB combines the speed of the FAST corner detector with the robustness of the BRIEF descriptor by using a multi-scale pyramid approach and assigning an orientation to each corner. BRISK is another feature extraction and descriptor algorithm that is scale invariant but has limited invariance to rotation.

The second step after feature extraction is to match these features between the consecutive images. As the matching algorithms essentially depend on the point appearance, some points could share the same appearance although they do not belong to the same object point. Therefore, matching outliers' removal is the third step in the relative pose estimation procedures of computer vision approaches. A Matching outlier detection and filtering algorithm such as the standard RANSAC, Least Median of Squares (LMedS), Normalized eight-point algorithm (Norm8Point), or M-estimator SAmple Consensus (MSAC) are typically used. RANSAC works by randomly selecting a minimal subset of matched points, computing the transformation parameters based on these points, and testing the accuracy of the estimated transformation by counting the number of inliers. LMedS seeks to minimize the median of the squared errors between the matched points and the estimated fundamental matrix. The normalized eight-point algorithm improves the robustness and accuracy of the eight-point algorithm by normalizing the input data. MSAC is an extension of RANSAC that uses an M-estimator function to better handle outliers in the input data.

The final step to estimate the relative pose between two image frames is to decompose the fundamental matrix that relates corresponding points in two views of the same scene. The decomposition of fundamental matrix provides information about the relative orientation and position of the image views.

3.1 Deep learning network

The proposed method for relative pose estimation uses a regression deep neural network to determine the relative vehicle orientation as a function of optical flow between successive frames. The optical flow is the apparent motion of the point or objects of the scene due to the motion of the imaging platform. Optical flow is a computer vision technique that is used to estimate the motion of objects in an image sequence. It analyses the similarities between pixel neighbourhoods along consecutive frames in an image sequence to determine the apparent motion of objects in the scene.

By tracking the motion of pixels over time, optical flow algorithms can estimate the direction and speed of the object's motion. It is defined mathematically as the apparent velocities v_x, v_y of the image point between two frames.

One of the most widely used optical flow algorithms is the Lucas-Kanade method, which computes the flow field by solving a linear system of equations for each pixel in a local neighbourhood. The algorithm assumes that the motion of pixels in the neighbourhood is similar and estimates the flow vector that minimizes the sum of squared differences between the corresponding pixels in the two frames. The Lucas-Kanade method is computationally efficient and can be easily parallelized, making it suitable for real-time applications.

The velocity of the point through the successive frames is affected by many factors: the distance of the point from the moving platform, velocity of the platform, the complexity of the motion in the scene (e.g., cars, pedestrians...etc.). Therefore, predicting pose from the optical flow is not straight forward.

The proposed deep neural network attempts to learn the complex motion within the scene (static objects, moving objects, near objects, far objects, …etc.), overcome the problem of variation in light conditions, and make predictions with good estimates for 3-D pose angular rotation.

To do so, the experiment was implemented using KITTI landmark dataset which is one of the most popular datasets in mobile robotics and autonomous driving. The KITTI dataset is a large-scale benchmark dataset for computer vision tasks related to autonomous driving. It is named after the Karlsruhe Institute of technology and Toyota Technological Institute at Chicago, where it was created. It consists of multiple drives with different traffic scenarios recorded with multiple sensors, including RGB, grey scale stereo cameras, 3D laser scanner along with ground truth data provided by navigation system (OXTS RT3003 inertial and GPS navigation system, 6 axis, 100 Hz, L1/L2 RTK, resolution: 0.02m / 0.1°). Around 76,624 grey scale frame images from KITTI dataset as well as their ground truth navigation data were used for training the proposed network.

The project workflow is as follows:

1. Generate equally spaced grid of points to serve as points of interest other than generating real interest points by either SIFT, SURF, Harris…, or any other operator which is computationally expensive. The grid size was selected to be 10 pixels, which produces (36x121) points in each frame (uniformly distributed). Figure 1. shows the distribution of the point on an image sample.

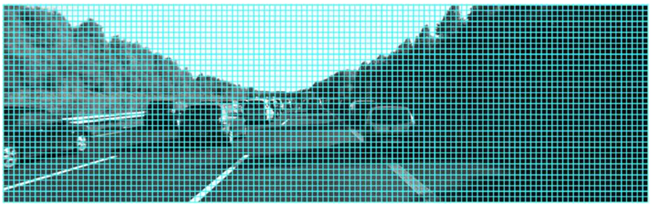
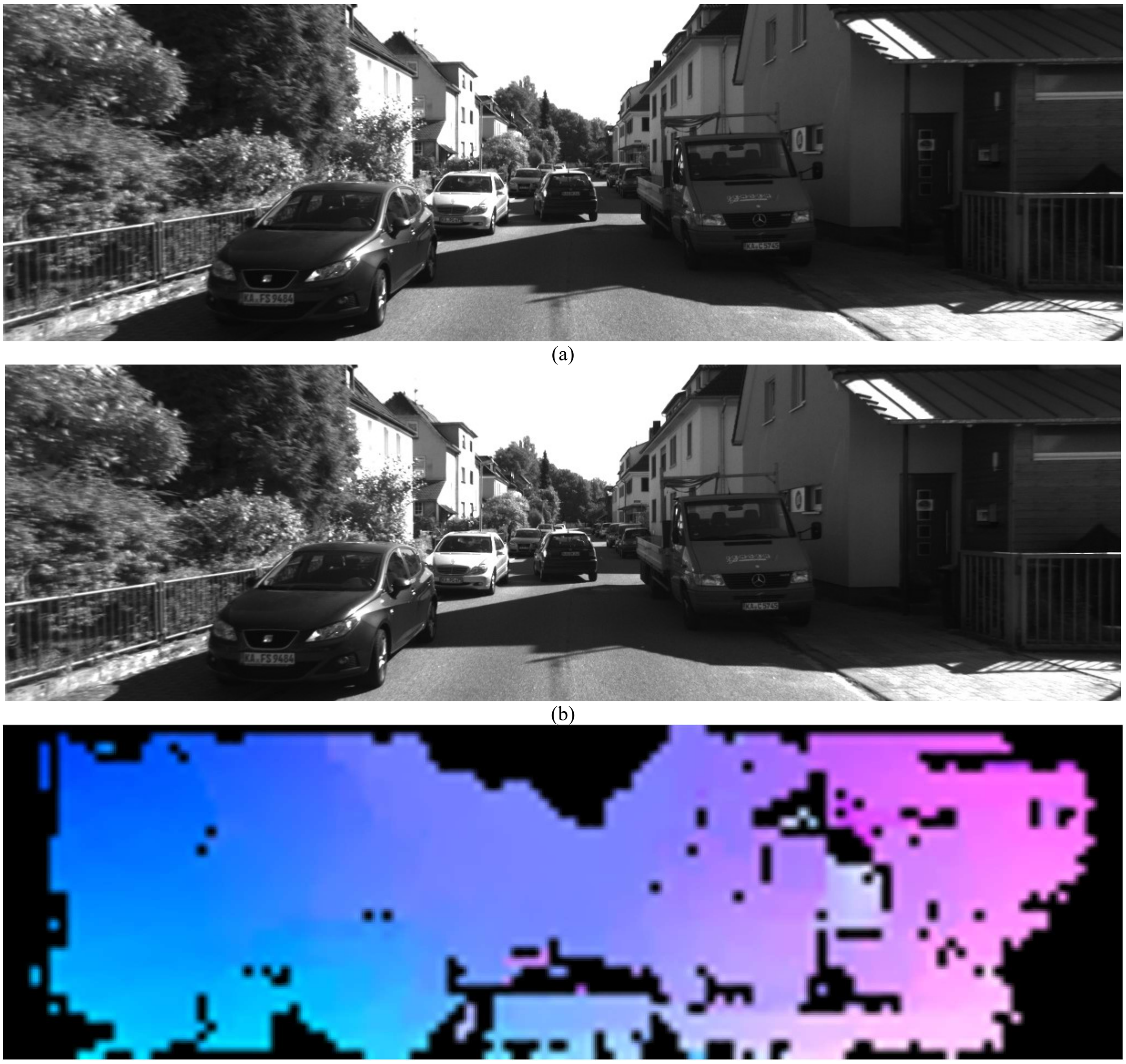


Figure 1. The regular grid of 10-pixel spacing points superimposed on the image.

2. Generate optical flow images for the points generated in step 1. using KANADE-LUCAS-TOMASI (KLT) as a point tracker. Then, the computed (*vx, vy*, confidence score) at each point are used to generate synthetic images (with layers *vx, vy,* score) that represent the optical flow of the original images. Figure 2 shows two successive frames (a), (b) and the generated synthetic image (c) from the optical flow which serves as the input for the proposed network.



(a)

(b)

(c)

Figure 2. The first frame (a), the successive frame (b), and the generated synthetic image of the image pair (c).

3. Generate labels for each frame by computing the relative pose change of vehicle from the ground truth of the navigation data as (Δ *roll*, Δ *pitch*, Δ *heading*)

3.1.1 NETWORK ARCHITECTURE

The proposed network architecture is illustrated in Figure 3. The architecture consists mainly of two convolutional layers each followed by average pooling and two fully connected layers ends with regression of relative orientation of the pose. While deeper architecture can typically improve the network performance, a relatively shallow architecture has been adopted to reduce the computational cost and to meet the real-time requirements on low-cost processing systems. Using KLT based optical flow synthetic images as inputs can significantly help reduce the need for a deeper architecture as a major part of the estimation process has been provided to the network.

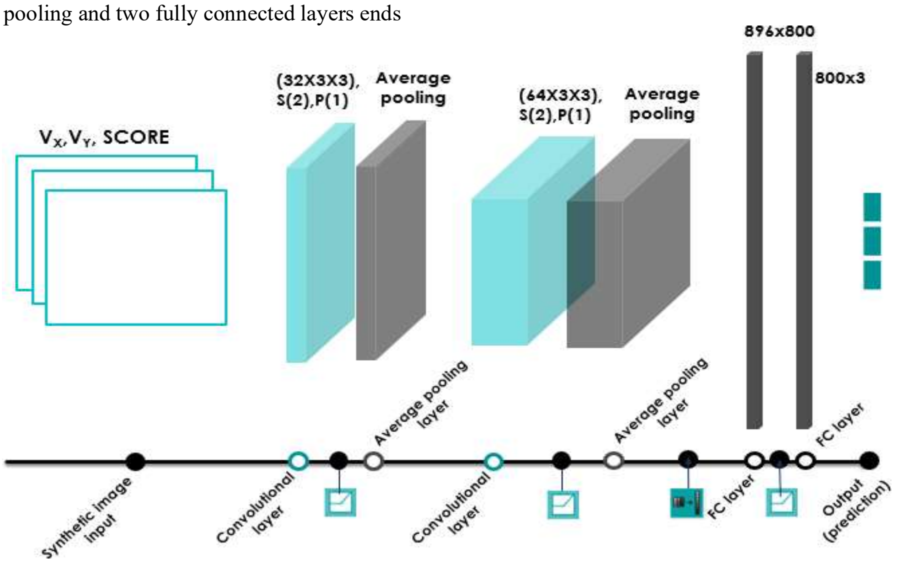


Figure 3. The proposed Network Architecture.

First, the 3-channel synthetic image input (layer for *vx*, other for *vy,* and the third of score) is convolved by the first layer of 32 filters with a spatial extent of 3x3, stride=2 and padding=1. This convolutional layer is followed by a ReLU activation function and an average pooling layer.

The second convolution layer is 64 filters with a spatial extent of 3x3, stride=2 and padding=1. This convolutional layer is also followed by a ReLU activation function and an average pooling layer. Then two fully connected layers with 896 and 800 nodes respectively are used to estimate the three relative angular orientation of pose.

3.1.2 TRAINING NETWORK

The *Loss function* chosen was *Mean Square Error* (MSE) between the model output and the target value defined by the label.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \qquad (1)$$

where MSE = mean squared error
n = number of data points
y_i= predicted values
$\hat{y}_i$= target values

The batch size was chosen to be 128 and the initial learning rate was 1e-5, which has been tuned through several trials to get the optimal value. The stochastic gradient optimization method chosen was Adaptive Moment Estimation (Adam). This method keeps an exponentially decaying average of past gradients. The decay rate was chosen to be 0.01.

3.1.3 THE ACCURACY MEASURES

To evaluate the accuracy predicted relative angular estimates, the following accuracy measures are selected:

1. The *loss function* as MSE between the output and target value Eq .1.
2. The *roll* angle mean error (deg).
3. The *roll* angle standard deviation of error (deg).
4. The *pitch* angle mean error (deg).
5. The *pitch* angle standard deviation of error (deg).
6. The RMSE of relative orientation (deg/frame).
7. The *heading* angle mean error (deg).
8. The *heading* angle standard deviation of error (deg).
9. Accumulated *heading* angle error per minute (deg).

4. EXPERIMENTAL RESULTS AND ANALYSIS

As mentioned before, the first step in relative pose estimation using computer vision approaches is feature extraction. A comparison between processing times for five popular feature extraction algorithms SURF, ORB, FAST, BRISK and SIFT is implemented. The results indicated in table 1 show that the least processing time (around 2 ms) is for FAST algorithm, while the maximum processing time is for BRISK with an average 180 ms.

Table 1. The processing time of different feature extraction and matching algorithms of consecutive frames in (ms)

	Average processing time per pair (ms)
SURF	26.471
ORB	19.198
FAST	2.374
BRISK	179.356
SIFT	129.3767

The second step is to match the extracted features between the successive frames. Figure 4 shows a sample of successive frames coloured as red and cyan. In Figure 4, the matched points with the

SURF features including outliers are indicated by green crosses and red circles. As shown in the figure, there are some outliers in the matched points therefore an outlier detection and removal algorithm is necessary. Figure 5 shows the same images with the correct matched points after outlier points removal with the MSAC algorithm. Table 2 shows the processing time for consecutive frames with different outlier removal algorithms. The standard method RANSAC has the largest processing time, while its variant MSAC converges faster, with approximately half the processing time of RANSAC. The least processing time is for the Norm8Point about 2 ms. However, in practice Norm8Point algorithm cannot be used solely as it does not involve iterative outlier removal and therefore is prone to significant matching errors.

Table 2. Processing time of different outlier removal algorithms processing consecutive frames in (ms)

	Average processing time per pair (ms)
RANSAC	159.344
LMedS	170.428
Norm8Point	2.078
MSAC	54.081

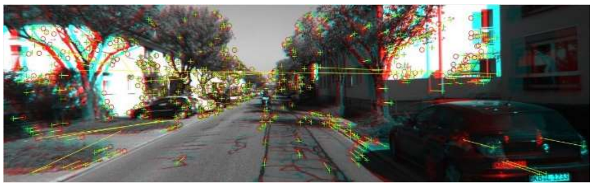


Figure 4. Putatively matched points (including outliers).

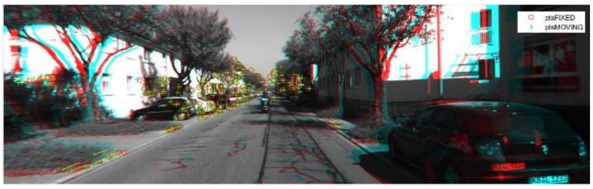


Figure 5. Matched points (inlier only).

4.1 Training for DL Neural Networks

As previously discussed, the computer vision algorithms require enough reliably matched interest points and are sensitive to the environment light conditions. Moreover, the computer vision algorithms are computationally expensive in real time applications as indicated in Table 1 and Table 2. The experiments of the proposed neural network were implemented with a 12th Gen Intel® Core™ i7 Processor H-Series laptop with NVIDIA® GeForce RTX™ 3070 GPU and installed RAM 16GB.

The training involves several trials with different hyperparameters. The final trial consists of 1000 epoch each of which took an average of 6.5 seconds. Table 3. Indicates the results obtained by a test data of 3825 frames. The RMSE of relative orientation is 0.017278 (degree/frame). The largest error component results from the heading angle whose accumulated error per minute drive was -0.696657 degrees.

The results show the prediction time for 3825 frames was 8.4863 (ms), which approximately equals an average of 0.22 (ms) per hundred frames. If we added the average processing time of KLT per frames pair which equals 28 (ms) and compare the results with the shortest possible conventional pipeline as a combination of FAST (2.374 ms) per pair and MSAC outlier detection and removal algorithm (54.081 ms); our proposed method will consume about a half processing time of the conventional method.

Table 3. Results of the proposed pose estimation approach

Total number of test frames	**3825**
RMSE of relative orientation (deg/frame)	0.017278
Relative roll angle mean error (deg/frame)	0.0008966
Relative roll angle std of error (deg)	0.0136598
Relative pitch angle mean error(deg/frame)	0.0039906
Relative pitch angle std of error (deg)	0.0085616
Relative heading angle mean error (deg/frame)	-0.0011611
Relative heading angle std of error (deg)	0.024852
Accumulated heading angle error per minute (deg)	-0.696657
Prediction took (ms)	8.4863

Table 4 shows three different scenarios for processing hundred frame pairs in (ms); the first is our proposed method, the second is a combination of the fastest image detection and matching algorithm (FAST) with the fastest matching outlier and detection algorithm (MSAC) and the third is the most common approach using SIFT as detector and the standard RANSAC as outlier detection and removal algorithm.

Table 4. Processing time in (ms) of different scenarios for processing hundred frame pairs.

Proposed		Scenario 2		Scenario 3	
KLT	2800	FAST	237.4	SIFT	12937.67
NN	0.22	MSAC	5408.1	RANSAC	15934.4
2800.22 (ms)		**5645.5** (ms)		**28872.07** (ms)	

Figure 6 shows the predicted trajectory vs. the ground truth, for the first 1000 frames of a test drive with the accumulated heading angle error per min of -0.696657º. The Figure 7 through Figure 9 show the trajectories of the successive three thousand frames in the same test drive.

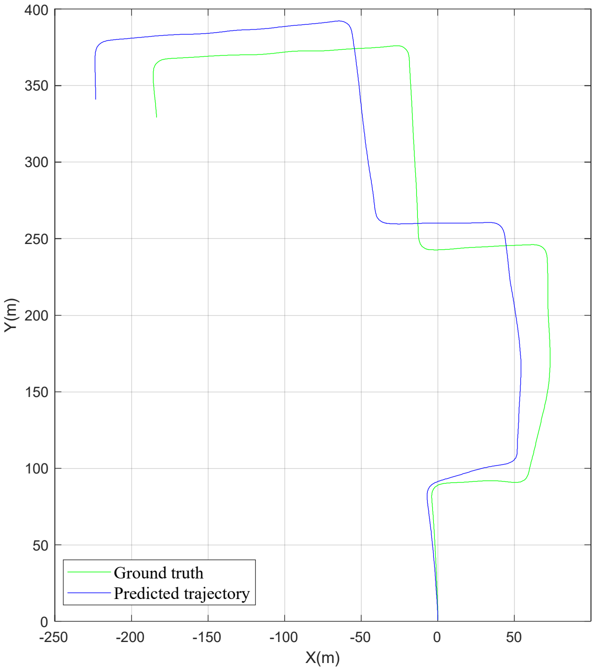


Figure 6. Predicted trajectory vs. Ground truth (Frames 1-1000).

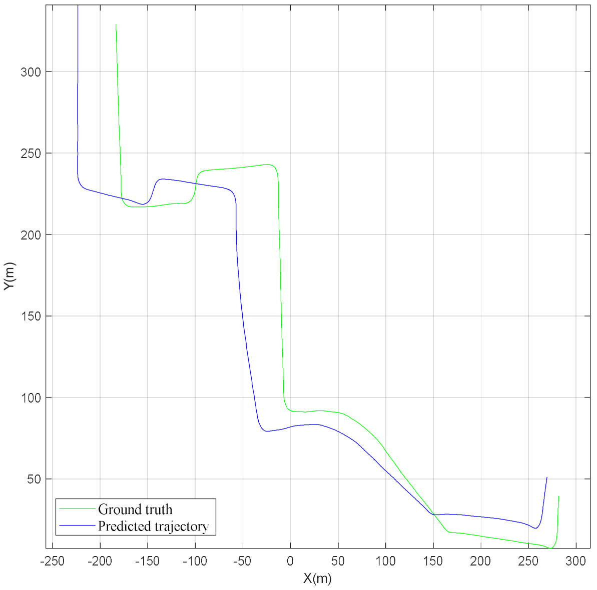


Figure 7. Predicted trajectory vs. Ground truth (Frames 1001-2000).

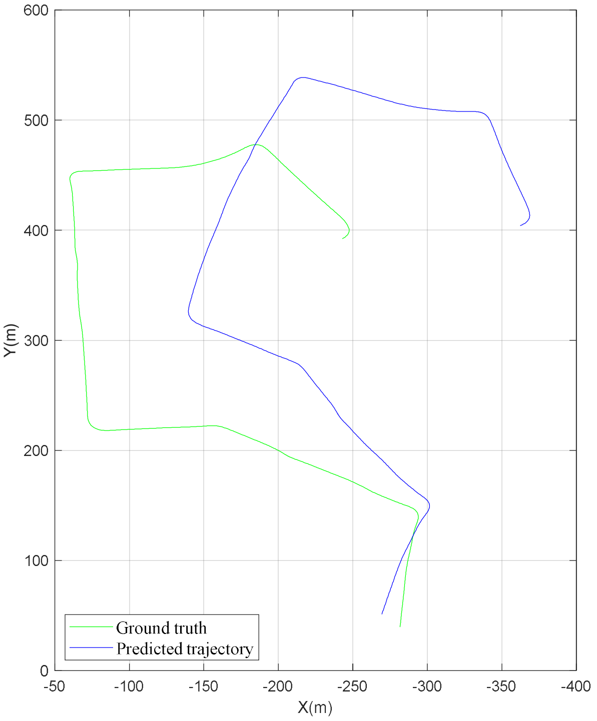


Figure 8. Predicted trajectory vs. Ground truth (Frames 2001-3000).

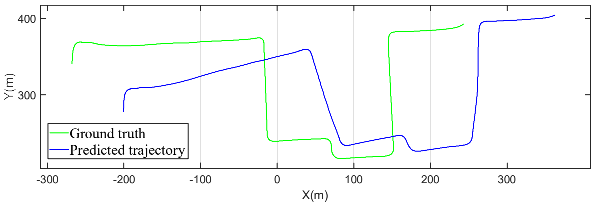


Figure 9. Predicted trajectory vs. Ground truth (Frames 3001-4000).

As shown in the Figures Figure 6 through Figure 9, the drift of the predicted trajectory increases with the number of frames. With fewer degrees of freedom, the system becomes more constrained. Therefore, both errors in attitude and displacement components are accumulated in the resulting attitude estimate. Any slight variations in displacement components can have a relatively substantial impact on the attitude. To improve the performance in the future, the network architecture can be extended to predict displacement components, other than orientation components only.

The behaviour of the proposed network through a whole test drive distance can be shown in the Figure 10 through Figure 12. of sequence 00 is shown in Figure 10. The Figure 10 through Figure 12 show the predicted trajectory vs. ground truth for KITTI drive sequences 00, 05 and 10, respectively.

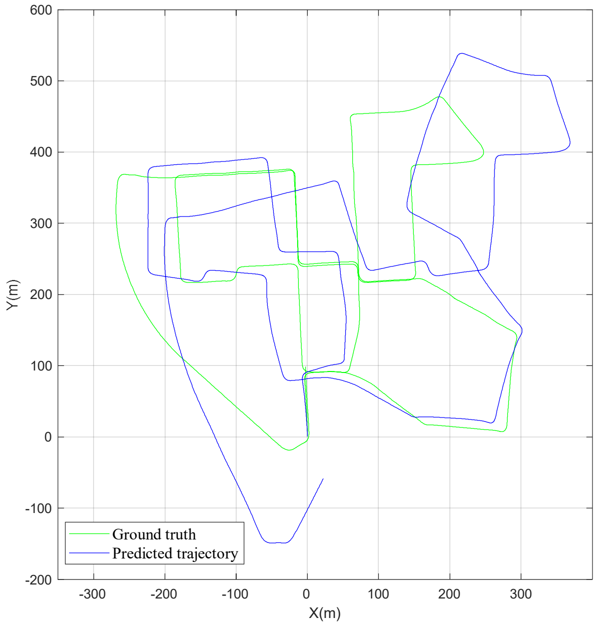


Figure 10. Predicted trajectory vs. Ground truth (Sequence_00).

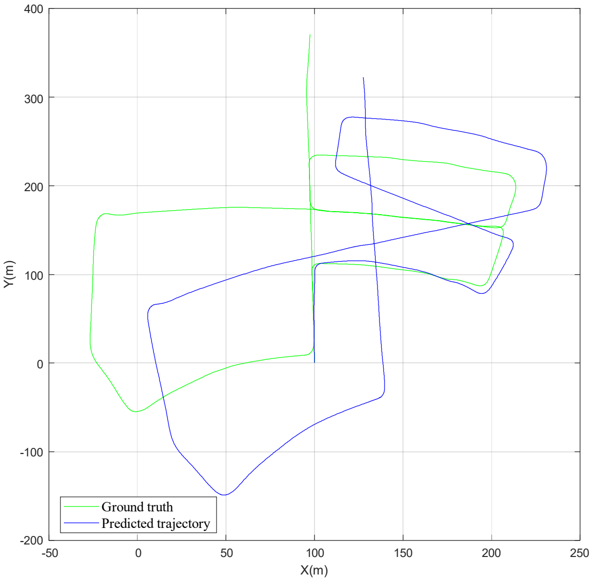


Figure 11. Predicted trajectory vs. Ground truth (Sequence_05).

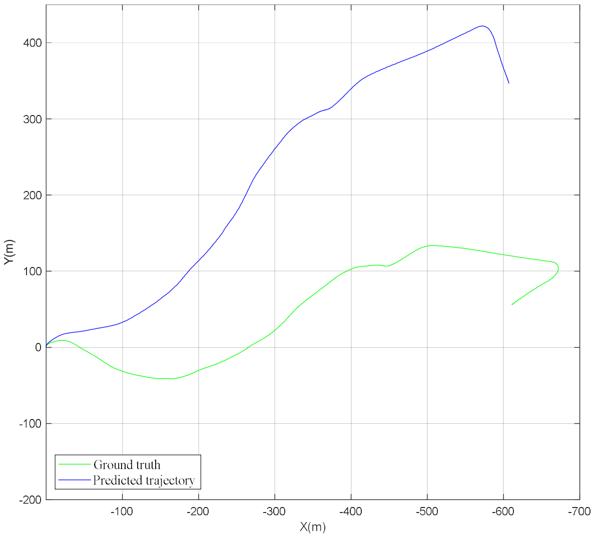


Figure 12. Predicted trajectory vs. Ground truth (Sequence_10).

5. CONCLUSION

The proposed neural network has demonstrated its ability to capture input-output dependencies effectively, resulting in less than one degree of accumulated heading angle error per minute drive, despite being built with few layers to reduce its computational cost.

While computer vision techniques are mathematically based and easy to trace, their pair-wise computations based on feature extraction, matching, and tracking are time-consuming, which is a crucial issue for real-time applications, such as autonomous driving vehicles. On the other hand, deep learning neural networks have emerged as a game changer in solving complex computer vision tasks and delivering predictions within milliseconds. Even though the neural networks require tons of data to learn; but once they learn, prediction can be achieved in a relatively short time, especially if simple architectures are adopted.

The proposed approach could deal with the confusing scenarios of the moving objects, occlusions, and changes in lighting conditions. The proposed network can estimate the relative attitude of the vehicle with a RMSE of relative orientation with 0.017 degrees per frame. The prediction time takes 0.22 milliseconds per hundred frames. This minimized computational cost can meet the real-time requirements on low-cost processing systems. Further tuning of the grid parameters and effective region of interests can significantly reduce the computational burden of the proposed approach.

For future work, the network architecture can be modified to incorporate Recurrent with CNN architecture to capture the sequential dependencies of frames as the pose changes gradually over time. This modification can enhance the results further. Additionally, an extension to predict displacement components, other than orientation components only, can be explored.

ACKNOWLEDGMENT

This research has been supported by funding of Prof. Naser El-Sheimy from NSERC CREATE and Canada Research Chairs programs.

REFERENCES

Lowe, D. G. ,2004. Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision, 60(2), 91–110. doi.org/10.1023/B: VISI.0000029664.99615.94

E. Rosten and T. Drummond, 2006. “Machine learning for high-speed corner detection,” in Proc. Eur. Conf. Comput. Vis., Graz, Austria, pp. 430–443.

H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, 2008. “Surf: Speeded up robust features,” Comput. Vis. Image Understand., vol. 110, no. 3, pp. 346–359, 2008.

E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, 2011. “ORB: An efficient alternative to SIFT or SURF,” in Proc. Int. Conf. Comput. Vis., Barcelona, Spain, pp. 2564–2571.

M. Calonder, V. Lepetit, C. Strecha, and P. Fua, 2010. “Brief: Binary robust independent elementary features,” in Proc. Eur. Conf. Comput. Vis., Crete, Greece, pp. 778–792.

Tomasi, Carlo, and Takeo Kanade, 1991. "Detection and tracking of point." Int J Comput Vis 9, pp.137-154.

M. A. Fischler and R. C. Bolles, 1981. “RANSAC sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography,” Commun. ACM, vol. 24, no. 6, pp. 381–395.

P. Torr and D. Murray, 1997. “The development and comparison of robust methods for estimating the fundamental matrix,” Int. J. Comput. Vis., vol. 24, no. 3, 1997, pp. 271–300.

K. Sim and R. Hartley, 2006. “Recovering camera motion using l1 minimization,” IEEE Conf. Computer Vision and Pattern Recognition, 2006, pp. 1230–1237.

Scaramuzza, Davide, 2009. “Real-Time Monocular Visual Odometry for on-Road Vehicles with 1-Point RANSAC.” 2009 IEEE International Conference on Robotics and Automation, IEEE, pp. 4293–99.

J. Engel, J. Sturm, and D. Cremers, 2013. “Semi-dense visual odometry for a monocular camera,” in Proc. IEEE Int. Conf. Comput. Vis., Sydney, NSW, Australia, pp. 1449–1456.

C. Forster, M. Pizzoli, and D. Scaramuzza, 2014. “SVO: Fast semi-direct monocular visual odometry,” in Proc. IEEE Int. Conf. Robot. Autom. (ICRA), Hong Kong, pp. 15–22.

A. Kendall, M. Grimes, and R. Cipolla, 2015. “PoseNet: A convolutional network for real-time 6-DOF camera relocalization,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Santiago, Chile, pp. 2938–2946.

A. Dosovitskiy ,2015. “FlowNet: Learning optical flow with convolutional networks,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Santiago, Chile, pp. 2758–2766.

Z. Yin and J. Shi, 2018. “GeoNet: Unsupervised learning of dense depth, optical flow and camera pose,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Salt Lake City, UT, USA, pp. 1983–1992

A. Valada, N. Radwan, and W. Burgard, 2018. “Deep auxiliary learning for visual localization and odometry,” in Proc. IEEE Int. Conf. Robot. Autom. (ICRA), Brisbane, QLD, Australia, May 2018, pp. 6939–6946.