E-CNN: ACCURATE SPHERICAL CAMERA ROTATION ESTIMATION VIA UNIFORMIZATION OF DISTORTED OPTICAL FLOW FIELDS

Dabae Kim, Sarthak Pathak, Alessandro Moro, Ren Komatsu, Atsushi Yamashita, and Hajime Asama

Department of Precision Engineering, The University of Tokyo, Japan

ABSTRACT

Spherical cameras, which can acquire all-round information, are effective to estimate rotation for robotic applications. Recently, Convolutional Neural Networks have shown great robustness in solving such regression problems. However they are designed for planar images and cannot deal with the non-uniform distortion present in spherical images, when expressed in the planar equirectangular projection. This can lower the accuracy of motion estimation. In this research, we propose an Equirectangular-Convolutional Neural Network (E-CNN) to solve this issue. This novel network regresses 3D spherical camera rotation by uniformizing distorted optical flow patterns in the equirectangular projection. We experimentally show that this results in consistently lower error as opposed to learning from the distorted optical flow.

Index Terms— Spherical camera, distortion, optical flow, CNN, rotation estimation

1. INTRODUCTION

Camera-based rotation estimation is essential for robotic applications like visual compassing [1], structure from motion [2], and visual odometry [3]. As opposed to perspective cameras, spherical cameras can acquire information from a much wider area. This spherical camera property has been shown to improve accuracy of motion/pose estimation [4].

Recently, Convolutional Neural Networks (CNNs) have shown powerful abilities even in complex regression problems such as camera motion estimation [5, 6, 7] and camera relocalization [8]. Particularly, they show great potential in estimating camera motion against different environmental conditions in a robust manner by learning from a large amount of data, whereas a conventional feature-based approach may fail due to the features (points, lines, etc.) being designed for specific types of scenarios. However these learning-based approaches work only on planar images. This is because CNNs depend on the convolution operation which is designed for planar matrices on uniform grids. Since spherical cameras acquire images projected on a spherical surface, it is difficult to construct a uniform grid. Spherical images can be expressed as planar equirectangular images at the cost of non-uniform distortion, as shown in Fig. 1. Typically, it is expected that since spherical images contain information from all directions, they can be rotated to any orientation without any effect on the image data. However, this is not true for the equirectangular projection due to the non-uniform distortion, which can make it difficult to learn from equirectangular images.

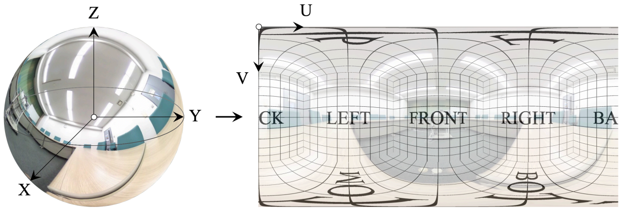

Fig. 1. Spherical images, when expressed in the equirectangular projection, contain distortion, especially towards the top and the bottom.

In this research, we solve this issue via a novel network which we call the *Equirectangular-Convolutional Neural Network (E-CNN)* which can regress the 3D rotation of a spherical camera by uniformizing the effect of the distortion. Specifically, we estimate the 3D rotation between two spherical image frames by the uniformization of distorted dense optical flow patterns. The reason for using dense optical flow is because it has been shown to induce less scene dependency [7] as compared to directly using the intensity values.

2. RELATED RESEARCH AND OUR APPROACH

There are many approaches in literature that attempt to deal with the distortion of spherical images. Spherical SIFT [9, 10] constructs a scale space directly on the spherical surface. A-KAZE [11] is a popular feature descriptor that is robust to distortions via the creation of a non-linear scale space.

There have also been attempts at introducing uniform grids for spherical images [12, 13] via interpolation, which could potentially work with CNN-based approaches. Spher-

This work was in part supported by Council for Science, Technology and Innovation, "Cross-ministerial Strategic Innovation Promotion Program (SIP), Infrastructure Maintenance, Renovation, and Management" (funding agency: NEDO). Contact:{kimdabae, pathak, moro, komatsu, yamashita, asama}@robot.t.u-tokyo.ac.jp

ical convolution [14, 15] has also been attempted. However, the above approaches are still in their infancy and are too computationally heavy for use with the thousands of operations that take place during the CNN learning process.

Other techniques attempt to work with equirectangular images while keeping spherical image geometry in mind. In [16], the authors attempt to learn distance from spherical images in the equirectangular projection. They reason that the yaw orientation of the camera should not affect distance. Accordingly, they use roll-branching to redistribute the information around the yaw axis during the learning and nullify its effect. In a similar light, [17] pitch rotates spherical images and picks features from the central region during each rotation, in order to avoid distorted parts. [18] also attempts a stereo rectification within the equirectangular projection for depth-based motion estimation.

In this research, we focus on the use dense optical flow to do so as it has been shown to have low scene dependency. Due to the distortion, the optical flow patterns in the equirectangular projection are non-uniform. We handle this distortion by uniformizing the optical flow patterns via rotation of spherical images, which can be done without losing any information.

3. E-CNN: OPTICAL FLOW PATTERNS UNIFORMIZATION NETWORK

The inputs to our method consist of two spherical image frames with some rotation between them. We construct E-CNNs to learn the dense optical flow patterns with respect to 3D rotation angles. As mentioned earlier, the distortion of the equirectangular image affects dense optical flow patterns. This can be understood as follows. On the spherical projection, roll, pitch, and yaw rotations are equivalent. However, on the equirectangular projection, the optical flow due to yaw rotation shows straight lined patterns, whereas roll and pitch rotations show curved patterns, as shown in Fig. 2. In other words, for equivalent spherical camera rotations, the optical flow patterns appearing on the equirectangular image would be a mixture of the lined and curved patterns. This non-uniformity in the optical flow patterns is difficult to learn and can interrupt accurate estimation.

We uniformize these patterns in order to make learning easier. This is done by taking advantage of the fact that spherical images contain information from all directions and can hence be rotated to any desired orientation. The E-CNN (Equirectangular-Convolutional Neural Network) consists of 2 networks: the uniformization network, that uniformizes the patterns, and the feature extraction network, that regresses the rotation, as shown in Fig. 4.

The uniformization network takes the two image frames as input and rotates them in the directions of roll and pitch by 90 degrees, in order to generate two additional image pairs. Then, it calculates the dense optical flow for every frame pair - the original (non-rotated), roll-rotated, and pitch-rotated. This essentially results in equalizing the proportion of the lined and curved optical flow patterns for all rotations. The state of the uniformized optical flow patterns is shown in Fig. 3. Every optical flow has 2 channels, expressing the horizontal and vertical components, creating 6 channels in total.

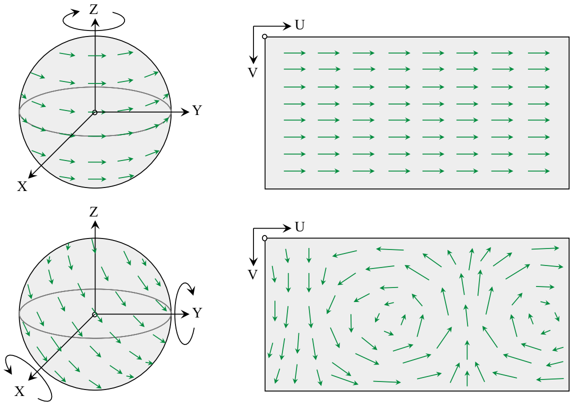

Fig. 2. Lined optical flow patterns for yaw rotation (top). Curved optical flow patterns for roll and pitch rotations (bottom).

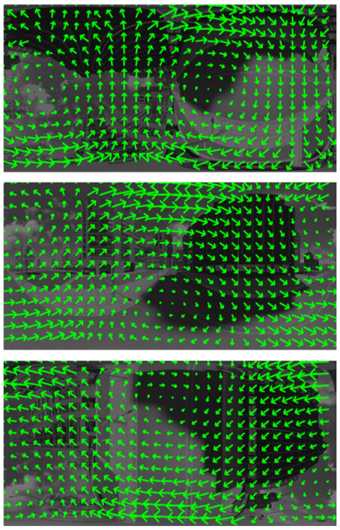

Fig. 3. Equirectangular optical flow uniformization: roll-rotated optical flow (top), original optical flow (middle), and pitch-rotated optical flow (bottom).

The next part of the network is the feature extraction network that regresses 3D rotation via an end-to-end CNN. The feature extractor has the following blocks: conv1/2/3$_{[64]}$, pool1, conv4/5/6$_{[128]}$, pool2, conv7/8/9$_{[256]}$, pool3, conv10/11 /12$_{[512]}$, pool4, fc$_{[1024]}$, fc$_{[4]}$. The notation conv$_{[c]}$ is a convolution layer with c filters of size 3×3 with stride 1×1 and a ReLU [19] activation layer, pool is a max-pooling layer of size 2×2 with stride 2×2, and fc$_{[n]}$ is a fully-connected layer with n nodes.

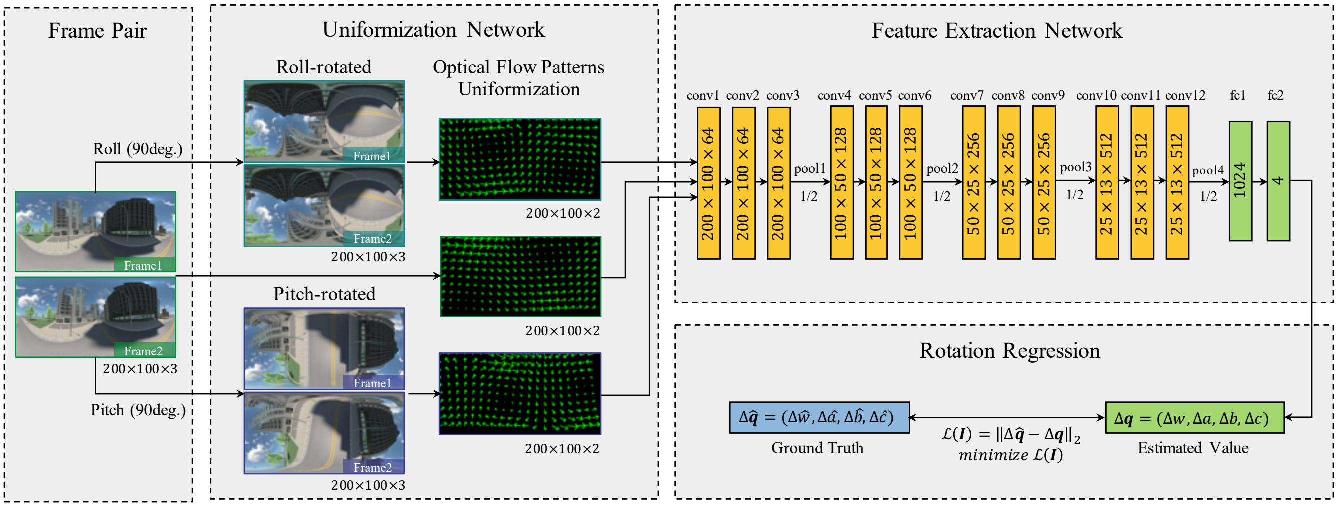

Fig. 4. E-CNN Structure: The Uniformization network rotates the images to uniformize the optical flow patterns. The Feature Extraction network takes the uniformized optical flow as input and regresses the 3D rotation.

Weights are updated using the euclidean distance between the ground truth and the estimated value, with the following loss function:

$$\mathcal{L}(\boldsymbol{I}) = \|\Delta\hat{\boldsymbol{q}} - \Delta\boldsymbol{q}\|_2. \quad (1)$$

The loss function represents the L2 error between the ground truth $\Delta\hat{\boldsymbol{q}}$ and the estimated value $\Delta\boldsymbol{q}$ on the unit sphere in quaternion space. The input to the network is the 6-channel uniformized optical flow $\boldsymbol{I}$.

We refer to this as E-CNN because it works completely within the equirectangular projection. In the next section, we verify the effectiveness of this proposed network via training and testing in different conditions and environments.

4. EXPERIMENTAL VERIFICATION

Experiments to verify the effectiveness of the proposed network were conducted on data gathered via both simulation and real (indoor/outdoor) environments. For dense optical flow calculation, we used the DeepFlow [20] method.

For the simulation environment, the Blender software was adopted, and an urban environment (obtained from [4]) was used. The dataset construction consisted of 3 steps. Firstly, 10 scenes (no. 1–10) were set for training and another 10 scenes (no. 11–20) for validation and testing. Each scene indicates a particular point in the simulation environment from where spherical equirectangular images of resolution of 200×100 pixels were generated. The initial pose of each training sample was randomly set in order to provide a generality to the learning. In addition, the rotation from the initial pose was limited to 0 to 10 degrees in each axis (roll, pitch, and yaw) in order to ensure that optical flow calculation was possible. Training data was generated with 0.5 degrees increments in every axis (roll, pitch, and yaw). The total number of the training data samples amounted to 9,261.

Meanwhile, the validation/test data were generated from scenes at different locations from the training data. As opposed to the training data which was collected in steps of 0.5 degrees, the rotation angles for the validation/test data were chosen randomly from within 0 to 10 degrees in each axis. The validation/test sets contained 1,000 image pairs, each. The dataset composition is summarized in Table 1.

In addition to the simulated environment, real images were also captured in indoor/outdoor environments using the Ricoh Theta S spherical camera, as shown in Fig. 5. The dataset composition was basically the same as that of simulation environment. However, unlike the simulated dataset, the real images were rotated manually by projection to a spherical surface, and the ground truth rotations were recorded. As earlier, the training and testing were done at different locations.

In order to evaluate the effect of the proposed E-CNN, it was compared to a network of similar size without the uniformization network. The original 2-channel optical flow was copied to produce 6 channels, the same input size as the E-

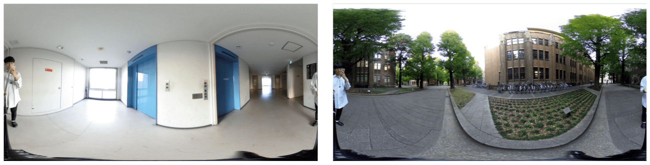

Fig. 5. Indoor (left) / outdoor (right) real environments

Table 1. Dataset composition

No. of samples	Training data	Validation data	Test data
Scene (simulation)	1 – 10	11 – 20	11 – 20
Scene (real)	1 – 20	21 – 40	21 – 40
Data quantity	9,261	1,000	1,000

Table 2. Estimation results

Avg. errors [deg.]	Non-uniformized	Uniformized (E-CNN)
Simulation	0.182 ±0.104	**0.144 ±0.069**
Real	0.144 ±0.113	**0.104 ±0.077**

CNN network, in order to allow for a fair comparison. Adam [21] was adopted as the optimizer. The learning process was conducted for 500 epochs with a fixed learning rate (0.001) and a batch size of 150, on a NVIDIA GeForce GTX 1080Ti (GPU) and an Intel Xeon E5-1650 v4 (CPU).

For both networks, the E-CNN and the non-uniformized network, all conditions and parameters were the same. The estimation results in both simulation and real environments are shown in Table 2. The errors were computed by considering only the angle in the angle-axis configuration. It can be seen that in both environments, simulated and real, the E-CNN network increased the accuracy of rotation estimation by about 20.9% and 27.8%, respectively.

In order to further study the effectiveness of our proposed E-CNN, an experiment was conducted with five cases of varying network depth, as detailed in Table 3. For this experiment, the real environmental data was used for training, validation, and testing. The results are shown in Fig. 6. In all cases, the E-CNN showed an improvement in the accuracy, as indicated by the lower error, and higher precision, as indicated by the lower standard deviation. Moreover, the E-CNN performed better with increasing network depth, indicating that it enables the network to properly learn the optical flow patterns.

5. DISCUSSION AND CONCLUSION

In this research, we constructed a novel E-CNN for accurate spherical camera rotation estimation. We aimed to handle the non-uniform, distorted optical flow patterns created when using the equirectangular projection by uniformizing them with respect to every rotation axis.

The effectiveness of our proposed E-CNN was experimentally verified in both simulation and real (indoor/outdoor) environments. Furthermore, it was shown that this improvement was consistent, irrespective of the network depth. In all cases, the E-CNN estimated rotation more accurately than the naive non-uniformized approach. In addition to the increased accuracy, the E-CNN also demonstrated lower standard deviation, which implies higher precision.

Table 3. Cases with various networks depths (no. 1–5): filter size, stride, and activation function are the same as explained in Section 3.

Case	Architecture blocks
case1 *(4 lyrs.)*	conv1$_{[64]}$, pool1, conv2$_{[128]}$, pool2, conv3$_{[256]}$, pool3, conv4$_{[512]}$, pool4, fc$_{[1024]}$, fc$_{[4]}$
case2 *(6 lyrs.)*	conv1/2$_{[64]}$, pool1, conv3/4$_{[128]}$, pool2, conv5$_{[256]}$, pool3, conv6$_{[512]}$, pool4, fc$_{[1024]}$, fc$_{[4]}$
case3 *(8 lyrs.)*	conv1/2$_{[64]}$, pool1, conv3/4$_{[128]}$, pool2, conv5/6$_{[256]}$, pool3, conv7/8$_{[512]}$, pool4, fc$_{[1024]}$, fc$_{[4]}$
case4 *(10 lyrs.)*	conv1/2$_{[64]}$, pool1, conv3/4$_{[128]}$, pool2, conv5/6/7$_{[256]}$, pool3, conv8/9/10$_{[512]}$, pool4, fc$_{[1024]}$, fc$_{[4]}$
case5 *(12 lyrs.)*	conv1/2/3$_{[64]}$, pool1, conv4/5/6$_{[128]}$, pool2, conv7/8/9$_{[256]}$, pool3, conv10/11/12$_{[512]}$, pool4, fc$_{[1024]}$, fc$_{[4]}$

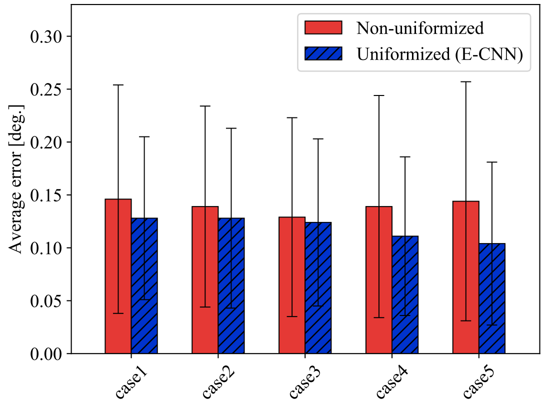

Fig. 6. Average errors for various network depths

An additional experiment was conducted to evaluate the effect of the network size on the estimation. Even deeper CNN architectures were unable to learn the distorted optical flow patterns. However, this problem was solved by our proposed E-CNN. This approach was feasible because 360-degree images record information from all directions, and hence, can be rotated without loss of information.

One drawback of the proposed approach is that it cannot perform under situations of large camera displacements as it requires the input of dense optical flow. This is a trade-off against learning from raw intensities, which can handle large displacements as well, but introduce scene dependency. Also, the scenarios were restricted to only rotation, whereas cameras can also translate arbitrarily in real situations. Taking these into consideration and solving the scene dependency problem remain as future work.

6. REFERENCES

[1] G. Caron and F. Morbidi, “Spherical visual gyroscope for autonomous robots using the mixture of photometric potentials,” in *Proceedings of the 2018 IEEE International Conference on Robotics and Automation*, 2018, pp. 820–827.

[2] A. Pagani and D. Stricker, “Structure from motion using full spherical panoramic cameras,” in *Proceedings of the 2011 IEEE International Conference on Computer Vision Workshops*, 2011, pp. 375–382.

[3] C. Forster, Z. Zhang, M. Gassner, M. Werlberger, and D. Scaramuzza, “Svo: Semidirect visual odometry for monocular and multicamera systems,” *IEEE Transactions on Robotics*, vol. 33, no. 2, pp. 249–265, 2017.

[4] Z. Zhang, H. Rebecq, C. Forster, and D. Scaramuzza, “Benefit of large field-of-view cameras for visual odometry,” in *Proceedings of the 2016 IEEE International Conference on Robotics and Automation*, 2016, pp. 801–808.

[5] G. Constante, M. Mancini, P. Valigi, and T. A. Ciarfuglia, “Exploring representation learning with cnns for frame-to-frame ego-motion estimation,” *IEEE Robotics and Automation Letters*, vol. 1, no. 1, pp. 18–25, 2016.

[6] S. Wang, R. Clark, H. Wen, and N. Trigoni, “Deepvo: Towards end-to-end visual odomtery with deep recurrent convolutional neural networks,” in *Proceedings of the 2017 IEEE International Conference on Robotics and Automation*, 2017, pp. 2043–2050.

[7] F. Guo, Y. He, and L. Guan, “Deep camera pose regression using motion vectors,” in *Proceedings of the 25th IEEE International Conference on Image Processing*, 2018, pp. 4073–4077.

[8] A. Kendall, M. Grimes, and R. Cipolla, “Posenet: A convolutional network for real-time 6-dof camera relocalization,” in *Proceedings of the 2015 IEEE International Conference on Computer Vision*, 2015, pp. 2938–2946.

[9] J. Cruz-Mota, I. Bogdanova, B. Paquier, M. Bierlaire, and J. P. Thiran, “Scale invariant feature transform on the sphere: Theory and applications,” *International Journal of Computer Vision*, vol. 98, no. 2, pp. 217–241, 2012.

[10] P. Hansen, W. Boles, and P. Corke, “Spherical diffusion for scale-invariant keypoint detection in wide-angle images,” in *Proceedings of the 2008 Conference on Digital Image Computing: Techniques and Applications*, 2008, pp. 525–532.

[11] P. Alcantarilla, J. Nuevo, and A. Bartoli, “Fast explicit diffusion for accelerated features in nonlinear scale spaces,” in *Proceedings of the British Machine Vision Conference*, 2013, pp. 13.1–13.11.

[12] J. D. Adarve and R. Mahony, “Spherepix: A data structure for spherical image processing,” *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 483–490, 2017.

[13] H. G. and W. A. P. Smith, “Brisks: Binary features for spherical images on a geodesic grid,” in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4886–4894.

[14] R. Khasanova and P. Frossard, “Graph-based classification of omnidirectional images,” in *Proceedings of the 2017 IEEE International Conference on Computer Vision Workshops*, 2017, pp. 860–869.

[15] Y. C. Su and K. Grauman, “Learning spherical convolution for fast features from 360° imagery,” in *Proceedings of the 31st Conference on Neural Information Processing Systems*, 2017, pp. 529–539.

[16] T. Wang, H. Huang, J. Lin, C. Hu, K. Zeng, and M. Sun, “Omnidirectional cnn for visual place recognition and navigation,” in *Proceedings of the 2018 IEEE International Conference on Robotics and Automation*, 2018, pp. 2341–2348.

[17] H. Taira, Y. Inoue, A. Torii, and M. Okutomi, “Robust feature matching for distored projection by spherical cameras,” *IPSJ Transactions on Computer Vision and Applications*, vol. 7, pp. 84–88, 2015.

[18] S. Pathak, A. Moro, H. Fujii, A. Yamashita, and H. Asama, “Distortion-robust spherical camera motion estimation via dense optical flow,” in *Proceedings of the 25th IEEE International Conference on Image Processing*, 2018, pp. 3358–3362.

[19] V. Nair and G. E. Hinton, “Rectified linear units improve restricted boltzmann machines,” in *Proceedings of the 27th international conference on machine learning*, 2010, pp. 807–814.

[20] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid, “Deepflow: Large displacement optical flow with deep matching,” in *Proceedings of the 2013 IEEE International Conference on Computer Vision*, 2013, pp. 1385–1392.

[21] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *Proceedings of the 2015 International Conference on Learning Representations*, 2015.