Automated Camera Calibration via Homography Estimation with GNNs

Giacomo D'Amicantonio, Egor Bondarev, Peter H.N. De With
Eindhoven University of Technology
{ g.d.amicantonio, e.bondarau, p.h.n.de.with } @tue.nl

Abstract

Over the past few decades, a significant rise of camera-based applications for traffic monitoring has occurred. Governments and local administrations are increasingly relying on the data collected from these cameras to enhance road safety and optimize traffic conditions. However, for effective data utilization, it is imperative to ensure accurate and automated calibration of the involved cameras. This paper proposes a novel approach to address this challenge by leveraging the topological structure of intersections.

We propose a framework involving the generation of a set of synthetic intersection viewpoint images from a bird's-eye-view image, framed as a graph of virtual cameras to model these images. Using the capabilities of Graph Neural Networks, we effectively learn the relationships within this graph, thereby facilitating the estimation of a homography matrix. This estimation leverages the neighbourhood representation for any real-world camera and is enhanced by exploiting multiple images instead of a single match. In turn, the homography matrix allows the retrieval of extrinsic calibration parameters. As a result, the proposed framework demonstrates superior performance on both synthetic datasets and real-world cameras, setting a new state-of-the-art benchmark.

1. Introduction

Camera calibration is a crucial aspect of computer vision (CV) applications, enabling the mapping of pixels to real-world coordinates. It serves as a prerequisite for various CV tasks, including object localization and immersive imaging. While calibrating cameras with a checkerboard pattern is now considered straightforward, this method is not always practical in real-world scenarios, especially when dealing with traffic-control cameras which are placed in busy intersections or in highways. These cameras are typically mounted on light posts and traffic lights, making them susceptible to slight movements caused by environmental factors and movement of large vehicles. Consequently, the original camera calibration may be affected by these movements, necessitating frequent and automated re-calibration. However, accessing these cameras in person is often unfeasible because of their locations at intersections or highways.

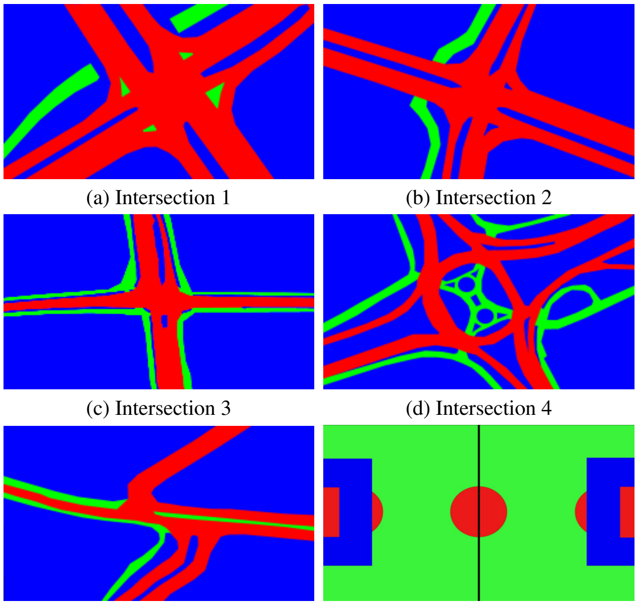

(a) Intersection 1
(b) Intersection 2
(c) Intersection 3
(d) Intersection 4
(e) Intersection 5
(f) Soccer field

Figure 1. Intersection images from [8]. Bird's-eye-views of the semantical segmentations of the intersections: Red = road, Blue = terrain and Green = bicycle path. The intersections differ in the complexity of their topology. Semantic segmentation of the soccer field: Red = goal box, penalty arc and center circle, Blue = penalty box, Green = rest of the pitch.

While deep learning techniques have been explored to address this problem, they often demand substantial data and computational resources, posing limitations for startups and small companies. As of today, there is no industry standard for achieving cost-effective, accurate, and reliable automated camera calibration. To this end, this paper introduces a homography estimation approach that can be easily trained on synthetic data and performs effectively in real-world settings.

The proposed method leverages the topological structure of intersections by creating a graph of synthetic templates from virtual cameras. Each template is generated by warping the bird's-eye-view (BEV) of the scene with a homography derived from each virtual camera. Using this graph, we find the best match between the camera image and a set of synthetic templates. This matching process is framed as a link-prediction task, executed by a Graph Neural Network (GNN) [11, 29].

The proposed model predicts a probability score to select the top-k closest matches for the input image. The embeddings learned by the GNN are then processed to regress a homography that transforms the input image to the highest scoring template, whose homography serves as the anchor. The homography estimation is performed by a Spatial Transformer Network (STN) [15].

Notably, this framework has two advantages. First, it estimates the homography between an image and the BEV by leveraging multiple views of the scene, rather than relying on a single image. Second, the framework is more computationally efficient than previous approaches, while outperforming them on synthetic datasets and real-world cameras. To the best of our knowledge, this is the first work to explore the use of GNNs for homography estimation.

The paper structure is as follows. Section 2 provides an overview of current camera calibration approaches and the literature on Graph Neural Networks. Section 3 details the models and the designed pipeline. The results on multiple intersections and five real-world cameras are presented in Section 4. Finally, Section 5 concludes the paper and indicates potential future work.

2. Related Work

2.1. Camera Calibration

Homography estimation typically involves exploitation of various image features and characteristics. The approaches mentioned in related work [2, 4, 21, 22, 27, 31, 35] extract and match these features in images and then estimate the homography matrix via Direct Linear Transformation (DLT) [13]. To handle erroneous matches, outlier detectors like RANSAC [10] or the more recent MAGSAC, [1] are employed. However, these methods are not tailored specifically for traffic scenes and lack full automation.

In recent years, deep learning-based approaches have emerged as a powerful alternative. DeTone *et al.* [7] were among the first to propose a simple end-to-end network for direct homography estimation. Subsequent works have built further upon this foundation, using deeper and more complex networks to achieve better performance [9, 18, 36]. The introduction of the Spatial Transformer Network (STN) [15] sparked the development of new approaches, such as the works in [8, 25, 30]. Addressing automated camera calibration, Bhardwaj *et al.* [3] presented a prominent method which exploits the presence of vehicles on the road, by employing vehicle-detection techniques, key-point extractors, and prior knowledge of the vehicle's geometrical properties. Another significant work in this area is reported in [37], where the authors propose two camera calibration methods. One method optimizes an energy function to minimize the 3D re-projection error, while the other leverages synthetic data. It is worth noting that although the mentioned approaches have shown success in various settings, none of them are specifically designed to handle the challenges unique to traffic scenes, *i.e.* the amount of vehicles in the scene and varying weather conditions.

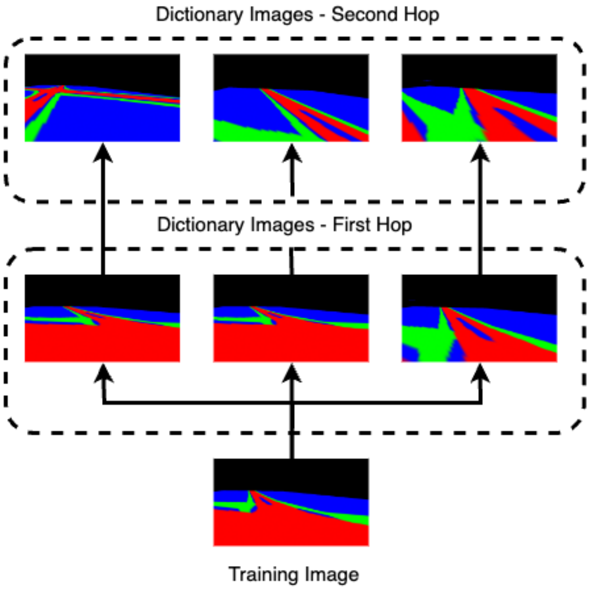


Figure 2. Example of the synthetic training and dictionary images. It is relevant to notice that in the second hop there can be images that are not visually similar to the training images or the representation of the corresponding part of the scene.

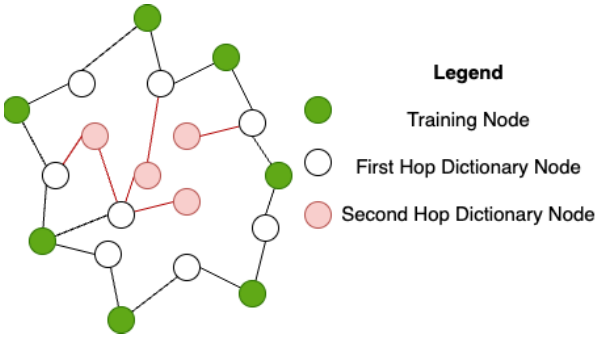


Figure 3. Reduced representation of a sub-graph of the generated graph. Each node represents an image and is connected to the 20 dictionary nodes with the lowest Topological Loss score.

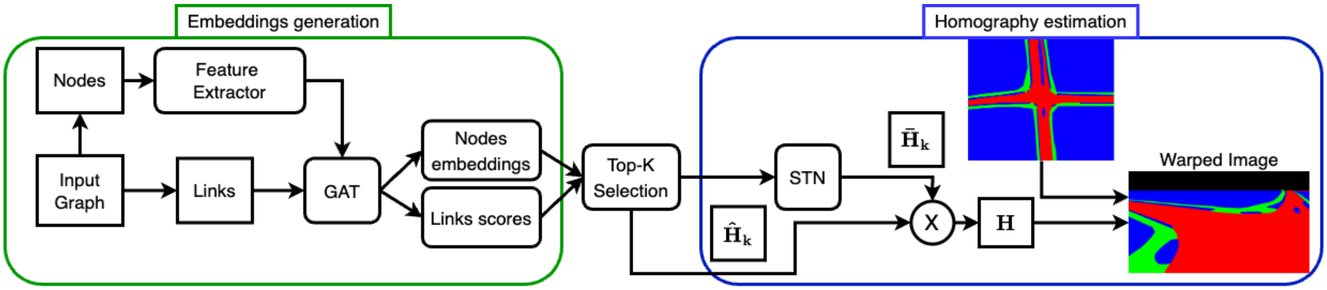


Figure 4. Overview of the proposed framework. In the first part, the features obtained from the nodes (images) in the sampled sub-graph are fed to the GAT network with the adjecency matrix of the links. The GAT produces embeddings for each node and predicts a score for each link. The top-k scoring links for each training (or testing) nodes are then processed by the STN to estimate the homography $\bar{\mathbf{H}}$.

2.2. Graph Neural Networks

Since their introduction in [11, 29], Graph Neural Networks (GNNs) have demonstrated promising results in various tasks, involving graph-structured data. For instance, in [17], Graph Convolutional Networks (GCNs) are employed for graph-level and node-level classification. Kipf and Welling [16] proposed an auto-encoder version of GNN for unsupervised link prediction. Over the years, GNNs have been employed for tasks such as nearest-neighbour estimation [23] and sentiment analysis [20]. Recent years have shown a growing interest in exploring the application of GNNs in image and video-related tasks. The graph embeddings learned by GNNs have facilitated various downstream tasks, including pose estimation [24, 26] and object detection [19]. In [34], the key points of an image are treated as nodes in a graph, with links connecting them, and a GNN model is trained to match these key points. Similarly, in [28], graphs representing key points are used for image matching. The successes of GNNs in image-related tasks have inspired us to apply this approach to the specific problem of automated camera calibration in traffic scenes. By constructing a graph of synthetic templates based on the topological structure of intersections, we train a GNN model to perform homography estimation. The use of GNNs sets our work apart from previous approaches and allows accurate automated camera calibration in traffic monitoring scenarios.

3. Method

The proposed framework leverages a GNN to match the input image with a dictionary of templates, assigning a score to all possible links. A Spatial Transformer Network (STN) estimates the homography transformation between the highest scoring template and the input image from the embeddings generated by the GNN. The images and the templates are generated, starting from the manually segmented bird's-eye-view (BEV) of the intersection. Whereas we train the framework only on synthetic images, its performance is evaluated on a test set of synthetic images and on five real-world cameras placed in Intersection 3 (see Figure 1c).

3.1. Data Generation

Following the approach described in [30] and [8], we generate virtual cameras by randomly sampling intrinsic and extrinsic parameters from a grid. The semantically segmented BEV of the scene, shown in Figure 1, is warped with the homography matrices of these virtual cameras, resulting in a set of approximately 20,000 images per scene. To create the training and testing splits, images are randomly sampled from the generated set, with an equal distribution among the two splits. Additionally, a dictionary split is sampled containing images to be leveraged as anchors for the final homography estimation task, following [30]. Each split forms a graph, where each image corresponds to a node connected to the top-k most similar images in the dictionary (empirically, k=20). The similarity score between images can be calculated using similarity metrics, such as Topological Loss [8]. During both training and testing, nodes from the respective graph are sampled to construct mini-batches. As illustrated in Figure 3, for each node in a mini-batch, we also sample a number of its neighbours at a 2-hop distance, following the approach presented in [12]. However, we discard all nodes in the second hop that are not part of the dictionary to prevent any cross-contamination between splits. As shown in Figure 2, the images in the second hop can be visually different from the training image, or representing a different part of the intersection.

3.2. Matching Process

In our approach, a light-weight feature extractor derives a feature vector from each image. In addition to the feature vectors, a matrix is constructed containing all possi-

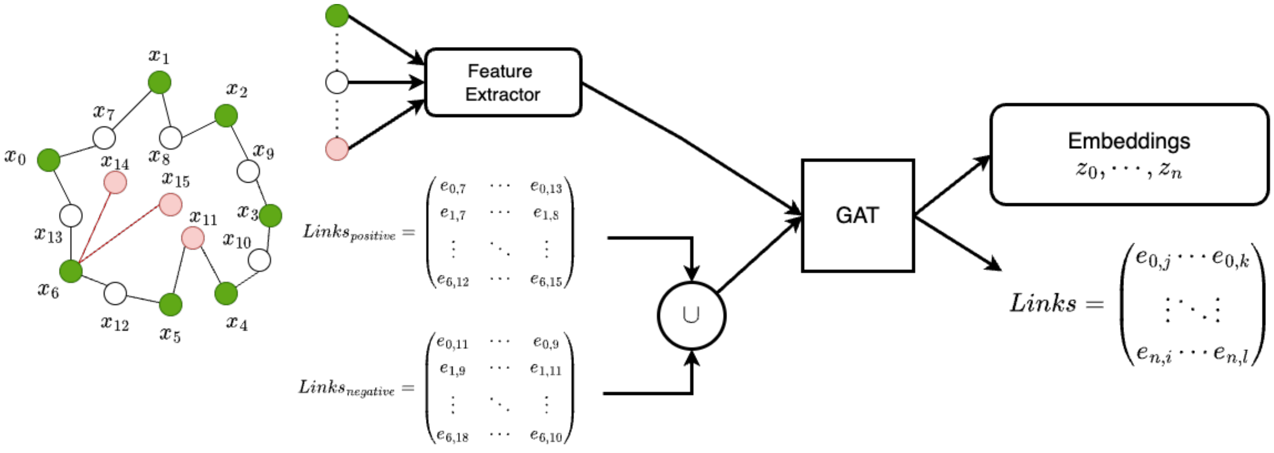

x_0
x_1
x_2
x_3
x_4
x_5
x_6
x_7
x_8
x_9
x_{10}
x_{11}
x_{12}
x_{13}
x_{14}
x_{15}
Feature Extractor
$Links_{positive} = \begin{pmatrix} e_{0,7} & \cdots & e_{0,13} \\ e_{1,7} & \cdots & e_{1,8} \\ \vdots & \ddots & \vdots \\ e_{6,12} & \cdots & e_{6,15} \end{pmatrix}$
$Links_{negative} = \begin{pmatrix} e_{0,11} & \cdots & e_{0,9} \\ e_{1,9} & \cdots & e_{1,11} \\ \vdots & \ddots & \vdots \\ e_{6,18} & \cdots & e_{6,10} \end{pmatrix}$
$\cup$
GAT
Embeddings
$z_0, \cdots, z_n$
$Links = \begin{pmatrix} e_{0,j} \cdots e_{0,k} \\ \vdots \ddots \vdots \\ e_{n,i} \cdots e_{n,l} \end{pmatrix}$

Figure 5. Overview of the embedding generation process. The features of all nodes in the mini-batch are generated by the feature extractor. The positive links are the links between each training node and its dictionary nodes in the mini-batch, while the negative links are all the links between training nodes and the other dictionary nodes. Merging positive and negative links creates the matrix of all possible links in the mini-batch. The GAT produces embeddings for each node and probability scores for the links, the latter being used to train both models via Equation (1).

ble links between nodes in the mini-batch. Both the feature vectors and the matrix of links serve as inputs to a Graph Attention Network (GAT) to perform the matching task. The GAT model has been proposed in [5] and represents an improvement over the preceding design reported in [32]. GATs follow the same principles as GCNs, but with a key difference in the way they handle neighbourhood features. While a standard GCN normalizes node features across neighbours during the convolutional operation, a GAT employs a masked attention mechanism to compute attention coefficients between nodes within the same neighbourhood, thereby preserving important structural information.

The feature extractor and the GAT are trained jointly, allowing the former to learn meaningful representations of the original images for the latter. The training loss function used for the GAT is the conventional Binary Cross-Entropy, defined as:

$$L_{GAT} = -\sum_{n=0}^{N-1} \big(y_n \log(p_n) + (1 - y_n) \log(1 - p_n) \big). \tag{1}$$

Here, N represents the number of possible links, y_n is the binary label for link n, and p_n is the score predicted by the GAT for that link. This loss function ensures effective training of both the feature extractor and the GAT for accurate and robust link prediction.

An overview of the link-prediction task is shown in Figure 5. The final outputs of the GAT are the embeddings for every node in the batch and a score for each possible link between these nodes.

3.3. Homography Estimation

The GAT predicts a score for every possible link between training and dictionary nodes in the batch. The top-k scoring links for each training (or testing) node are utilized to select the neighbours containing the most useful information for the Spatial Transformer Network (STN). The feature vectors are concatenated into a two-dimensional feature matrix and passed as input to the homography estimation component. The homography of the highest-scoring dictionary node, $\hat{\mathbf{H}}_k$, is the anchor for the corresponding training (or testing) node. In fact, the STN estimates a homography that transforms the best match found by the GNN to the target image. In other words, the homography $\bar{\mathbf{H}}$, estimated by the STN, is multiplied with its anchor $\hat{\mathbf{H}}_\mathbf{k}$, producing the final homography $\mathbf{H} = \hat{\mathbf{H}}_\mathbf{k}\bar{\mathbf{H}}$.

The STN described in [8] is adapted to accept the embeddings produced by the GAT. Following that work, we adopt the same training strategy: the semantic BEV is warped by the estimated homography and the resulting warped image is compared to the synthetic ground truth via a Topological Loss implementation of the MSE. This is described by Equation (2), where the running indexes $k, l \in \{-1, 0, +1\}$ and $i, j \in \{0, ..., \sqrt{N}\}$, while Y and $\hat{Y}$ are the ground-truth image and the warped image, respectively.

$$\mathcal{L}_{\text{patch}}(\hat{Y}, Y) = \text{MSE}(\hat{Y}_{i,j}, Y_{i,j}) + \\ \alpha \sum_k \sum_l \max\left(0, \text{MSE}(\hat{Y}_{i+k,j+l}, Y_{i+k,j+l}) - \beta\right),$$

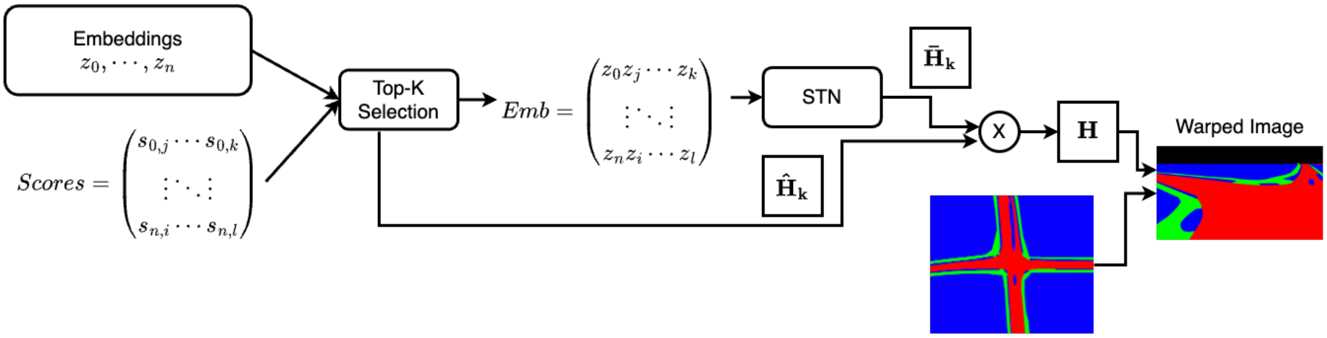

Embeddings
$z_0, \cdots, z_n$
$Scores = \begin{pmatrix} s_{0,j} \cdots s_{0,k} \\ \vdots \ddots \vdots \\ s_{n,i} \cdots s_{n,l} \end{pmatrix}$
Top-K Selection
$Emb = \begin{pmatrix} z_0 z_j \cdots z_k \\ \vdots \ddots \vdots \\ z_n z_i \cdots z_l \end{pmatrix}$
STN
$\bar{\mathbf{H}}_\mathbf{k}$
$\hat{\mathbf{H}}_\mathbf{k}$
X
$\mathbf{H}$
Warped Image

Figure 6. Overview of the homography estimation task. For each training node, the probability scores predicted by the GAT are needed to select the top-k best nodes, whose embeddings are then used by the STN to predict $\bar{\mathbf{H}}$. The homography of the node connected to a training node via the highest scoring link is the anchor for the final estimated homography that warps the BEV. The resulting warped images are employed to train end-to-end all models via Equation 2.

$$\text{and } \mathcal{L}_{\text{Top-MSE}} = \frac{1}{N}\sum_i \sum_j \mathcal{L}_{\text{patch}}(\hat{Y}_{i,j}, Y_{i,j}). \qquad (2)$$

4. Results

4.1. Experimental setup

The GAT and its feature extractor are trained jointly until convergence, using the loss for link prediction reported in Equation (1). In this way, the GAT learns the features of the nodes and their neighbours which can be better exploited by the STN. After this warmup period of 30 epochs, we start training the full pipeline of feature extractor, GAT and STN, using only the loss function in Equation (2). Therefore, the GAT may learn to choose nodes outside the neighbourhood of the training node as more relevant for the training objective. The IoU scores are measured between the ground-truth images and the BEV warped with the estimated homographies to evaluate the model. If the IoU score does not improve for 10 epochs, the learning rates of all models are halved, and if it does not improve for a further 15 epochs the training is stopped.

We evaluate the model on five different intersection scenes and the World Cup 2014 dataset [14]. The results shown in Table 1 are averaged over five training cycles for each dataset and different train/test/dictionary splits are sampled at every cycle. Table 3 validates the results by evaluating the model performance on images obtained from five cameras located in Intersection 3. The experimental results are compared to the baselines, reported in [8] and [30]. Furthermore, the performances are compared with the original GAT model proposed by [32] and with a standard GCN presented in [17].

4.2. Ablation study

Table 1 shows that the proposed method outperforms the state-of-the-art models with any of the three GNN networks, strongly supporting the idea of a graph-based approach for this task. The main difference between the proposed work and the competitors is in the way the structural information of the intersection is leveraged. In fact, both previous works employed a Siamese model to perform the matching task, for which the best template from the dictionary was selected. This approach has two downsides, as listed below.

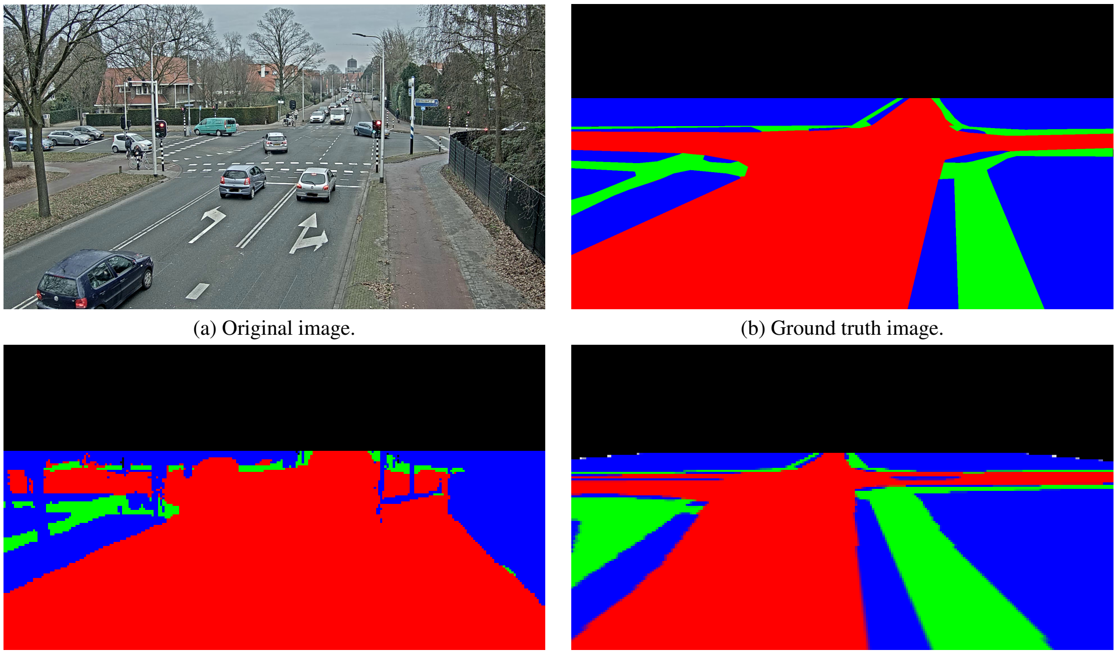
- Selecting a single match is valid but not optimal: the amount and quality of topological information that becomes available from a single match is inferior to the information that can be extracted from multiple matches and the relationships between them.
- The Siamese network as a matching component is computationally expensive: each input image is compared to every image in the dictionary, which places a heavy burden on computational resources and hinders the scaling capabilities of the model.

Framing the matching process as a link-prediction task in a graph solves both problems, allowing us to retrieve multiple matches at a lower computational cost. Furthermore, the GNN learns features for the nodes based on the related visual features and the relationships between them, which gives the STN a much richer input. Training the models end-to-end has been proven not effective in [8], due to the design of the matching process. In this work, training all models end-to-end provides much more flexibility to the GNN, allowing it to discard subpar matches directly connected to the training node in favor of other nodes in the neighbourhood that can be more relevant for the STN.

It should be noticed that the difference between the three

Table 1. Obtained IoU scores by different implementations of the proposed model, using three different GNNs and the baseline models, over the five synthetic intersection datasets and the World Cup 2014 dataset. All values (mean + std) are expressed in %.

Model	**Intersections**					**World Cup**
	1	**2**	**3**	**4**	**5**	**2014**
Sha *et al.* [30]	85.96	78.42	78.99	83.16	71.95	81.64
D'Amicantonio *et al.* [8]	87.91	87.00	86.66	84.51	73.75	84.21
Proposed 1 (GCN [17])	92.31±0.37	93.25±0.16	93.58±0.11	88.28±0.36	87.99±0.21	89.91±0.44
Proposed 2 (GAT [32])	95.61±0.99	**96.57**±0.68	93.98±1.45	88.89±1.2	96.27±0.24	96.89±0.7
Proposed 3 (GATv2 [5])	**95.96**±0.84	96.56±0.53	**95.38**±1.32	**89.59**±0.14	**96.86**±0.57	**97.31**±0.39



(a) Original image.

(b) Ground truth image.

(c) Segmented image.

(d) Warped image generated by the proposed model.

Figure 7. Visual results of segmentation and warping steps for a real-world image. The segmented image obtained from the SegFormer is less precise than the synthetic images on which the model was trained, which partially explains the drop in accuracy between testing images and real-world images.

types of GNN we experimented with, do not show a significant difference in performance. The GCN proves to be the least performing of the three, but it still outperforms the current state-of-the-art by 10% on average. Moreover, the difference in performance between GAT and GATv2 is minor.

The employed feature extractor to prepare the images for the GNN models is a lightweight model, comprising four convolutional layers only. To prove that such a simple model is not a bottleneck for our framework, Table 2 compares the performances with a larger and more complex feature extractor such as ResNet18. The larger feature extractor does not have a positive impact on the overall performance of the model, as it is outperformed by the much simpler model on all datasets. We conjecture that because of the much simpler nature of the semantic images, a more complex model is harder to train due to the higher amount of parameters.

4.3. Evaluation on real camera views

To validate the results obtained by the proposed model on synthetic images, the models are executed on images obtained from five cameras located in Intersection 3. The images are segmented by SegFormer [33], which is trained on Cityscapes [6]. Given that Cityscapes has 30 classes while our application only needs 4 (terrain, road, bicycle path and background), we discard or merge the additional classes in one of the four classes. While the merging strategy heavily depends on the individual scene, it is fair to assume that e.g. the car class can be merged with the road class and the bi-

Table 2. Comparison between the proposed model with different feature extractors. All values (mean + std) are expressed in %.

Model	**Intersections**					**World Cup**
	1	**2**	**3**	**4**	**5**	**2014**
Proposed (GATv2 + ResNet18)	95.13±0.13	95.75±0.37	94.45±0.17	87.65±0.61	95.97±0.07	95.41±0.44
Proposed (GATv2 + 4-Conv)	**95.96**±0.84	**96.56**±0.53	**95.38**±1.32	**89.59**±0.14	**96.86**±0.57	**97.31**±0.39

Table 3. Average IoU scores obtained by the proposed models and the baseline models on five real-world cameras located in Intersection 3. All values (mean + std) are expressed in %.

Model	**Cameras**
Sha *et al.* [30]	76.44±3.56
D'Amicantonio *et al.* [8]	81.12±2.54
Proposed 1 (GCN [17])	85.73±0.34
Proposed 2 (GAT [32])	86.48±0.85
Proposed 3 (GATv2 [5])	**87.89**±0.93

cycle class with the bicycle-path class.

In Table 3, the proposed model is compared with the previous baseline model. With each of the three GNNs, the proposed model outperforms the baseline models by up to 8%. It is worth noting that the segmented images are not as precise as the synthetic images, which may cause the drop in accuracy w.r.t. the testing dataset. Furthermore, we do not account for any distortion while generating the synthetic images, which may be a factor for real-world cameras, especially fish-eye cameras. As shown in Figures 7c and 7b, the segmentation obtained from the SegFormer model is not very precise in some areas, such as the bicycle path at the right or the poles and trees at the left. We conjecture that fine-tuning a segmentation model to discard obstructing objects may lead to even better results with the same model. We leave this for future work, as it is not in the scope of this paper.

5. Conclusions

In this work, we have proposed an automated homography estimation framework based on Graph Neural Networks. It is shown that framing multiple synthetic views of an intersection as a graph and training a GNN to learn the relationships between them, leads to the generation of rich embeddings which can be useful to estimate the homography of a given image. The proposed framework has been tested on five different intersections, five real-world cameras and a real-world dataset with three different types of GNNs. In all settings, the framework outperforms the current state-of-the-art by a significant margin, proving to be also very effective in real-world applications. In future work, it can be interesting to focus on improving the scoring mechanism for the matches retrieved by the GNN and refining the segmentation of camera images.

References

[1] Dániel Baráth, Jana Noskova, Maksym Ivashechkin, and Jiri Matas. Magsac++, a fast, reliable and accurate robust estimator. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 1301–1309. Computer Vision Foundation / IEEE, 2020. 2

[2] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In Aleš Leonardis, Horst Bischof, and Axel Pinz, editors, *Computer Vision – ECCV 2006*, pages 404–417, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. 2

[3] Romil Bhardwaj, Gopi Krishna Tummala, Ganesan Ramalingam, Ramachandran Ramjee, and Prasun Sinha. Autocalib: Automatic traffic camera calibration at scale. *ACM Trans. Sen. Netw.*, 14(3–4), nov 2018. 2

[4] Jia-Wang Bian, Wen-Yan Lin, Yun Liu, Le Zhang, Sai-Kit Yeung, Ming-Ming Cheng, and Ian Reid. GMS: Grid-based motion statistics for fast, ultra-robust feature correspondence. *International Journal of Computer Vision (IJCV)*, 2020. 2

[5] Shaked Brody, Uri Alon, and Eran Yahav. How attentive are graph attention networks? *arXiv preprint arXiv:2105.14491*, 2021. 4, 6, 7

[6] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 6

[7] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Deep image homography estimation. *CoRR*, 06 2016. 2

[8] Giacomo D'Amicantonio, Egor Bondarau, and Peter H.N. De With. Homography estimation for camera calibration in complex topological scenes. In *2023 IEEE Intelligent Vehicles Symposium (IV)*, pages 1–8, 2023. 1, 2, 3, 4, 5, 6, 7

[9] Farzan Erlik Nowruzi, Robert Laganiere, and Nathalie Japkowicz. Homography estimation from image pairs with hierarchical convolutional networks. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 913–920, 2017. 2

[10] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to

image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, jun 1981. 2

[11] Marco Gori, Gabriele Monfardini, and Franco Scarselli. A new model for learning in graph domains. *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, 2:729–734 vol. 2, 2005. 2, 3

[12] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017. 3

[13] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, New York, NY, USA, 2 edition, 2003. 2

[14] Namdar Homayounfar, Sanja Fidler, and Raquel Urtasun. Sports field localization via deep structured models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 5

[15] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. *CoRR*, abs/1506.02025, 2015. 2

[16] Thomas N. Kipf and Max Welling. Variational graph autoencoders, 2016. 3

[17] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*, 2017. 3, 5, 6, 7

[18] Hoang Le, Feng Liu, Shu Zhang, and Aseem Agarwala. Deep homography estimation for dynamic scenes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7652–7661, 2020. 2

[19] Kejie Li, Daniel DeTone, Yu Fan (Steven) Chen, Minh Vo, Ian Reid, Hamid Rezatofighi, Chris Sweeney, Julian Straub, and Richard Newcombe. Odam: Object detection, association, and mapping using posed rgb video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5998–6008, October 2021. 3

[20] Wenxiong Liao, Bi Zeng, Jianqi Liu, Pengfei Wei, Xiaochun Cheng, and Weiwen Zhang. Multi-level graph neural network for text sentiment analysis. *Computers & Electrical Engineering*, 92:107096, 2021. 3

[21] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, Nov. 2004. 2

[22] Jiayi Ma, Ji Zhao, Hanqi Guo, Junjun Jiang, Huabing Zhou, and Yuan Gao. Locality preserving matching. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 4492–4498, 2017. 2

[23] Federico Monti, Michael Bronstein, and Xavier Bresson. Geometric matrix completion with recurrent multi-graph neural networks. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. 3

[24] Negar Nejatishahidin, Will Hutchcroft, Manjunath Narayana, Ivaylo Boyadzhiev, Yuguang Li, Naji Khosravan, Jana Košecká, and Sing Bing Kang. Graph-covis: Gnn-based multi-view panorama global pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 6458–6467, June 2023. 3

[25] Ty Nguyen, Steven W. Chen, Shreyas S. Shivakumar, Camillo J. Taylor, and Vijay Kumar. Unsupervised deep homography: A fast and robust homography estimation model. *CoRR*, abs/1709.03966, 2017. 2

[26] Barbara Roessle and Matthias Nießner. End2end multi-view feature matching using differentiable pose optimization, 2022. 3

[27] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *2011 International Conference on Computer Vision*, pages 2564–2571, 2011. 2

[28] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 3

[29] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2009. 2, 3

[30] Long Sha, Jennifer Hobbs, Panna Felsen, Xinyu Wei, Patrick Lucey, and Sujoy Ganguly. End-to-end camera calibration for broadcast videos. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13624–13633, 2020. 2, 3, 5, 6, 7

[31] Yurun Tian, Xin Yu, Bin Fan, Fuchao Wu, Huub Heijnen, and Vassileios Balntas. Sosnet: Second order similarity regularization for local descriptor learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11016–11025, 2019. 2

[32] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017. 4, 5, 6, 7

[33] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, José M. Álvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *CoRR*, abs/2105.15203, 2021. 6

[34] Nancy Xu, Giannis Nikolentzos, Michalis Vazirgiannis, and Henrik Boström. Image keypoint matching using graph neural networks. In *Complex Networks & Their Applications X*, pages 441–451, Cham, 2022. Springer International Publishing. 3

[35] Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal Fua. LIFT: learned invariant feature transform. *CoRR*, abs/1603.09114, 2016. 2

[36] Jirong Zhang, Chuan Wang, Shuaicheng Liu, Lanpeng Jia, Jue Wang, and Ji Zhou. Content-aware unsupervised deep homography estimation. *CoRR*, abs/1909.05983, 2019. 2

[37] Wentao Zhang, Huansheng Song, Lichen Liu, Congliang Li, Bochen Mu, and Qian Gao. Vehicle localisation and deep model for automatic calibration of monocular camera in expressway scenes. *IET Intelligent Transport Systems*, 16, 04 2022. 2