Bayesian Tracking of Video Graphs Using Joint Kalman Smoothing and Registration: Supplementary

Aditi Basu Bal[1], Ramy Mounir[2], Sathyanarayanan Aakur[3], Sudeep Sarkar[2], and Anuj Srivastava[1]

[1] Florida State University, Tallahassee, FL 32309, USA
[2] University of South Florida, Tampa, FL 33620, USA
[3] Oklahoma State University, Stillwater, OK 74078
ab18z@fsu.edu, {ramy, sarkar}@usf.edu, saakurn@okstate.edu, anuj@stat.fsu.edu

In this supplementary document we present additional details on our proposed methodology and experimental results for joint Kalman smoothing and registration of time series graphs.

1 MEVA Dataset

The *real-world* dataset we used to evaluate our framework, MEVA was briefly introduced in the manuscript. Here, we present further details on it and its pre-processing.

1.1 Dataset

The Multiview Extended Video with Activities (MEVA) [7] is a large-scale dataset for human activity recognition. It consists of over 9300 hours (untrimmed and continuous) of scripted scenarios and spontaneous background activities from indoor and outdoor viewpoints. The dataset offers video data from 38 RGB and thermal IR cameras, as well as UAV footage. For our experiments, we use a scene from an indoor bus station where the actors are continuously moving in and out of the camera field of view while performing different activities. The bus station scene is 5 minutes long, sampled at 30 FPS (9000 frames). The MEVA dataset contains 79 different videos for the bus stop location, out of which 51 have no actors. We have chosen the video with the highest number of moving actors (24); including actors entering and exiting the scene.

1.2 Preprocessing

Object Detection We preprocess each frame to construct a graph, where each node is a person detection and their node embeddings represent visual features. More formally, given a set of frames $\mathcal{X} = \{x_1, \ldots, x_n\}$, where $x_i \in \mathbb{R}^{H,W,C}$ and n is the number of frames in the video, we apply a pretrained object detector (Detection Transformer (DeTR) [5]) on each frame resulting in a set of detections

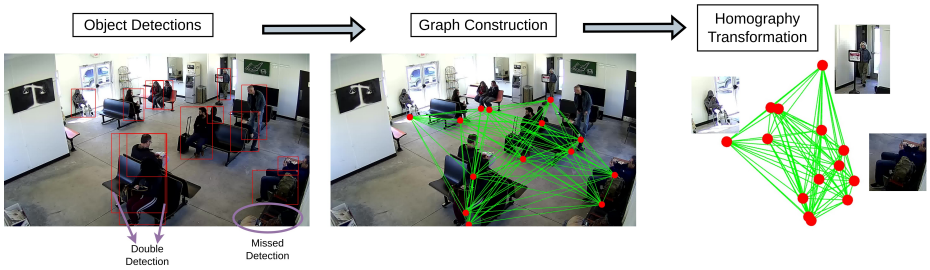

Fig. 1: Preprocessing pipeline. A pretrained object detector is used to detect humans in the scene and extract their visual features. Detections $\mathcal{O}$ are used to construct an undirected graph G. Homography transformation is applied to approximate the relative distances between detections, which are used to construct the affinity matrix.

$\mathcal{O} = \{o_1, \ldots, o_n\}$. The input image $x_i \in \mathbb{R}^{H,W,C}$ is transformed to a feature grid $x_i' \in \mathbb{R}^{h,w,D}$ by using a CNN backbone. Positional encoding is added to the features before flattening the 2D grid into a *set* of feature vectors. A transformer encoder processes the feature vectors and outputs a set of feature vectors of the same size as the input set. The transformer decoder is initialized with a fixed number of learned positional encodings (object queries) and attends to the output of the transformer encoder. A shared Feed Forward Network (FFN) is attached to each object query allowing the model to classify each detection and regress its location. Hungarian matching loss is applied between the two sets (queries and groundtruth). We extract the visual features vector f_i for every detection o_i from the output of the transformer decoder (before the FFN). Finally, each detection is represented by $o_i = (a_i, s_i, f_i)$, where a_i, s_i, f_i denotes the object class, spatial location, and visual features vector, respectively.

Graph Construction We use the detection results to model interactions and activity in the scene as an undirected graph $G = (V, E)$, where $V = \{v_i, \ldots, v_n\}$ represents the nodes and $E \in \mathbb{R}^{|V|\times|V|}$ represents the edges. Each node v_i is a unique detection $o_i \in \mathcal{O}$, where the node embeddings are the features vector f_i extracted from the last layer of the transformer decoder. We filter the nodes by applying Non-max suppression on the detections, followed by dropping low confidence detections. Only person detection are used in our experiments.

Homography Transformation The edges $e_{ij} \in E$ are constructed by calculating the relative distances between detections. We approximate the relative distances between nodes by calculating the homography transformation matrix $\mathcal{H} \in \mathbb{R}^{3\times3}$ to transform the input camera view to a top view. Geometric clues, such as floor tiles, are used for approximating the homography transformation. Each detection location s_i is pre-multiplied by the homography matrix $\mathcal{H}$ to

calculate the location from the top view. The distances between the top view locations are normalized and used to construct the affinity matrix for graph G.

1.3 Related Papers

Trajectory Modelling and Prediction
* TRiPOD: Human Trajectory and Pose Dynamics Forecasting in the Wild [3]
* STGAT: Modeling Spatial-Temporal Interactions for Human Trajectory Prediction [9]
* The Trajectron: Probabilistic Multi-Agent Trajectory Modeling With Dynamic Spatiotemporal Graphs [10]
* Social-BiGAT: Multimodal Trajectory Forecasting using Bicycle-GAN and Graph Attention Networks [12]
* Trajectron++: Dynamically-Feasible Trajectory Forecasting With Heterogeneous Data [18]
* Social-STGCNN: A Social Spatio-Temporal Graph Convolutional Neural Network for Human Trajectory Prediction [15]
* GraphTCN: Spatio-Temporal Interaction Modeling for Human Trajectory Prediction [21]

Video Object Segmentation
* Video Object Segmentation with Episodic Graph Memory Networks [14]
* Zero-Shot Video Object Segmentation via Attentive Graph Neural Networks [22]

Tracking
* Learning a Neural Solver for Multiple Object Tracking [4]
* Graph Networks for Multiple Object Tracking [13]
* Joint object detection and multi-object tracking with graph neural networks [23]
* GNN3DMOT: Graph Neural Network for 3D Multi-Object Tracking with Multi-Feature Learning [24]

Pattern Theory
* Generating Open World Descriptions of Video using Commonsense Knowledge in a Pattern Theory Framework [2]
* Going Deeper with Semantics: Exploiting Semantic Contextualization for Interpretation of Human Activity in Videos [1]

2 Registration of Ground Truth to Estimated Graphs (MEVA Dataset)

For the real-world evaluation, the estimation errors reported in section 5.4 in the manuscript are computed against a ground truth series $\{\mathbf{x}_t\}$ that was generated manually. In this series the nodes (subjects) are ordered arbitrarily. Hence, before computing the $\mathbb{L}^2$ norm of the difference between the estimated graph $\hat{\mathbf{x}}_t$ and the ground truth $\mathbf{x}_t$ we include a *graph matching* step. Every $\mathbf{x}_t$ is registered to

its corresponding $\hat{\mathbf{x}}_t$ using the same graph matching algorithm [8] we used to register the $\{\mathbf{y}_t\}$ series. Our estimation error is dependent on the accuracy of this graph matching step. If nodes of $\mathbf{x}_t$ and $\hat{\mathbf{x}}_t$ are registered incorrectly, we end up reporting higher estimation error than there actually is.

3 Definition of Missed and False Detections

The graph matching algorithm we employ in our framework allows the registration of real or actual nodes in one graph to *null nodes* in the other. Null nodes are dummy nodes added to graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ before graph matching to make them of the same size ($|V_1| + |V_2|$) and to facilitate optimal registration over a larger search space. Their attributes are generally set to zeros denoting null values. Assuming we obtained fully accurate registration between $\mathbf{x}_t$ and $\hat{\mathbf{x}}_t$, when a real node present in $\mathbf{x}_t$ is matched to a null node in $\hat{\mathbf{x}}_t$, the node is a *missed detection*. This means that this subject is present in the scene but is undetected. Conversely, when a real node present in $\hat{\mathbf{x}}_t$ is matched to a null node in $\mathbf{x}_t$, the node is a *false/noisy detection*. This may mean that there is no subject in the scene and yet something is detected in the observation. Another possibility is that a subject is present in the scene but is detected multiple times. As a result, a node in $\hat{\mathbf{x}}_t$ denoting one of these multiple detections will be matched to the real node denoting this subject in $\mathbf{x}_t$ while the rest of the detections are matched to null nodes in $\mathbf{x}_t$. The latter detections are the false/noisy detections. An example of this scenario is in Fig. 10 (bottom left) in the manuscript. Node 8 and node 14 in $\mathbf{y}_t$ (coral) are double detections of the same subject and their estimates in $\hat{\mathbf{x}}_t$ are shown in green. One of them, node 14, is matched to the real node corresponding to this subject in the ground truth $\mathbf{x}_t$ (blue). Node 8, therefore, counts as a false detection.

4 Missed and False Detections for Synthetic Datasets 1 and 2

The rates of missed and false detections in Synthetic Datasets 1 and 2 were not specifically reported in the manuscript. The reasons are addressed here:

4.1 Synthetic Dataset 1

In this datatset, we simulate the *ideal* observation setting where the observations are not degraded by clutter or missed detections. Moreover, this dataset assumes that no subject leaves the scene and no new subject enters it. Consequently, the time series graphs demonstrating this ideal scenario have the same number of nodes from the initiation of time till the end and they are perfectly registered across $\mathbf{x}_t$, $\mathbf{y}_t$ and $\hat{\mathbf{x}}_t$. As a result, there are no missed or false detections in this case.

4.2 Synthetic Dataset 2

In this dataset, we first generate a series of fully connected graphs $\{\mathbf{x}_t\}$. To simulate missed detections or the entry/exit of a subject in the scene, 2-3 nodes in $\mathbf{x}_t$ are removed at randomly chosen time points t spaced out over $t = 0, 1, 2, .., T$, generating the observed series $\{\mathbf{y}_t\}$. We then register $\{\mathbf{y}_t\}$ sequentially as described in Algorithm 1 in the manuscript. The registration obtained at this step turned out to be very successful, i.e., the null nodes in $\mathbf{y}_t$ corresponding to the nodes (subjects) missed in the observation were matched with the right nodes (subjects) observed in $\mathbf{y}_{t-1}$. These null nodes in $\mathbf{y}_t$ were then assigned real node status in $\hat{\mathbf{x}}_t$, demonstrating an important contribution of our framework where a subject that failed to be detected at time t is retrieved through accurate graph matching with the previous time point $t-1$ where this subject was, in fact, detected. In our dataset with controlled noise and detection failures, the registration turned out to be 100% accurate for all time points, all missed detections were retrieved in $\hat{\mathbf{x}}_t$ in the above-mentioned manner. Therefore, once again, there are no missed or false detections in this evaluation.

5 Estimation Errors for Alternative Models (MEVA Dataset)

Due to limited availability of space in the manuscript, we were unable to include a comparative report on the mean and standard deviation of estimation errors in the real-world evaluation using various models including our Joint Kalman Smoother and Registration and other alternatives. We present these in Table 1. The three data columns refer to three short segments of the full 5-minute video sequence under consideration. Clip 1 refers to time points $t = 0 - 500$, Clip 2 to $t = 0 - 1000$ and Clip 3 to $t = 782 - 1200$. Each of these clips cover specific events of interest such as a subject entering the bus station and walking across the room or taking a seat. These errors are measured for the last 100 time points in each set, while the rest are used for model training (in the alternative approaches). Our Kalman-Smoother with registration exhibits lower errors in estimating the location and tracking of subjects than the alternative models. It should also be noted that our method functions online and does not require training. Furthermore, our proposed method can retrieve and track subjects that were undetected at certain time points as well as identify noisy/multiple detections thereby overcoming several limitations in computer-vision based object detections in video sequences.

6 Combined Node Attribute for Real-world Evaluation

For evaluation of our proposed method on *real-world* data, we briefly mention the utilization of a combination of several node attributes to aid the registration of the observed series $\{\mathbf{y}_t\}$. We combine "(a) top-view node position coordinates obtained from homography transformation of the input camera view, (b) eight

Approach	Clip 1		Clip 2		Clip 3	
	Prediction Errors		Prediction Errors		Prediction Errors	
	Nodes	Edges	Nodes	Edges	Nodes	Edges
Multi-Epoch Training						
1-Step FFN	1.045 ±0.133	1.518 ±0.210	0.961 ±0.119	1.286 ±0.167	1.022 ±0.136	1.373 ±0.490
RNN	1.005 ±0.046	1.015 ±0.008	1.014 ±0.040	1.011 ±0.007	1.087 ±0.023	1.007 ±0.005
GRU	1.002 ±0.045	1.022 ±0.011	1.008 ±0.045	1.014 ±0.008	1.084 ±0.021	1.003 ±0.002
Seq2Seq-RNN	1.050 ±0.020	1.004 ±0.004	1.052 ±0.020	1.007 ±0.008	1.100 ±0.000	1.004 ±0.001
Seq2Seq-GRU	0.973 ±0.043	1.018 ±0.012	0.989 ±0.046	1.012 ±0.010	1.054 ±0.042	1.009 ±0.007
Transformer	1.096 ±0.024	1.095 ±0.021	0.970 ±0.014	1.166 ±0.026	1.049 ±0.039	1.291 ±0.018
Online Training						
RNN	1.018 ±0.027	1.049 ±0.015	1.002 ±0.039	1.046 ±0.005	1.027 ±0.013	1.037 ±0.011
GRU	1.011 ±0.036	1.039 ±0.009	0.998 ±0.033	1.043 ±0.008	1.024 ±0.019	1.034 ±0.015
Kalman Filter	**0.324 ±0.064**	**1.305 ±0.122**	**0.519 ±0.132**	**1.595 ±0.274**	**0.244 ±0.063**	**1.350 ±0.324**
Kalman Smoother	**0.321 ±0.085**	**1.297 ±0.124**	**0.529 ±0.115**	**1.631 ±0.295**	**0.253 ±0.078**	**1.360 ±0.303**
Static Prediction						
Median Filter	0.555 ±0.197	1.322 ±0.160	0.528 ±0.253	1.375 ±0.299	0.554 ±0.201	1.292 ±0.328

Table 1: Quantitative evaluation on Video Clips 1, 2 and 3 from MEVA dataset. Table reports mean and std. deviation of the $\mathbb{L}^2$ errors, for edge and node estimates.

principal components of the 256 length node embeddings explaining 85.3% of the variation, and (c) four coordinates of the bounding box of the detected subject, resulting in a $m = 14$-length attribute vector for each node". Further details on this node attribute are as follows:

- The choice of this combination was a result of several comparative experiments. We wanted to retain both visual and spatial information to best describe the nodes. The said combination of node attributes led to better sequential registration in comparison to any other individual or combination of these attributes.
- The values of node embeddings, the node position coordinates and the bounding box coordinates belong to different ranges by orders of magnitude. Therefore we normalize the entire $m = 14$-length attribute vector for each node in every graph across $\{\mathbf{y}_t\}$.

7 Videos of Time Series Graphs

We generated videos using the corresponding time series graphs for the datasets used in this paper showing 5 frames per second. Due to space constraint we include only two of the videos in this supplementary, Synthetic2.avi (first 50 time points) and MEVAClip1.avi denoting Synthetic Dataset 2 and a part of Clip 1 of the MEVA bus station dataset respectively. In each of the videos, the blue, coral and green graphs denote the system $\mathbf{x}_t$, the observation $\mathbf{y}_t$ and the estimate $\hat{\mathbf{x}}_t$ respectively. We observed that our proposed framework offers very accurate estimates of the system graphs in the synthetic datasets, where the $\hat{\mathbf{x}}_t$ almost coincide with the $\mathbf{x}_t$ at each t. In the case of real-world data, our model estimates the ground truth with high precision in most scenes with less movement among the subjects while recovering objects that failed to be detected. However, the entry of a new subject into the scene, and its movement, often disrupts the registration temporarily among nodes across $\{\mathbf{y}_t\}$. Nevertheless, the model quickly recovers from this disturbance within a few time points and proceeds to show precise edge and node estimations for all subjects including the ones moving across the frame.

8 Comparison with GNNs and Past Papers

Our problem setup requires several tools – graph-based representations, time-series analysis, graph registration, handling missing/spurious nodes, and performing joint (nodes + edges) inferences. This framework allows us to handle variable graph structures over time, not only in terms of node and edge values but also in graph sizes. (If needed, a GNN can be added downstream to our processing for classification.) Our paper brings together all these tools and applies it to noisy detections from videos. While previous papers (refs in [25,20,16,17,19,6,11] and others) can handle some of these challenges, there is no paper that handles all these issues simultaneously. For instance, some GNNs for node registration

don't allow for unmatched nodes (using null nodes as we do) to handle variable graph sizes. Most GNN-based time-series papers assume that graphs are pre-registered in time. Furthermore, our formulation is on the manifold of all graphs, i.e., each point in this manifold is a graph, while in a typical geometric GNN one considers individual graphs as representing manifolds and a point there is a node in the graph.

9 Practicality of the Approach and Time Complexity

The quotient-space formulation for graphs is theoretically sound and computationally efficient. To give an idea, the full detection pre-processing cost for MEVA dataset is about 0.307 sec/frame. After detection, the total cost for tracking around 15-node graphs over 1000 frames ($\approx$ 33.33 sec video length) is 69 sec, i.e., 0.069 sec/frame. The graph registration at every frame is the main cost ($\approx$ 28% of total time) in the current implementation. We can further improve speed by mixing the (slower Umayama-based) quadratic assignment with the (much faster Hungarian-based) linear assignment for graph registration. A similar strategy can be applied to fast moving (use Umeyama) versus slow moving (use Hungarian) objects in the scene.

References

1. Aakur, S., de Souza, F.D., Sarkar, S.: Going deeper with semantics: Video activity interpretation using semantic contextualization. In: 2019 IEEE Winter Conference on Applications of Computer Vision (WACV). pp. 190–199. IEEE (2019)
2. Aakur, S.N., de Souza, F.D.M., Sarkar, S.: Generating open world descriptions of video using common sense knowledge in a pattern theory framework. Quarterly of Applied Mathematics (2019)
3. Adeli, V., Ehsanpour, M., Reid, I.D., Niebles, J.C., Savarese, S., Adeli, E., Rezatofighi, H.: TRiPOD: Human trajectory and pose dynamics forecasting in the wild. CoRR **abs/2104.04029** (2021), `https://arxiv.org/abs/2104.04029`
4. Brasó, G., Leal-Taixé, L.: Learning a neural solver for multiple object tracking. CoRR **abs/1912.07515** (2019), `http://arxiv.org/abs/1912.07515`
5. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: European Conference on Computer Vision. pp. 213–229. Springer (2020)
6. Chen, F., Chen, Z., Biswas, S., Lei, S., Ramakrishnan, N., Lu, C.T.: Graph convolutional networks with kalman filtering for traffic prediction. In: In 28th International Conference on Advances in Geographic Information Systems (SIGSPATIAL '20) (2020)
7. Corona, K., Osterdahl, K., Collins, R., Hoogs, A.: MEVA: A large-scale multiview, multimodal video dataset for activity detection. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). pp. 1060–1068 (January 2021)
8. Guo, X., Srivastava, A., Sarkar, S.: A quotient space formulation for statistical analysis of graphical data. Journal of Mathematical Imaging and Vision **63**, 735–752 (March 2021)

9. Huang, Y., Bi, H., Li, Z., Mao, T., Wang, Z.: Stgat: Modeling spatial-temporal interactions for human trajectory prediction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6272–6281 (2019)
10. Ivanovic, B., Pavone, M.: The trajectron: Probabilistic multi-agent trajectory modeling with dynamic spatiotemporal graphs. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2375–2384 (2019)
11. Knyazev, A., Malyshev, A.: Accelerated graph-based nonlinear denoising filters. Procedia Computer Science **80**, 607–616 (2016)
12. Kosaraju, V., Sadeghian, A., Martín-Martín, R., Reid, I., Rezatofighi, S.H., Savarese, S.: Social-bigat: Multimodal trajectory forecasting using bicycle-gan and graph attention networks. arXiv preprint arXiv:1907.03395 (2019)
13. Li, J., Gao, X., Jiang, T.: Graph networks for multiple object tracking. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) (March 2020)
14. Lu, X., Wang, W., Danelljan, M., Zhou, T., Shen, J., Gool, L.V.: Video object segmentation with episodic graph memory networks. CoRR **abs/2007.07020** (2020), `https://arxiv.org/abs/2007.07020`
15. Mohamed, A., Qian, K., Elhoseiny, M., Claudel, C.: Social-stgcnn: A social spatio-temporal graph convolutional neural network for human trajectory prediction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14424–14432 (2020)
16. Paaßen, B., Göpfert, C., Hammer, B.: Time series prediction for graphs in kernel and dissimilarity spaces. Neural Processing Letters **48**(2), 669–689 (2018)
17. Rudi, A., Ciliberto, C., Marconi, G., Rosasco, L.: Manifold structured prediction. Advances in Neural Information Processing Systems **31** (2018)
18. Salzmann, T., Ivanovic, B., Chakravarty, P., Pavone, M.: Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVIII 16. pp. 683–700. Springer (2020)
19. Shi, L.: Kalman filtering over graphs: Theory and applications. IEEE Transactions on Automatic Control **54**(9), 2230–2234 (2009)
20. Vázquez-Enríquez, M., Alba-Castro, J.L., Docío-Fernández, L., Rodríguez-Banga, E.: Isolated sign language recognition with multi-scale spatial-temporal graph convolutional networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3462–3471 (2021)
21. Wang, C., Cai, S., Tan, G.: Graphtcn: Spatio-temporal interaction modeling for human trajectory prediction. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 3450–3459 (2021)
22. Wang, W., Lu, X., Shen, J., Crandall, D.J., Shao, L.: Zero-shot video object segmentation via attentive graph neural networks. CoRR **abs/2001.06807** (2020), `https://arxiv.org/abs/2001.06807`
23. Wang, Y., Kitani, K., Weng, X.: Joint object detection and multi-object tracking with graph neural networks. In: 2021 IEEE International Conference on Robotics and Automation (ICRA). pp. 13708–13715. IEEE (2021)
24. Weng, X., Wang, Y., Man, Y., Kitani, K.M.: Gnn3dmot: Graph neural network for 3d multi-object tracking with 2d-3d multi-feature learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6499–6508 (2020)
25. Yan, S., Xiong, Y., Lin, D.: Spatial temporal graph convolutional networks for skeleton-based action recognition. In: Thirty-second AAAI Conference on Artificial Intelligence (2018)