Detecting Invisible People

Tarasha Khurana[1] Achal Dave[1] Deva Ramanan[1,2]
[1]Carnegie Mellon University [2]Argo AI
{tkhurana, achald, deva}@cs.cmu.edu

In this supplement, we provide further analysis of our method and implementation details of our experiments. Section 1 presents additional ablation studies of our method. Section 2 extends our detection evaluation to tracking to use the popular IDF1 and MOTA (multi-object tracking accuracy) metrics. Section 3 provides details regarding our human vision experiment, which analyzes people's ability to detect and localize highly occluded objects. Section 4 discusses the experimental setup for PANDA and MOT-20 datasets, and Section 5 reports the runtime and presents pseudocode of our final depth-aware tracking algorithm. Please refer to the supplementary video for a qualitative analysis of our approach.

1. Ablation Study

In this section, we analyze the impact of using different depth estimators (Section 1.1), segmentation masks in place of bounding boxes for estimating average depth (Section 1.2), more sophisticated forecasters (Section 1.3), the performance of our method on moving *vs*. stationary cameras (Section 1.4), and, finally, the importance of different hyperparameters (Section 1.5).

1.1. Monocular Depth Estimators

Our method relies on an off-the-shelf monocular depth estimator to enable occlusion reasoning in 3D. In our main paper, we used the MegaDepth [12] estimator throughout our experiments. Here, we evaluate whether recent advances in monocular depth estimation provide more reliable *relative* depth estimates of people as used by our method. Specifically, we replace the MegaDepth estimator with the MannequinChallenge [11] and MIDAS [9] depth estimators in our method. We evaluate on MOT-17 using the Faster-RCNN set of public detections, and set all hyperparameters in our pipeline to their default values and disable the depth-aware noise scaling. This simple variant of our pipeline allows us to evaluate the quality of depth estimates from each of the three methods. Table 1 shows that the per frame depth estimator from Mannequin Challenge [11] does worse than MegaDepth [12] by 1.2 Top-5 F1 for invisible people and MIDAS [9] similarly does worse by 1.0 point. By the standard Top-1 F1 metric, these estimators degrade accuracy by 1.2 and 0.2 points respectively. As this simple variant of our pipeline is aimed at evaluating the relative depth orderings output from the depth estimators, these results suggest that while these depth estimators have become more accurate and generalizable over the years, the relative depth orderings of objects has not significantly improved.

	Top-5 F1		Top-1 F1		IDF1	
Depth est.	Occl	All	Occl	All	Occl	All
MegaDepth [12]	35.4±0.2	69.8±0.0	26.7	68.4	9.5	53.3
Mannequin [11]	34.2±0.2	69.4±0.0	25.5	68.0	8.5	53.3
MIDAS [9]	34.4±0.1	69.5±0.0	26.5	68.2	9.1	53.8

Table 1. Comparison of monocular depth estimators used in our pipeline. More recent depth estimators do not seem to provide more reliable *relative* depth orderings, which are used by our method.

Since monocular depth estimators can take as input images of varying sizes, we evaluate the effect of using higher resolution images as input to the estimator. Using a higher resolution input can increase the size of smaller objects in the scene (e.g., people far away), potentially allowing depth estimators to output more precise depth estimates. We evaluate using higher resolutions as input with the MIDAS [9] estimator in Table 2. By default, we resize images to a resolution of 512×384 pixels ('1x', the resolution MIDAS is trained with) from their original resolution of 1920×1080. We evaluate MIDAS [9] at 2× and 3× this default resolution and find in that doing so improves the Top-5 F1 for invisible peopleby 3.1%. We note here that this is not the case with the other two depth estimators [12, 11] whose performance decreases or stagnates with higher resolutions (not shown).

1.2. Boxes vs Masks

Our method estimates a person's depth by taking the average of the depth estimates within the person's bounding box. However, these pixels may contain background regions, leading to incorrect depth estimates. To address this, we evaluate

Depth	Res.	Top-5 F1 Occl	Top-5 F1 All	Top-1 F1 Occl	Top-1 F1 All	IDF1 Occl	IDF1 All
MIDAS	1x	34.4±0.1	69.5±0.0	26.5	68.2	9.1	53.8
MIDAS	2x	35.5±0.2	70.0±0.0	27.0	68.5	9.8	53.9
MIDAS	3x	37.5±0.2	69.9±0.0	27.0	68.2	10.8	53.9

Table 2. We evaluate a recent depth estimator, MIDAS [9], at varying input resolutions. At higher resolutions (3x), the estimator improves Top-5 F1 by 3.1 points, suggesting higher resolutions can improve depth estimates, likely by providing more reliable relative depths for faraway pedestrians.

	Top-5 F1 Occl	Top-5 F1 All	Top-1 F1 Occl	Top-1 F1 All	IDF1 Occl	IDF1 All
Boxes	39.8 ±0.2	70.5 ±0.1	26.7	68.5	10.5	54.8
Masks	40.6±0.3	71.3±0.0	27.3	69.1	11.0	54.7

Table 3. Replacing boxes by masks for getting mean depth of a person only helps by a small amount suggesting that boxes can reasonably replace masks.

a variant which uses an off-the-shelf instance segmentation method to only compute the average depth within a predicted person mask. To do this, we pass the Faster R-CNN public detections from MOT-17 as proposals into the mask head of Mask R-CNN [5]. Occasionally, this instance segmentation method may fail to produce a reasonable mask for a person. We design a simple strategy for detecting a common failure case: if the output segmentation mask covers less than 25% of the bounding box (in cases where the people are too small or out-of-distribution), we discard the predicted mask and treat the full bounding box as the mask. We do not use masks for the forecasted boxes of occluded people, as these boxes cover unknown occluders. In Table 3, we find that masks modestly help our method, increasing Top-5 and Top-1 F1 by 0.6 and 0.8 points for occluded people. Interestingly, we also see an increase in overall F1 by the same amount.

1.3. Forecasting Approaches

As described in the main paper, we use a constant velocity forecaster in our probabilistic approach. In Sec 4.3, we showed that replacing our our simple linear forecaster with more sophisticated state-of-the-art forecasters that exploit social cues did not improve performance. Here, we provide implementation details for these experiments, and analyze different variants. The approaches discussed in the main paper, SGAN [4] and STGAT [7] are supplied the top-down views from our algorithm. Both SGAN and STGAT forecast 20 samples and then choose the closest trajectory to the groundtruth from these 20. This advantage is not feasible for

		Top-5 F1 Occl	Top-5 F1 All	Top-1 F1 Occl	Top-1 F1 All	IDF1 Occl	IDF1 All
Single	SGAN-8	35.4±0.2	70.2±0.0	24.6	67.8	8.9	54.3
	SGAN-12	35.0±0.1	70.1±0.0	24.2	67.7	8.7	54.2
	STGAT-8	35.1±0.1	70.1±0.0	24.5	67.6	8.6	54.3
	STGAT-12	35.6±0.2	70.3±0.0	24.7	67.9	9.1	54.4
Multi	SGAN-8	36.0±0.2	70.3±0.0	24.8	67.9	9.2	54.4
	SGAN-12	36.0±0.3	70.3±0.0	24.9	67.9	9.3	54.4
	STGAT-8	36.2±0.3	70.3±0.0	24.5	67.8	8.8	54.3
	STGAT-12	36.4±0.1	70.4±0.0	24.8	67.9	9.2	54.4

Table 4. MOT-17 train forecasting ablations with state-of-the-art social forecasting models.

an online approach where groundtruth cannot be supplied to the algorithm. To simulate the online setting, we sample the mean trajectory from these approaches by requesting the trajectory corresponding to the zero noise vector. We calculate an approximate average scale factor of 20.0 between the trajectory values learnt by these models and the trajectory values available for input from our method, which we use to scale down our input values. Additionally, each of these methods has an 8- and 12-timestep forecasting model. In the main paper, we report the best of these models for both approaches and report other models in Table 4. For STGAT, the 8- and 12-timestep models used are trained on the ETH [16] dataset and for SGAN, the 8- and 12-timestep models are trained on the ZARA1 [10] dataset. Each of these models is made to predict for 30-timesteps by supplying the last 8 forecasted timesteps iteratively. The occlusion phase may not last 30 timesteps for all people. We therefore use the information from our pipeline about the number of occluded timesteps and replace the x and z values from the output of our pipeline with SGAN and STGAT's forecasted x and z values. In Table 4, we additionally report the performance of the methods when we provide past trajectories of *multiple* people as input, allowing the method to leverage social cues. For the Top-5 evaluation, we use the blind baseline described in Sec. 4 of our main paper. The conclusion remains that simple linear models suffice for short, frequent occlusions as our approach always performs better than any of the social forecasting settings of SGAN and STGAT.

1.4. Moving vs Stationary Camera Sequences

In the MOT-17 dataset, 3 camera sequences are stationary and 4 are captured from a moving camera. We separately study the effect of using different components of our pipeline on these sets of camera sequences. Table 5 shows that compensating for camera egomotion and filtering estimates lying in freespace helps the moving camera sequences by 4.5% and 4.0% Occluded Top-5 F1 respectively while for the sta-

	Top-5				Top-1 F1		IDF1	
	Occl F1	Occl Prec	Occl Rec	All F1	Occl	All	Occl	All
Moving sequences								
DeepSORT	27.3 ±0.3	49.7	18.8	72.4 ±0.0	17.3	67.0	2.2	56.5
+ Forecast	21.3 ±0.1	15.4	34.6	68.4 ±0.1	13.3	63.6	5.6	50.2
+ Egomotion	25.8 ±0.0	19.4	38.7	71.3 ±0.0	17.1	66.9	8.7	53.2
+ Freespace	29.8 ±0.3	28.0	31.8	72.8 ±0.0	19.9	69.2	9.4	55.2
+ Dep. noise	34.3 ±0.1	32.8	35.9	73.3 ±0.1	20.2	69.4	9.8	55.9
Stationary sequences								
DeepSORT	29.2 ±0.1	94.0	17.3	66.2 ±0.0	21.7	65.9	1.1	55.0
+ Forecast	39.1 ±0.4	62.2	28.5	70.2 ±0.0	28.7	68.6	10.1	55.4
+ Egomotion	38.0 ±0.1	60.2	27.8	69.8 ±0.0	28.5	68.5	9.6	55.3
+ Freespace	40.0 ±0.0	76.1	27.1	68.9 ±0.0	30.3	67.9	10.0	54.9
+ Dep. noise	43.6 ±0.3	78.7	30.2	68.8 ±0.0	31.4	67.9	11.2	54.1

Table 5. MOT-17 train ablations for moving *vs*. stationary camera sequences.

tionary camera sequences, enforcing smoother tracks for faraway objects and filtering freespace estimates helps by 3.6% and 2.0% F1 respectively.

1.5. Hyperparameter tuning

We describe a few parameters of our approach and how to tune them, in addition to the ones described in the paper. The N_{age} parameter in our pipeline controls the number of frames that an occluded track is forecasted for before it is deleted. We show in Figure 1 that the DeepSORT baseline is largely invariant to this parameter, as it does not report its internal forecasts. Reporting these estimates, whether directly (corresponding to 'DeepSORT+Forecast') or with our approach (corresponding to 'Our Pipeline'), highlights the impact of the parameter. This behaviour results in a precision-recall 'curvelet' which shows that by increasing N_{age}, we can trade-off the precision and recall for invisible people detection. The difficulty of this task can be highlighted by the trend that increasing N_{age} hardly increases recall beyond a point but instead decreases precision dramatically because of the introduction of many false positive boxes in the scene. We use the number of frames as a surrogate for uncertainty, as we find that this correlates well with the uncertainty estimated by the Kalman Filter, as shown in Figure 4 in the main paper.

We use a hyperparameter $f_{process}$ to scale the process noise covariance (refer Section 3.3 in the main paper). We additionally scale the observation noise covariance by $f_{observation}$ to account for the removal of default scaling by height of [18]. In our algorithm, we use $f_{process} = 900$ and $f_{observation} = 600$.

2. IDF1-Occluded & MOTA-Occluded

In the main paper, we report detection results using the probabilistic and standard F1 metrics. Here, we supplement

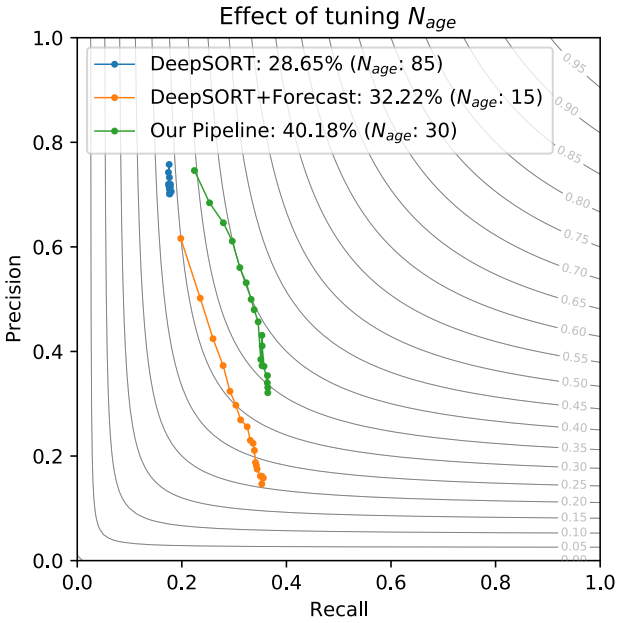

Figure 1. Detecting occluded people is sensitive to the threshold used to declare a detection-under-high-occlusion. We fix the number of N_{age} frames that a track is allowed to be in an occluded state. By increasing N_{age}, we can tradeoff precision and recall in invisible-people-detection which results in a "PR-curvelet". The curvelets represent the experiments in rows 1, 2 and 5 of Table 4 in the main paper.

these results with the IDF1 and MOTA (Multi-Object Tracking Accuracy) tracking metrics [2]. To do this, we follow the strategy in the main paper: We do not penalize tracks that match to visible people, but we reward only tracks that match to occluded people.

IDF1: To evaluate tracking, we report the standard IDF1 metric and also modify it for evaluating occluded people. Specifically, we divide the groundtruth tracks into visible and occluded segments, and perform matching only on the occluded segments. Once the tracks are matched, we compute IDTP as the number of matched occluded boxes, IDFP as the number of unmatched occluded *or* visible predictions, and IDFN as the number of unmatched occluded groundtruth boxes. In Tables 6, 7, 8, 9, we show that we improve the tracking of occluded people by a large margin (upto 14.3%) while maintaining the overall tracking performance. The conclusions in all cases remain the same as the detection metrics, except for the peculiar case of PANDA where we see an 8.1% drop in the overall IDF1 metric. We attribute this to the small size of people in PANDA and the top-down camera viewpoint which changes the distribution of the depth estimates returned by the monocular depth estimator. By tuning noise parameters to adapt to this new distribution, we can recover 6.9% of this drop.

Detections	Tracks	Occl Strat	Online?	IDF1	
				Occl	All
Groundtruth (vis.)	Groundtruth	Interpolate	✗	77.8	96.7
Faster R-CNN	Groundtruth	Interpolate	✗	20.5	67.4
Groundtruth (vis.)	DeepSORT	Interpolate	✗	21.3	81.0
Faster R-CNN	DeepSORT	Interpolate	✗	6.4	53.3
Faster R-CNN	DeepSORT	Forecast	✓	7.6	53.3

Table 6. Supplementary oracle ablations on MOT-17 train.

MOTA: In addition to reporting standard MOTA, we modify it for occluded tracks by counting detections matched to occluded groundtruth as true positives (TP), unmatched detections as false positives (FP), and unmatched groundtruth as false negatives (FN), and only count ID-switches (IDS) for tracks corresponding to occluded groundtruth. Perhaps surprisingly, we find in Table 9 that the MOTA metric is negative for all ablations. To better understand this, we note that MOTA is a simple combination of TP, FP, IDS, divided by the total number of groundtruth boxes (GT):

$$\text{MOTA} = 1 - \frac{\sum_t \text{FP}_t + \text{FN}_t + \text{IDS}_t}{\sum_t \text{GT}_t}$$

Thus, a method which simply reports no tracks will achieve a MOTA of 0 (as $\text{FP} = 0, \text{FN} = \text{GT}, \text{IDS} = 0$), seemingly outperforming all approaches in Table 9. This suggests MOTA penalizes methods for even *trying* to detect occluded people. In general, if a tracker produces more false positives than true positives, MOTA will always be negative! This indicates that MOTA is not an appropriate metric for challenging tasks, such as detecting occluded people.

3. Human Vision Experiment

In the main paper, we briefly described our human vision experiment to understand the challenges in detecting occluded people, and to motivate our evaluation and probabilistic approach. We provide further details here. We ask 10 in-house annotators to label fully occluded people in the MOT-17 [14] training set. To focus annotation effort on occluded people, we sampled track segments (1) containing at least 10 contiguous occluded frames, preceded by (2) 10 frames where the person is visible (and at least one where the person has $> 70\%$ visibility). Additionally, we avoid annotating small people (< 20 pixels on either side), and limit the number of total frames in a segment to 50.

Annotators labeled at 10 fps (every 3rd frame in a 30fps video) in a simulated *online* setup. When an annotator is asked to label frame t, she has access to past frames (before t), but *not* future frames $> t$. Once the annotator submits a label for t, she is shown the next frame to label, and is no longer allowed to edit the annotation for frame t.

Overall, 10 people labeled a total of 113 tracks, 46 of which were unique. This resulted in a total of 991 annotated boxes. Our key finding was that even for complete occlusions (less than 10% visibility), annotators still agreed to a fair extent (60% IoU-agreement), making the problem harder than localizing visible people, but still feasible for humans. To account for these observations, we evaluate with our invisible-people detection metric at an IoU of 0.5.

		IDF1	
		Occl	All
MOT-17	DPM	2.9	36.9
	+ Ours	7.2	36.8
	FRCNN	1.5	55.6
	+ Ours	10.5	54.8
	SDP	10.9	64.6
	+ Ours	17.0	64.7
	Tracktor++	1.3	65.1
	+ Ours	15.6	66.8
	MIFT	9.4	61.7
	+ Ours	16.5	62.6
	CTrack	5.4	65.0
	+ Ours	16.2	70.2
MOT-20	FRCNN	2.9	42.2
	+ Ours	5.0	42.0
PANDA	GT (visible)	2.5	70.2
	+ Ours	4.6	62.1

Table 7. Supplementary tracking results on MOT-17 [14], MOT-20 [3] and PANDA [17] train.

		IDF1	
		Occl	All
MOT-17	Ours	14.7	58.7
	MIFT [6]	10.4	56.4
	UnsupTrack [8]	9.7	62.6
	GNNMatch [15]	6.9	56.1
	GSM_Tracktor [13]	7.4	57.8
	Tracktor++ [1]	5.2	55.1
MOT-20	Ours	11.2	51.1
	Tracktor++ [1]	10.2	48.8
	UnsupTrack [8]	9.6	50.6
	SORT20 [18]	8.8	45.1

Table 8. Supplementary tracking results on MOT-17 and MOT-20 test set. The best, second-best and third-best methods are highlighted.

4. PANDA and MOT-20

We first discuss the quality of visibility labels in PANDA followed by the criteria we follow for disabling the depth

	IDF1		MOTA	
	Occl	All	Occl	All
DeepSORT	1.5	55.6	-11.9	49.4
+ Forecast	7.6	53.3	-85.7	42.0
+ Egomotion	9.1	54.5	-72.1	44.6
+ Freespace	9.7	55.0	-35.2	48.1
+ Dep. noise	10.5	54.8	-31.5	48.5

Table 9. Analysis of IDF1- and MOTA-occluded for the MOT-17 train ablations. Note that MOTA is not useful for distinguishing trackers for difficult tasks, as it leads to negative values (while an approach which reports no detections would achieve MOTA of 0).

and freespace reasoning in our method for a subset of videos in PANDA [17] and MOT-20 [3].

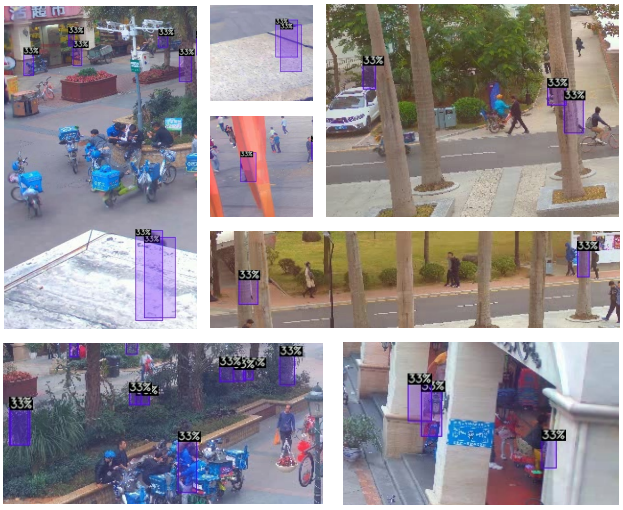

Figure 2. 'Heavy occlusion' or 33% visibility labels in PANDA are closer to the $< 10\%$ visibility labels in the MOT-17 and MOT-20 datasets. For this reason, we set the visibility threshold in the PANDA dataset to 33%.

PANDA classifies the visibility of people into 4 discrete classes – 'without occlusion', 'partial occlusion', 'heavy occlusion' and 'disappearing'. According to the dataset authors, these correspond to 100%, 66%, 33% and 0% visibility labels on a continuous 0-100 scale. On qualitative inspection, we find that most 33% visible people in PANDA are fully-occluded (by our definition of $< 10\%$ visibility). Though the visibility annotation protocol is not detailed in the paper, we hypothesize that this anomaly exists because only those people are marked with 0% visibility which strictly have 0 visible pixels. Some examples are shown in Figure 2. Owing to this, we set the threshold of calling a person invisible in the PANDA dataset as 33% visibility.

Some sequences in PANDA and MOT-20 are top-down view videos where occlusions are unlikely to occur. In such sequences, we revert to using the standard DeepSORT tracker. For MOT-20, we disable our method on two sequences captured from a camera mounted at a high height based on visual inspection. For the PANDA dataset, which specifies the building floor on which the camera is mounted, we use DeepSORT for cameras mounted on or above the 8th floor. We note that this decision can be easily made in the real world by practitioners based on the height of the camera.

5. Runtime & Pseudo-code

We precompute depth maps and detection features at 4.5 FPS and 11 FPS respectively. These are used in an online manner by our pipeline that runs at 4 FPS without explicit optimization. In Algorithm 1, we present the pseudocode of our approach.

References

[1] Philipp Bergmann, Tim Meinhardt, and Laura Leal-Taixe. Tracking without bells and whistles. In *ICCV*, 2019. 4

[2] Keni Bernardin and Rainer Stiefelhagen. Evaluating multiple object tracking performance: the clear mot metrics. *EURASIP Journal on Image and Video Processing*, 2008:1–10, 2008. 3

[3] Patrick Dendorfer, Hamid Rezatofighi, Anton Milan, Javen Shi, Daniel Cremers, Ian Reid, Stefan Roth, Konrad Schindler, and Laura Leal-Taixé. Mot20: A benchmark for multi object tracking in crowded scenes. *arXiv preprint arXiv:2003.09003*, 2020. 4, 5

[4] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social gan: Socially acceptable trajectories with generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2255–2264, 2018. 2

[5] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *ICCV*, 2017. 2

[6] Piao Huang, Shoudong Han, Jun Zhao, Donghaisheng Liu, Hongwei Wang, En Yu, and Alex ChiChung Kot. Refinements in motion and appearance for online multi-object tracking. *arXiv preprint arXiv:2003.07177*, 2020. 4

[7] Yingfan Huang, HuiKun Bi, Zhaoxin Li, Tianlu Mao, and Zhaoqi Wang. Stgat: Modeling spatial-temporal interactions for human trajectory prediction. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6272–6281, 2019. 2

[8] Shyamgopal Karthik, Ameya Prabhu, and Vineet Gandhi. Simple unsupervised multi-object tracking. *arXiv preprint arXiv:2006.02609*, 2020. 4

[9] Katrin Lasinger, René Ranftl, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *arXiv preprint arXiv:1907.01341*, 2019. 1, 2

[10] Alon Lerner, Yiorgos Chrysanthou, and Dani Lischinski. Crowds by example. In *Computer graphics forum*, volume 26, pages 655–664. Wiley Online Library, 2007. 2

Algorithm 1: Invisible-people Kalman Tracker

Data: Detections $\mathcal{D}$ in current frame, $f_i \in \mathcal{F}$, the set of all frames

Result: Set of active tracks, $\mathcal{T} = \{t_1, \ldots, t_k\}$ s.t. $t_j \in \{\mathcal{T}_{occluded}, \mathcal{T}_{visible}\}$

def *update()*:
- X, Y1, Y2, Z = *match()*;
- Update the tracks with the KF Update step for all pairs in X;
- Initialise new tracks for Z;
- Increase age of all tracks in Y1;
- Add Y2 to $\mathcal{T}_{occluded}$;

def *match()*:
- Compare forecasted depth, z_f with horizon depth, z_o;
- If $z_f < \alpha_{supp} z_o$, keep track in $\mathcal{T}_{visible}$ but don't output;
- Else, trigger occluded state logic by adding track to $\mathcal{T}_{occluded}$;
- Bipartite-match detections to active tracks to based on last-known appearance;
- Match unclaimed visible tracks to unclaimed detections using IoU;
- Let X be matched tracks and detection;
- Let Y be unclaimed tracks;
- Let Z be unclaimed detections;
- Separate Y into visible (Y1) and occluded (Y2) tracks;
- **for** *all tracks in Y2* **do**
  - If $z_f < \alpha_{delete} z_o$, delete track;
- **end**
- Return X,Y1,Y2,Z;

def *predict()*:
- Find warp marix W between current and past frame;
- **for** *all active tracks* **do**
  - Warp the mean of current tracker state with the warp matrix;
  - Assume a Constant Velocity Model;
  - If track is occluded, assume no velocity for a and h;
  - Else, assume constant velocity for a and h;
  - Assume temporal process noise for all state variables (e.g., process noise $f\frac{\epsilon_X}{Z}$ for x);
  - Carry out the KF Predict step to get a new state from the warped state;

def *main()*:
- **for** *every incoming frame* **do**
  - predict new states for all tracks using *predict()*;
  - update all tracks with detections from the current frame using *update()*;
  - output all active tracks that are either currently occluded or visible;

[11] Zhengqi Li, Tali Dekel, Forrester Cole, Richard Tucker, Noah Snavely, Ce Liu, and William T Freeman. Learning the depths of moving people by watching frozen people. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4521–4530, 2019. 1

[12] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2041–2050, 2018. 1

[13] Qiankun Liu, Qi Chu, Bin Liu, and Nenghai Yu. Gsm: Graph similarity model for multi-object tracking. International Joint Conferences on Artificial Intelligence Organization, 2020. 4

[14] Anton Milan, Laura Leal-Taixé, Ian Reid, Stefan Roth, and Konrad Schindler. Mot16: A benchmark for multi-object tracking. *arXiv preprint arXiv:1603.00831*, 2016. 4

[15] Ioannis Papakis, Abhijit Sarkar, and Anuj Karpatne. Gcnnmatch: Graph convolutional neural networks for multi-object tracking via sinkhorn normalization. *arXiv preprint arXiv:2010.00067*, 2020. 4

[16] Stefano Pellegrini, Andreas Ess, Konrad Schindler, and Luc Van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *ICCV*. IEEE, 2009. 2

[17] Xueyang Wang, Xiya Zhang, Yinheng Zhu, Yuchen Guo, Xiaoyun Yuan, Liuyu Xiang, Zerun Wang, Guiguang Ding, David Brady, Qionghai Dai, et al. Panda: A gigapixel-level human-centric video dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3268–3278, 2020. 4, 5

[18] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *ICIP*, 2017. 3, 4