Detecting Invisible People

Tarasha Khurana

CMU-RI-TR-21-24

August 19, 2021

The Robotics Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA

Thesis Committee:
Prof. Deva Ramanan, *advisor*
Prof. Simon Lucey
Prof. Katerina Fragkiadaki
Dr. Achal Dave

Submitted in partial fulfillment of the requirements for the degree of Master of Science in Robotics.

To my brother.

Abstract

Monocular object detection and tracking have improved drastically in recent years, but rely on a key assumption: that objects are visible to the camera. Many offline tracking approaches reason about occluded objects *post-hoc*, by linking together tracklets after the object re-appears, making use of reidentification (ReID). However, online tracking in embodied robotic agents (such as a self-driving vehicle) fundamentally requires object permanence, which is the ability to reason about occluded objects *before* they re-appear. In this work, we re-purpose tracking benchmarks and propose new metrics for the task of detecting invisible objects, focusing on the illustrative case of people. We demonstrate that current detection and tracking systems perform dramatically worse on this task. We introduce two key innovations to recover much of this performance drop. We treat occluded object detection in temporal sequences as a short-term forecasting challenge, bringing to bear tools from dynamic sequence prediction. Second, we build dynamic models that explicitly reason in 3D from monocular videos without calibration, using observations produced by monocular depth estimators. To our knowledge, ours is the first work to demonstrate the effectiveness of monocular depth estimation for the task of tracking and detecting occluded objects. Our approach strongly improves by 11.4% over the baseline in ablations and by 5.0% over the state-of-the-art in F1 score.

As will be described, our approach is dependent on good *amodal* detectors and plausible monocular depth estimates. In this regard, we explore two directions of future work. First, we note that no video dataset exists that focuses explicitly on amodal object annotations across their tracks, although this is a more reasonable object detection task as objects do not cease to exist where their visual footprint ends. We propose TAO-Amodal, an extension of our older work, TAO. Second, we note that a seemingly harmful protocol in depth estimation inference is to downsample input. This leads to loss of high-frequency details in images, and important objects like people and vehicles in high-resolution images. We probe a few hypotheses in this direction.

Acknowledgments

I'm the most thankful to my advisor, Prof. Deva Ramanan, for his patience and guidance in the last two years. Deva gave me a chance to speak with him every week without fail, and with each meeting the horizon of my knowledge broadened. He played an important role in increasing my inclination towards research in general and this problem statement in particular. Without his supervision, honest feedback and encouragement to pursue challenging directions, this work would not have been possible.

I would like to thank Dr. Achal Dave, for first, being a friend, and second, being a mentor. Most of the ground work behind honing my presentation, experimentation, coding and writing skills came from Achal. Achal graciously cleared my doubts at odd times in the night and was *always* available for help. My thesis is what it is today because of his constant support.

Next, I want to thank Prof. Simon Lucey, Prof. Katerina Fragkiadaki and Prof. Laura Leal Taixe for insighful discussions on this project; for understanding and critiquing the problem statement and approach. Their conversations helped me refine the motivation for this project. This thesis would also have been incomplete without the in-depth reviews from internal reviewers at CMU, specifically the Smith Hall vision group. We also could not have understood the cognitive ability of humans behind solving this task without the participation of volunteers of the human vision experiment. A big thanks to them for taking the time!

Lastly, I would like to thank my family, including the new member, Akash Sharma, for being my sounding board and supporting me during my nervous breakdowns. I love my brother the most and he keeps me on my toes in maintaining sharp aptitude skills. Thank you, Mehar Khurana!

Funding

This work was supported by the CMU Argo AI Center for Autonomous Vehicle Research, the Defense Advanced Research Projects Agency (DARPA) under Contract No. HR001117C0051, and the National Science Foundation (NSF) under Grant No. IIS-1618903.

Contents

When this dissertation is viewed as a PDF, the page header is a link to this Table of Contents.

List of Figures

List of Tables

Chapter 1

Introduction

Object detection has seen immense progress, albeit under a seemingly harmless assumption: that objects are *visible to the camera* in the image. However, objects that are fully occluded (and thus, invisible) continue to exist and move in the world. Indeed, object permanence is a fundamental visual cue exhibited by infants in as early as 3 months [3, 26]. Practical autonomous systems must similarly reason about objects under such occlusions to ensure safe operation (Figure 1.1). Interestingly, existing work on object detection and tracking tends to de-emphasize this capability, either choosing to completely ignore highly-occluded instances for evaluation [17, 40, 52, 60], or simply downweighting them because they occur so rarely that they fail to materially affect overall performance [43]. One reason that invisible-object detection may have been under-emphasized in the tracking community is that for *offline* analysis, one can post-hoc reason about the presence of an occluded object by relinking detections *after* it reappears. This approach has spawned the large subfield of reidentification (ReID). However, in an *online* setting (such as an autonomous vehicle that must make decisions given the available sensor information), intelligent agents must be able to instantaneously reason about occluded objects *before* they re-appear.

1.1 Problem formulation

We begin by introducing benchmarks and metrics for evaluating the task of detecting and tracking invisible people. To do so, we repurpose existing tracking benchmarks

Figure 1.1: We visualize an online tracking scenario from Argoverse [10] that requires tracking a pedestrian through a complete occlusion. Such applications cannot wait for objects to re-appear (, as re-identification approaches do): autonomous agents must properly react *during* the occlusion. We treat online detection of occluded people as a *short-term forecasting* challenge.

and introduce metrics for evaluating this task that appropriately reward detection of occluded people. To ensure benchmarks are online, we forbid algorithms from accessing future frames when reporting object states for the current frame. Although this task requires reasoning about object trajectories, it can be evaluated as both a *detection* and a *tracking* problem. For the latter, we introduce extensions to tracking metrics later in the thesis. When analyzing our metrics, it becomes readily apparent that human annotation of ground-truth occluded objects is challenging. We provide pilot human vision experiments in experiments chapter that show annotators are still consistent, but exhibit larger variation in labeling the pixel position of occluded instances. This suggests that algorithms for occluded object detection should report *distributions* over object locations rather than precise discrete (bounding box) locations. Inspired by metrics for evaluating multimodal distributions in the forecasting literature [10], we explore probabilistic algorithms that make k predictions which are evaluated by Top-k accuracy.

1.2 Analysis

Perhaps not surprisingly, our first observation is that performance of state-of-the-art detectors and trackers plummets on occluded people, from 68.5% to 28.4%; it is far easier to detect visible objects than invisible ones! This underscores the need for the community to focus on this underexplored problem.

We introduce two simple but key innovations for addressing this task, which improve performance from 28.4% to 39.8%. (a) We recast the problem of online tracking of occluded objects as a *short-term forecasting* challenge. We explore state-of-the-art deep forecasting networks, but find that classic linear dynamics models (Kalman filters) perform quite well. (b) Because modeling occlusions is of central importance, we cast the problem as one of 3D tracking given 2D image measurements.

1.3 Novelty

While there exists considerable classic work on 3D tracking from 2D [8, 11, 50, 54], much focuses on 3D modeling of tracked objects. Instead, we find that the 3D structure of scene occluders is important for understanding where tracked objects can "hide". Typically such dense 3D understanding requires calibrated multiview sensors [15, 55]. Instead, we show that recent advances in uncalibrated *monocular depth estimation* provide "good enough" estimates of relative depth that still enable dense freespace reasoning.

This is crucial because monocular depth has the potential to be far more scalable [57]. To our knowledge, ours is the first work to use uncalibrated depth estimates for multi-object tracking and detection of occluded objects.

1.4 Overview

After reviewing related work, we present our core algorithmic contributions, including straightforward but crucial extensions to classic linear dynamics models to (a) incorporate putative depth observations from a monocular network and (b) forecast object state even during occlusions. We conclude with extensive evaluations on three datasets [13, 43, 56] repurposed for detecting occluded objects.

Chapter 2

Related Work

2.1 Amodal object detection

Amodal object detection aims to segment the full extent of objects that may be partially (but not *fully*) occluded. [67] introduces this task with a dataset labeled by multiple annotators, which is later expanded by [68]. More recently, [48] introduces a larger dataset of amodal annotations on the KITTI [21] dataset. Approaches in this setting largely rely on training variants of standard detectors ([24]) on amodal annotations generated synthetically from modal datasets [14, 37, 62, 65]. As this line of work addresses detection from a single image, it requires objects to be at least *partially visible*. By contrast, we target fully occluded people, which cannot be recovered from a single frame.

2.2 Multi-object tracking

Multi-object tracking requires tracking across partial and full occlusions. Approaches for this task address occlusions post-hoc in an *offline* manner, using appearance-based re-identification models to identify occluded objects after they become visible. These appearance-based models can be incorporated into tracking approaches, as part of a graph optimization problem [4, 47, 64] or online linking [5, 58]. In this work, we point out that some approaches *internally* maintain online estimates of the position of

occluded people [5, 7, 58], but explicitly choose not to report these internal predictions, as they tend to be noisy and, thus, are penalized heavily by current benchmarks. We provide two simple extensions to these internal predictions that significantly improve detection of occluded people while preserving accuracy on visible people. [22] tracks occluded objects using contextual 'supporters', but requires a user to initialize a single object to track in uncluttered scenes; by contrast, we simultaneously detect and track people in large crowds.

Other work shares our motivation of tracking in 3D but relies on additional depth sensors [20] or stereo setups [9, 29]. Finally, many surveillance-based tracking systems explicity reason about object occupancy and occlusion, but require calibrated cameras to compute ground plane coordinates [1, 19, 28, 31, 32]. By contrast, our work emphasizes detection of *occluded* people in *uncalibrated, monocular* videos. To do so, we use monocular depth estimators via technical innovations that address noise in predicted depth estimates. Our method generalizes to arbitrary videos, since estimating monocular depth is far more scalable than retrieving additional sensor information for any video.

2.3 Forecasting

Forecasting approaches predict pedestrian trajectories in future, unobserved frames. These approaches leverage social cues from nearby pedestrians or semantic scene information to better model person trajectories [33, 35, 42, 46, 53, 61]. Recently, data-driven approaches have also been proposed for learning social cues [2, 51]. We note that detection of fully occluded people can be formulated as forecasting the trajectory of a visible person in future frames, where the positions of the occluded person are unobserved, but the rest of the frame *can* be observed. Our approach uses a constant-velocity model to forecast trajectories, equipped with depth cues from the observed frames, to improve detection of occluded people. In Section 4.7, we show that while this approach can use a more powerful forecasting model, the constant-velocity approximation is sufficient in our setting.

Chapter 3

Method

We build an online approach for detecting invisible people starting with a simple tracker, using estimated trajectories of visible people to forecast their location during occlusions. We describe our tracking mechanism, building upon [59]. While such trackers *internally* forecast the location of occluded people for improved tracking, these forecasts tend to be noisy and cannot directly localize occluded people. To address this, we incorporate depth cues from a monocular depth estimator to reason about occlusions in 3D.

3.1 Background

To detect people during occlusions, we build on a simple online tracker [59] that estimates the trajectories of visible people. We briefly describe aspects relevant to our approach, but refer the reader to [59] for a more detailed explanation. In the first frame, this tracker instantiates a track for each detected person. The tracker adds each track to its "active" set, representing people that have been seen so far. Each track maintains a Kalman Filter whose state space encodes the position (x, y), aspect ratio (a), height (h), and corresponding velocities ($\dot{x}, \dot{y}, \dot{a}, \dot{h}$) of the person.The filter's process model assumes a constant velocity model with gaussian noise (i.e., $x_t = x_{t-1} + \dot{x_{t-1}} + \epsilon_x$). At each successive frame, the tracker first runs the *predict* step of the filter, using the process model to forecast the location of the track in the new frame. Next, each detection in the current frame is matched to this set of

active tracks based on appearance features, and distance to the tracks' forecasted location (as estimated by the filter). A new track is created for all detections that are unmatched. If a track is matched to a detection, the detection is used as a new observation to update the track's filter, and the detection is reported as part of the track. Importantly, if a track does not match to any detection, its forecasted box is *not* reported. When a track is not matched to a detection for more than N_{age} frames, it is deleted.

3.2 Short-term forecasting across occlusions

Although this tracker *internally* forecasts the positions of all tracks at each step, its estimates are used only to improve the association of tracks to detections, and are not reported externally. However, these internally forecasted track locations are crucial as they may correspond to an occluded person. We show that naively reporting these track locations leads to significant *recall* of occluded people, but the noise in these estimates results in poor precision. Further, these noisy estimates lead to a small decrease in *overall* accuracy, as standard benchmarks largely focus on visible people. We improve these estimates by augmenting them with 3D information. Specifically, we use a monocular depth estimator [38] to get per pixel depth estimates of the scene. We then augment our Kalman Filter state space with the *inverse* depth. Inverse depth is a commonly used representation predicted by depth estimators [34, 38] due to important benefits, including the ability to represent points at infinity and ability to model uncertainty in pixel disparity space (commonly used for stereo-based depth estimation [44]). Our state space thus additionally includes $1/z$ variable.

3.3 Tracking in 3D camera coordinates using 2D image coordinates

Equipped with depth estimates, we formulate tracking with a constant velocity model in 3D using 2D measurements. Unlike prior work which assumes linear dynamics in (projected) 2D image measurements, our dynamics model operates in 3D using depth cues, resulting in far more realistic person trajectories. We derive our uncalibrated

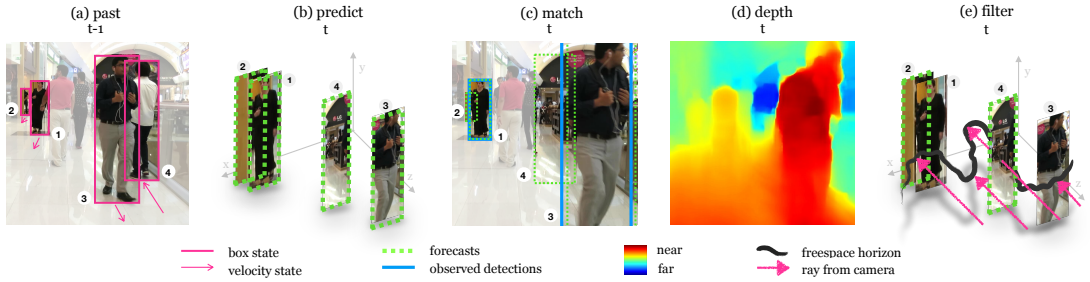

Figure 3.1: (a) Frame $t-1$ has active tracks $\{1, 2, 3, 4\}$, each with an internal state of its 2D position, size, velocity, and *depth* (see text). (b) We forecast tracks in 3D for frame t. (c) Tracks are matched to observed detections at t using spatial and appearance cues. Matched tracks are considered visible (1, 3). Tracks which don't match to a visible detection (2, 4) may be occluded, or simply incorrectly forecasted. (d) To resolve this ambiguity, we leverage depth cues from a monocular depth estimator, to compute (e) the *freespace horizon*. The region between the camera and the horizon must be freespace, while the area beyond it is unobserved, and so may contain *occluded* objects. Tracks lying beyond the freespace horizon are reported as occluded (2). Tracks *within* freespace (4) should have been visible, but did not match to any visible detections. Hence, we assume these tracks are incorrectly forecasted, and we delete them.

tracker by demonstrating that the unknown camera focal length f can be folded into a motion noise parameter that can be easily tuned on a training set. Hence our final method runs without calibration on arbitrary videos.

Let us model objects as cylinders with centroids (X_t, Y_t, Z_t), height H and aspect ratio A_t. We model object height as constant, but allow for varying aspect ratios because people are non-rigid. We can then compute image-measured bounding boxes with centroid (x_t, y_t) and dimensions (h_t, a_t) as follows:

$$x_t = f\frac{X_t}{Z_t}, \quad y_t = f\frac{Y_t}{Z_t}, \quad h_t = f\frac{H}{Z_t}, \quad a_t = A_t \tag{3.1}$$

We extend the commonly used constant velocity model with Gaussian noise from 2D [7, 58] to 3D:

$$X_t = X_{t-1} + \dot{X}_{t-1} + \epsilon_X, \quad \epsilon_X \sim \mathcal{N}(0, \sigma_X), \tag{3.2}$$

where similar equations hold for Y_t, Z_t and A_t. Let the observed (inverse) depth from a depth estimator associated with an object be $1/z_t$. Since image measurements

are given by perspective projection of real world coordinates, we have the following equations (assuming Gaussian image noise):

$$x_t = f\frac{X_t}{Z_t} + \epsilon_x, \quad \epsilon_x \sim \mathcal{N}(0, \sigma_x) \tag{3.3}$$

$$\frac{1}{z_t} = \frac{1}{Z_t} + \epsilon_z, \quad \epsilon_z \sim \mathcal{N}(0, \sigma_z) \tag{3.4}$$

with similar equations for y_t, h_t, and a_t. Note that inverse depth naturally assumes a large uncertainty in far away regions, and a small uncertainty in nearby regions. Defining a 3D state space leads us to a modified formulation, written as

$$\left(f\frac{X_t}{Z_t}, f\frac{Y_t}{Z_t}, \frac{1}{Z_t}, A_t, f\frac{H}{Z_t}, f\frac{\dot{X}_t}{Z_t}, f\frac{\dot{Y}_t}{Z_t}, \dot{A}_t\right) \tag{3.5}$$

We can therefore rewrite Equation (3.2) as:

$$f\frac{X_t}{Z_t} \approx f\frac{X_t}{Z_{t-1}} = f\frac{X_{t-1}}{Z_{t-1}} + f\frac{\dot{X}_{t-1}}{Z_{t-1}} + f\frac{\epsilon_X}{Z_{t-1}} \tag{3.6}$$

$$x_t \approx x_{t-1} + \dot{x}_{t-1} + f\frac{\epsilon_X}{Z_{t-1}} \tag{3.7}$$

where the approximation holds if depths are smooth over time ($Z_t \approx Z_{t-1}$). Technically, the above is no longer a linear dynamics model since the noise depends on the state. But the equation suggests that *one can approximately apply a Kalman filter on 2D image measurements augmented with a temporal noise model that is scaled by the estimated inverse-depth of the object*. Intuitively, this suggests that one should enforce smoother tracks for objects far away. Our approach thus scales the process noise (ϵ_X) for far away objects, leading to more accurate predictions. Algorithmically, [59] by default scales process and observation noise covariances according to the person's height; our approach instead multiplies the process covariance by the person's estimated depth, computed by aggregating past monocular depth observations and state estimates over time.

Assumptions. Because we do not assume calibrated cameras, we do not know f. Rather, we make use of training videos provided in standard tracking benchmarks and simply tune scaled variances $\sigma'_X = f\sigma_X$ directly on the training set. We make

two additional assumptions: that people move with constant velocity in 3D, and that depth estimates are smooth over time. Although these do not always hold in real world scenarios, we empirically find that our method generalizes to diverse scenarios.

Filtering estimates lying in freespace. Equipping our state space with depth information allows us to forecast 3D trajectories. Meanwhile, applying a monocular depth estimator allows us to determine regions in 3D space that are occluded to the camera without requiring calibration. Specifically, if our approach forecasts a person at a point $P_f = (x_f, y_f, z_f)$, we can determine whether P_f should be visible to the camera by estimating whether P_f lies in the freespace [15] between the camera and its nearest occluder. In the filter stage in Figure 3.1, we visualize one slice of the "freespace horizon": points beyond this horizon are occluded, while points between the camera and the horizon should be visible.

Concretely, let z_o be the (observed) depth of the horizon at (x_f, y_f). If the forecasted depth (z_f) lies closer to the camera than the horizon depth (z_o), as with person "4" in Figure 3.1 (e), then the person must be in the *freespace* between the camera and its closest object, and therefore visible. If we *do not* detect this person, then we assume the forecast is an error, and either suppress the forecasted box for the current frame (in the case of small errors, when $z_f < \alpha_{\text{supp}} z_o$) or delete the track entirely (for large errors, when $z_f < \alpha_{\text{delete}} z_o$). A key advantage of this approach is the ability to reason about occlusions arising not only from interactions between tracked people, but also from natural occluders such as trees or cars. Section 4.7 shows that this modification is critical for improving the precision of our trajectory forecasts.

Camera motion. Camera motion is challenging, as our approach assumes linear dynamics for trajectories. To address this, we follow prior work (e.g., [5]) in estimating a non-linear pixel warp W between neighboring frames which maps pixel coordinates (x_{t-1}, y_{t-1}) in one frame to the next (x_t, y_t). This warp is then used to align boxes forecasted using frames up to $t-1$ with frame t. Note that this alignment assumes the motion of dynamic objects is small relative to the scene motion, allowing for the use of an image registration algorithm [16]. Despite the simplicity of this modification, we show that it helps considerably for the moving camera sequences. In Algorithm 1, we present the pseudocode of our approach for detecting occluded people. Execution starts from the `MAIN()` function.

Algorithm 1 Invisible People Kalman Tracker

Detections $\mathcal{D}$ in current frame, $f_i \in \mathcal{F}$, the set of all frames
Set of active tracks, $\mathcal{T} = \{t_1, \ldots, t_k\}$ s.t. $t_j \in \{\mathcal{T}_{occluded}, \mathcal{T}_{visible}\}$

procedure 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 $\mathcal{T}_{occluded}$
end procedure

procedure MATCH()(X, Y1, Y2, Z)
    Compare forecasted depth, z_f with horizon depth, z_o
    if $z_f < \alpha_{supp} z_o$ **then**
        keep track in $\mathcal{T}_{visible}$ but don't output
    else
        trigger occluded state logic by adding track to $\mathcal{T}_{occluded}$
    end if
    Bipartite-match detections to tracks with 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 $z_f < \alpha_{delete} z_o$ **then**
            delete track
        end if
    end for
    return X,Y1,Y2,Z
end procedure

procedure 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 scaled process noise for all variables (e.g., f·ex/Z for x)
        Carry out the KF Predict step to get a new state from warped state
    end for
end procedure

procedure 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
    end for
end procedure
```

We now proceed to an empirical analysis of the task and prior methods, showing the benefits of each component of our proposed approach.

Chapter 4

Experimental Results

We first describe our proposed benchmarks, including the datasets and our proposed metrics for evaluating the task of detecting occluded people. Next, we conduct an oracle study in Section 4.5 to analyze how well existing approaches can detect occluded people. We then compare our proposed approach to these state-of-the-art approaches in multiple settings in Section 4.6. Finally, we analyze each component of our approach with a detailed ablation study in Section 4.7.

4.1 Human Vision Experiment

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 [43] 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.

4.2 Datasets

Evaluating our approach is challenging, as most datasets do not annotate occluded objects. The MOT-17 [43], MOT-20 [13] and PANDA [56] datasets are key exceptions which label both visible and occluded people, along with a *visibility* field indicating what portion of the person is visible to the camera. We find that a majority of the annotations in these datasets (over 85% in each dataset) are people that are at least partially visible, leading standard evaluations on these datasets to underemphasize occluded people. To address this, we separately evaluate accuracy on the subset of fully *occluded* people (indicated by $< 10\%$ visibility). MOT-17 contains 7 sequences with publicly available groundtruth, and 7 test sequences with held-out groundtruth. We evaluate on these 14 sequences. MOT-20 contains 8 sequences, of which 4 have held-out groundtruth. PANDA officially releases a high-resolution 2FPS groundtruth for its 10 train and 5 test sequences. Because tracking and forecasting is challenging at such low frame rates, we reached out to the authors who provided a high-frame rate (30FPS), low-resolution groundtruth for 9 train videos. We report results on MOT-20 and PANDA train set without tuning our pipeline on any of the videos in these datasets. From visual inspection, we found that visibility labels in PANDA tend to be noisy, and so we define objects with up to 33% visibility as occluded. We carry out the analysis including oracle and ablation study on MOT-17 train and report the final results on MOT-17 test, MOT-20 and PANDA datasets. In all, these three datasets target a diverse set of application scenarios – static surveillance cameras, car-mounted cameras, and hand-held cameras.

4.2.1 PANDA and MOT-20

We first discuss the quality of visibility labels in PANDA followed by the criteria we follow for disabling the depth and freespace reasoning in our method for a subset of videos in PANDA [56] and MOT-20 [13].

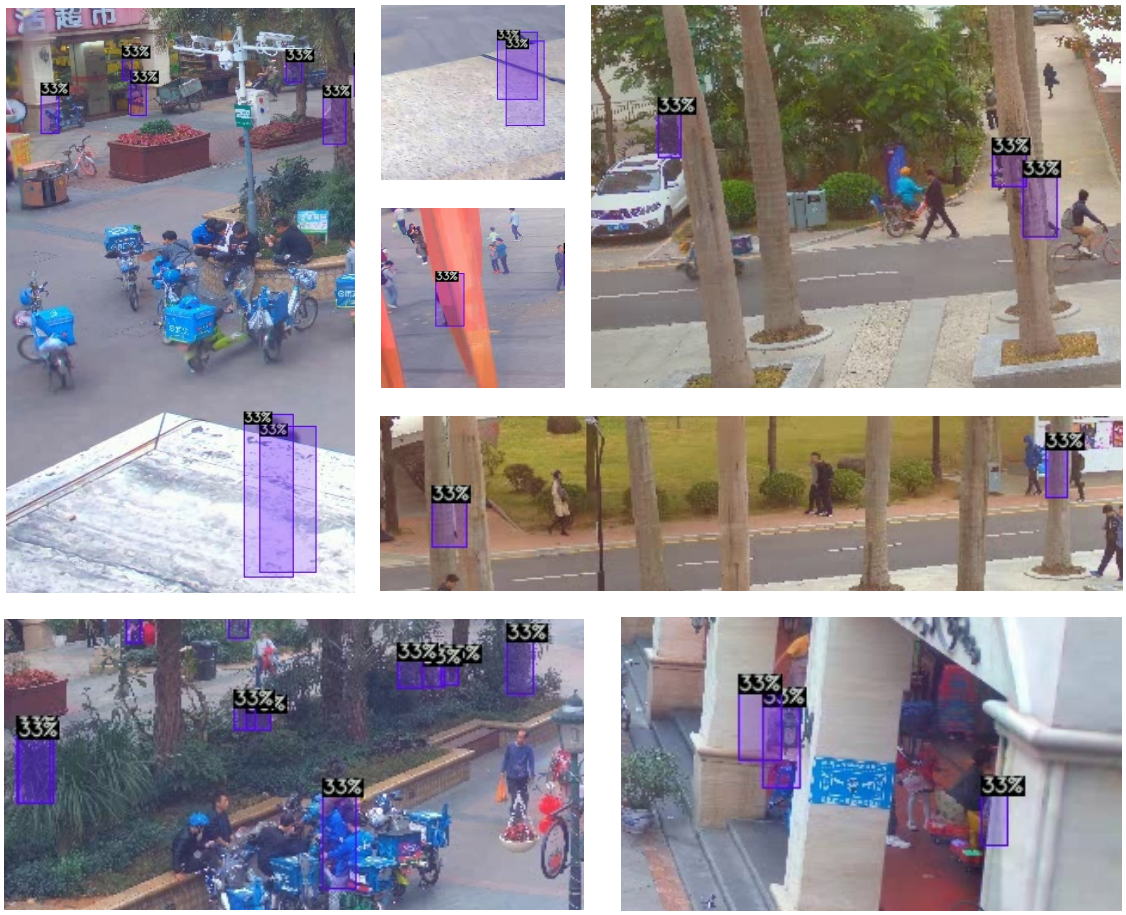

Figure 4.1: 'Heavy occlusion' or 33% visibility labels in PANDA are closer to the $< 10\%$ visibility labels in the MOT-17 and MOT-20 datasets. For this reason, we set the visibility threshold in the PANDA dataset to 33%.

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

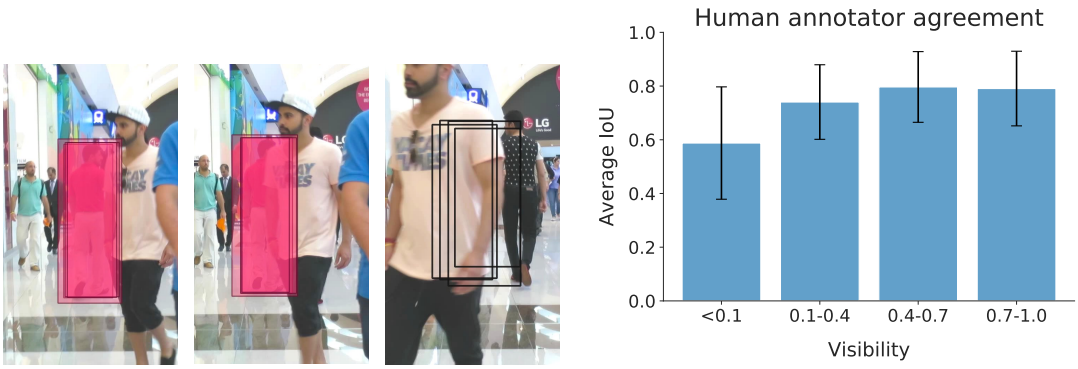

Figure 4.2: We visualize bounding boxes labeled by multiple (4) in-house annotators (**left**). During small occlusions, annotators strongly agree. During large occlusions (less than 10% visible, last frame), annotators still agree to a fair extent (average IoU overlap of 60%, **right**), but require temporal video context. We use these to justify our Top-k evaluation and motivate our probabilistic tracking approach.

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 4.1. 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.

4.3 Metrics

As most benchmarks consist primarily of visible people, existing metrics which measure performance across all people underemphasize the accuracy of detecting occluded people. We propose detection and tracking metrics which evaluate accuracy on occluded people, as indicated by visibility $< 10\%$ and on all (visible and invisible)

Table 4.1: Oracle ablations on MOT-17 train reporting Top-5 F1, Top-1 F1 and IDF1 for occluded and all people, using Faster R-CNN detections. 'Occl strat' stands for Occlusion Strategy. We report the Top-5 mean and standard deviation for 3 runs.

Detections	Tracks	Occl Strat	Online?	Top-5				Top-1 F1	
				Occl F1	Occl Prec	Occl Rec	All F1	Occl	All
Groundtruth (vis.)	Groundtruth	Interpolate	✗	87.3 ±0.1	83.8 ±0.2	91.1 ±0.1	98.0 ±0.0	79.8	96.8
Faster R-CNN	Groundtruth	Interpolate	✗	46.4 ±0.1	65.5 ±0.1	35.9 ±0.1	70.5 ±0.0	34.4	68.1
Groundtruth (vis.)	DeepSORT	Interpolate	✗	53.3 ±0.2	86.7 ±0.1	38.5 ±0.2	92.3 ±0.0	44.4	92.0
Faster R-CNN	DeepSORT	Interpolate	✗	32.2 ±0.0	60.8 ±0.2	21.9 ±0.0	69.9 ±0.0	23.2	68.4
Faster R-CNN	DeepSORT	Forecast	✓	29.8 ±0.2	29.5 ±0.4	30.2 ±0.1	69.4 ±0.0	20.9	66.5

Table 4.2: Supplementary oracle ablations on MOT-17 train.

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

people. Since localizing fully-occluded people involves higher positional uncertainty than visible people, we allow algorithms to predict k potential locations for each person.

4.3.1 Top-k F1

We start by modifying the standard detection evaluation protocol [17, 40]. For every person, we allow methods to report k predictions, $P = \{p_1, p_2, \ldots, p_k\}$. We match these predictions to all groundtruth boxes based on intersection-over-union (IoU). We define the overlap between a groundtruth g and P as the maximum overlap with the predictions p_i in P — , $\text{IoU}(g, P) = \max_i \text{IoU}(g, p_i)$. We use this overlap definition and perform standard matching between predictions and groundtruth, with a minimum overlap threshold of α_{IoU}.

When evaluating accuracy across all people, matched groundtruth boxes are true positives (TP), all unmatched groundtruth are false negatives (FNs, or misses), and

unmatched detections are false positives (FP). When evaluating accuracy on occluded people, only matched *occluded* groundtruth boxes count as TPs, only unmatched *occluded* groundtruth boxes count as FNs, and all unmatched detections count as FPs. Intuitively, when evaluating metrics for occluded people, we do not penalize a detector for correctly detecting a visible person, but we *do* penalize it for false positives that do not match any visible or occluded person.

We now describe how the k-vector of predictions is obtained: in addition to a state mean (first sample), our probabilistic method maintains covariances for x and z state variables which result in a 2D gaussian. Since these gaussians may extend incorrectly into freespace, we perform rejection sampling to accumulate k-1 predictions which respect freespace constraints. This gives us P. For baseline methods that are not probabilistic or do not have access to a depth map, we artificially simulate this distribution by tuning two scale factors that control the size of gaussians as a function of a bounding box's height. We tune these scale factors on MOT-17 train and use them throughout experiments.

4.3.2 Top-1 F1

When $k = 1$, this metric is simply the standard F1 metric. We additionally report this Top-1 F1 for occluded and *all* people. We do not use the standard 'average precision' (AP) metric as most detectors and trackers on the MOT and PANDA datasets do not report confidences.

4.3.3 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. We similarly modify MOTA later.

To guide evaluation, we conduct a human vision experiment with 10 in-house annotators who annotated 991 boxes in 59 tracks with occlusion phases. Figure 4.2

shows that annotators have lower consistency when labeling occluded people than visible people. To address this ambiguity in localizing occluded people, we choose a low $\alpha_{IoU} = 0.5$ and $k = 5$ in our experiments.

4.4 Implementation details

We empirically set parameters in our approach on MOT-17 train with Faster R-CNN [49] detections. The optimal thresholds for filtering forecasts on the train set are $\alpha_{\text{delete}} = 0.88, \alpha_{\text{supp}} = 1.06$[1]. During occlusion we treat a person as a point, freezing its aspect ratio and height. We fix N_{age} to 30. Further details of our method, parameters and their tuning protocol, including improvements by tuning N_{age} have also been covered. We tune on MOT-17 train and apply these tuned parameters on MOT-17 test, MOT-20, and PANDA. We find that our method and its hyperparameters tuned on the train set generalize well to the test set. We use [38] for monocular depth estimates, which has been shown to work well in the wild. While these estimates can be noisy, we qualitatively find that the *relative* depth orderings used in our approach are fairly robust.

4.5 Oracle Study

4.5.1 What is the impact of *visible* detection on occluded detection?

We first evaluate an offline approach which uses groundtruth detections and tracks for visible people to (linearly) interpolate detections for occluded people in Table 4.2. As this method perfectly localizes visible people, and most people in this benchmark are visible, it achieves a high overall Top-5 F1 of 98.0 (Table 4.2, row 1). Additionally, despite using simple linear interpolation, this oracle also achieves a high Top-5 F1 of 87.3 for *invisible* people. This result indicates that although long-term forecasting

[1]Note that $\alpha_{\text{supp}} > 1$ allows the forecasted depth to be closer to the camera than the observed depth, accounting for potential noise in the depth estimator to reduce the number of forecasts that are suppressed.

of pedestrian trajectories may require higher-level reasoning [35, 42, 53], short-term occlusions may be modeled with simple linear models.

Next, we evaluate the same approach with detections from a Faster R-CNN [49] model in place of groundtruth (Table 4.2, row 2). This leads to a significant drop in both overall and occluded accuracy, indicating that improvements in *visible* person detection can improve detection for invisible people. Finally, although Occluded Top-5 F1 drops, it is significantly above chance, suggesting that current detectors equipped with appropriate trackers can detect invisible people.

4.5.2 What is the impact of *tracking* on occluded detection?

So far, we have assumed oracle linking of detections, allowing for linear interpolation of bounding boxes to detect people through occlusion. We now evaluate the impact of using an online tracker, equipped with re-identification, on detecting occluded people. Removing the oracle results in a drastic drop in accuracy: the Top-5 F1 score for occluded people drops by over 30 points (87.3 to 53.3, Table 4.2 row 3) using groundtruth detections, and 14 points with Faster R-CNN detections (46.4 to 32.2, Table 4.2 row 4). Despite this significant drop in Occluded Top-5 F1, the overall Top-5 F1 is significantly more stable (from 98.0 to 92.3 for groundtruth detections and 70.5 to 69.9 for Faster R-CNN), showing that *overall* person detection and tracking underemphasizes the importance of detecting occluded people.

4.5.3 Can online approaches work?

These results indicate that in the offline setting, existing visible-person detection and tracking approaches can detect invisible people via interpolation. We now evaluate a simple *online* approach, which uses an off-the-shelf visible person detector (Faster R-CNN), equipped with a tracker (DeepSORT) and linear (constant velocity) forecasting for detecting invisible people (Table 4.2, row 5). Moving to an online setting results in a similar Top-5 F1 score but significantly reduces the precision for occluded persons, from 60.8 to 29.5. This is expected as even though linear forecasting recalls slightly more number of boxes than offline interpolation (recall from 21.9 to 30.2), its naive

Table 4.3: Detection and tracking results on MOT-17 [43], MOT-20 [13] and PANDA [56] train. We evaluate on public detections provided with MOT-17 (DPM [18], FRCNN [49], SDP [63]), two trackers that operate on public detections (Tracktor++ [5], MIFT [25]), and CenterTrack [66] which does not use public detections. We use (public FRCNN, *visible* groundtruth) detections for (MOT-20, PANDA). Our method improves on occluded people across all trackers.

		Top-5 F1		Top-1 F1		IDF1	
		Occl	All	Occl	All	Occl	All
MOT-17	DPM	17.2	46.7	13.2	46.5	2.9	36.9
	+ Ours	24.6 (+7.4)	49.3 (+2.6)	17.4	48.4	7.2	36.8
	FRCNN	28.4	68.5	20.1	67.4	1.5	55.6
	+ Ours	39.8 (+11.4)	70.5 (+2.0)	26.7	68.5	10.5	54.8
	SDP	45.2	80.5	35.8	79.8	10.9	64.6
	+ Ours	51.2 (+6.0)	80.8 (+0.3)	38.5	79.4	17.0	64.7
	Tracktor++	32.4	77.0	22.7	76.8	1.3	65.1
	+ Ours	45.4 (+13.0)	77.2 (+0.2)	33.2	76.5	15.6	66.8
	MIFT	37.8	75.9	29.9	75.1	9.4	61.7
	+ Ours	44.9 (+7.1)	75.6 (-0.3)	33.8	74.3	16.5	62.6
	CTrack	38.7	84.8	29.4	84.2	5.4	65.0
	+ Ours	47.9 (+9.2)	84.4 (-0.4)	36.4	83.4	16.2	70.2
MOT-20	FRCNN	42.5	71.2	27.5	70.7	2.9	42.2
	+ Ours	46.1 (+3.6)	71.5 (+0.3)	28.6	70.9	5.0	42.0
PANDA	GT (visible)	45.5	90.6	30.5	90.5	2.5	70.2
	+ Ours	49.5 (+4.0)	90.5 (-0.1)	34.1	90.3	4.6	62.1

nature results in many more false positives resulting in a much lower precision and therefore, a similar F1 score. In Section 4.7, we present simple modifications to this approach that recover much of this performance gap.

4.6 Comparison to Prior Work

Next, we apply our approach to the output of existing methods to evaluate its improvement over prior work. Table 4.3 shows results on the MOT-17 train set, showing our approach improves significantly in Occluded Top-5 F1 ranging from 6.0

		Top-5 F1		Top-1 F1		IDF1	
		Occl	All	Occl	All	Occl	All
MOT-17	Ours	43.4	76.8	31.4	75.6	14.7	58.7
	MIFT [25]	38.4	77.3	29.7	76.7	10.4	56.4
	UnsupTrack [30]	35.9	78.1	26.6	77.4	9.7	62.6
	GNNMatch [45]	35.2	74.3	26.3	73.7	6.9	56.1
	GSM_Tracktor [41]	35.4	73.8	26.2	73.2	7.4	57.8
	Tracktor++ [5]	33.3	73.3	24.8	73.0	5.2	55.1
MOT-20	Ours	46.9	76.7	33.3	75.2	11.2	51.1
	Tracktor++ [5]	44.2	76.0	34.2	75.3	10.2	48.8
	UnsupTrack [30]	41.7	71.4	30.9	70.8	9.6	50.6
	SORT20 [59]	38.5	65.2	27.3	63.6	8.8	45.1

Table 4.4: Results on MOT-17 and MOT-20 test set. The best, second-best and third-best methods are highlighted.

to 13.0 points, while maintaining the overall F1. Detecting invisible people requires reliable amodal detectors for visible people (ref. Section 4.5). For this reason, we use *visible* groundtruth detections from PANDA, similar to the oracle experiments in Section 4.5, as no public set of amodal detections come with PANDA (unlike MOT-17 or MOT-20). Table 4.3 shows that our method improves the detection of occluded people by 4.0% on PANDA using groundtruth visible detections and by 3.6% on MOT-20 using the Faster-RCNN public detections. We explicitly do not tune our hyperparameters for these two datasets, showing that our method is robust to changes in video data distribution. MOT-20 and PANDA contain a few sequences with top-down views, where occlusions are rare. We disable our depth and occlusion reasoning on such sequences.

As MOT-17 and MOT-20 test labels are held out, we worked with the MOTChallenge authors to implement our metrics on the test server. Table 4.4 shows that MIFT[2][25] and Tracktor++ [5] achieve the highest Occluded Top-5 F1 amongst prior online approaches on MOT-17 and MOT-20 test respectively. Applying our approach on top of these methods improves results significantly by 5.0% to 43.4 F1 and by 2.7% to 46.9 F1, leading to a new state-of-the-art for occluded person detection on

[2]MIFT is referred to as ISE_MOT17R on the MOT leaderboards

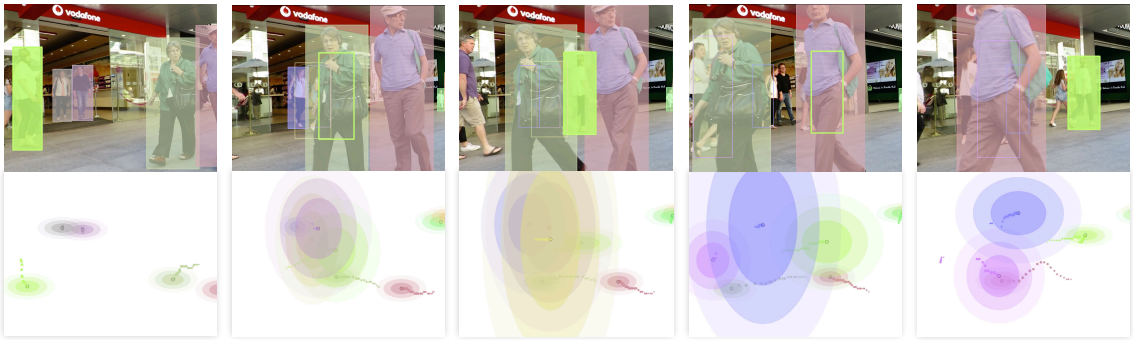

Figure 4.3: Our probabilistic model reports a *distribution* over 3D location during occlusions. We visualize (occluded, visible) detection with (outlined, filled-in) bounding boxes (**top**). We provide "birds-eye-view" top-down visualizations of Gaussian distributions over 3D object centroids with covariance ellipses (**bottom**). During occlusion, variance grows roughly linearly with the number of consecutively-occluded frames. We are also able to correctly predict depth of occluded people in the top down view, e.g. in the second last frame, which would not be possible with single-frame monocular depth estimates. During evaluation, we truncate the uncertainty using our freespace estimates (not visualized).

MOT-17 and MOT-20 test.

Table 4.3 shows that our method consistently improves occluded F1. However, it sometimes results in a drop in overall accuracy. We attribute this to the increased number of false positives introduced while tackling the challenging task of detecting invisible people. These false positives for invisible people are counted as false positives for *all* people, whether visible or invisible. This causes existing metrics to penalize methods for even *trying* to detect invisible people. In safety critical applications, where worst-case accuracy may be more appropriate, our approach significantly improves during complete occlusions by up to 13.0% on MOT-17, while mildly decreasing average accuracy by 0.4%.

4.7 Ablation Study

We now study the impact of each component of our approach in Table 4.5, focusing on the Occluded Top-5 F1 metric using Faster R-CNN detections on the MOT-17 train set. First, we show that the DeepSORT tracker, upon which our approach is built,

results in a 28.4 Occluded Top-5 F1. Reporting the internal, linear forecasts from the tracker increases the score to 29.8, driven primarily by a 12.5% improvement in recall. Compensating for camera motion provides another 2.4% improvement. Next, leveraging depth cues to incorporate freespace constraints, as detailed in Section 3.3, improves accuracy by 3.5%, driven primarily by a 14.6% jump in precision, indicating that this component drastically reduces false positives. Finally, we add depth-aware process noise to handle perspective transformations between 2D and 3D coordinates, which leads to an improvement of 4.1%, resulting in a final score of 39.8. Only a 1.0% improvement in F1 as compared to 4.1% with Top-5 F1 suggests that our uncertainty estimates are significantly improved by the depth-aware process noise scaling. In all, our approach leads to an improvement of 11.4% over the baseline. Figure 4.3 presents a sample result from our approach, where the person in the green bounding box is detected throughout two full occlusion phases, marked with an unfilled box. All of our qualitative analysis, including 3D visualization of a scene from a 2D monocular video, is available online at https://youtu.be/StEfnshXrCE.

One concern with our approach might be that the average depth inside a person's bounding box may contain pixels from the background or an occluder. To verify the impact of this, we evaluate a variant where we use segmentation masks for all the bounding boxes in MOT-17's FRCNN public detections using MaskRCNN [24]. We initialize the z state variable in the model with the average depth inside this mask. On doing so, the Top-1 occluded F1 increases from 26.7 to 27.3, indicating that masks can help with estimating the person's depth, but boxes are a reasonable approximation.

4.7.1 Forecasting

We evaluate replacing our linear forecaster with state-of-the-art forecasters. We supply these forecasters with a birds-eye-view representation of visible person trajectories. As these forecasters forecast only the birds-eye-view (x, z) coordinates, we rely on our approach's estimates of the height, width, and y coordinate. We evaluate two trajectory forecasting approaches for crowded scenes, Social GAN (SGAN) [23] and STGAT [27]. SGAN and STGAT result in Occluded Top-5 F1 scores of 36.0 and 36.4 respectively. While this improves over the baseline at 28.4, it underperforms our

Table 4.5: MOT-17 train ablations. Each row adds a component to the row above. 'Dep. noise' is depth-aware noise.

	Top-5				Top-1 F1	
	Occl F1	Occl Prec	Occl Rec	All F1	Occl	All
DeepSORT	28.4 ±0.1	71.9 ±0.2	17.7 ±0.1	68.5 ±0.0	20.1	67.4
+ Forecast	29.8 ±0.2	29.5 ±0.4	30.2 ±0.1	69.4 ±0.0	20.9	66.5
+ Egomotion	32.2 ±0.2	33.1 ±0.3	31.3 ±0.1	70.4 ±0.0	23.2	67.9
+ Freespace	35.7 ±0.0	47.7 ±0.1	28.6 ±0.0	70.4 ±0.0	25.7	68.4
+ Dep. noise	39.8 ±0.2	52.6 ±0.6	32.0 ±0.0	70.5 ±0.1	26.7	68.5

linear forecaster at 39.8. This suggests that simple linear models suffice for short, frequent occlusions.

As described above, we use a constant velocity forecaster in our probabilistic approach. 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, SGAN [23] and STGAT [27] 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 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. We report the best of these models for both approaches and report other models in Table 4.6. For STGAT, the 8- and 12-timestep models used are trained on the ETH [46] dataset and for SGAN, the 8- and 12-timestep models are trained on the ZARA1 [36] 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

		Top-5 F1		Top-1 F1		IDF1	
		Occl	All	Occl	All	Occl	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 4.6: MOT-17 train forecasting ablations with state-of-the-art social forecasting models.

replace the x and z values from the output of our pipeline with SGAN and STGAT's forecasted x and z values.

In Table 4.6, 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 previously. 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.

4.7.2 Monocular Depth Estimators

Our method relies on an off-the-shelf monocular depth estimator to enable occlusion reasoning in 3D. In general, we used the MegaDepth [38] 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 [39] and MIDAS [34] 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

	Top-5 F1		Top-1 F1		IDF1	
Depth est.	Occl	All	Occl	All	Occl	All
MegaDepth [38]	35.4±0.2	69.8±0.0	26.7	68.4	9.5	53.3
Mannequin [39]	34.2±0.2	69.4±0.0	25.5	68.0	8.5	53.3
MIDAS [34]	34.4±0.1	69.5±0.0	26.5	68.2	9.1	53.8

Table 4.7: Comparison of different 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.

of our pipeline allows us to evaluate the quality of depth estimates from each of the three methods. Table 4.7 shows that the per frame depth estimator from Mannequin Challenge [39] does worse than MegaDepth [38] by 1.2 Top-5 F1 for invisible people and MIDAS [34] 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.

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 [34] estimator in Table 4.8. 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 [34] 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 [38, 39] whose performance decreases or stagnates with higher resolutions (not shown).

		Top-5 F1		Top-1 F1		IDF1	
Depth	Res.	Occl	All	Occl	All	Occl	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 4.8: We evaluate a recent depth estimator, MIDAS [34], 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.

4.7.3 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 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 [24]. 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 4.9, 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.

4.7.4 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 4.10 shows that compensating

	Top-5 F1		Top-1 F1		IDF1	
	Occl	All	Occl	All	Occl	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 4.9: Replacing boxes by masks for getting mean depth of a person only helps by a small amount suggesting that boxes can reasonably replace masks.

	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 4.10: MOT-17 train ablations for moving stationary camera sequences.

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

4.8 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

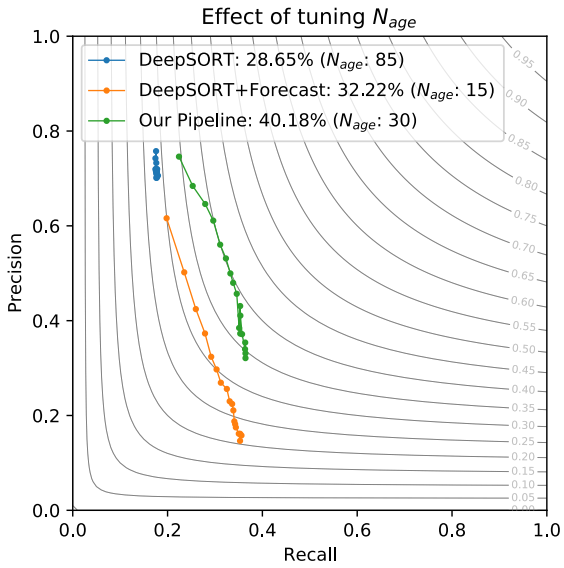

Figure 4.4: Detecting occluded people is sensitive to the threshold used to declare a detection-under-high-occlusion. We fix the number of N_{age} frames that a track is allowed to be in an occluded state. By increasing N_{age}, we can tradeoff precision and recall in invisible-people-detection which results in a "PR-curvelet". The curvelets represent the experiments in rows 1, 2 and 5 of ablation experiments table.

number of frames that an occluded track is forecasted for before it is deleted. We show in Figure 4.4 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.

We use a hyperparameter $f_{process}$ to scale the process noise covariance. We additionally scale the observation noise covariance by $f_{observation}$ to account for the removal of default scaling by height of [59]. In our algorithm, we use $f_{process} = 900$

and $f_{observation} = 600$.

4.9 IDF1-Occluded & MOTA-Occluded

We previously reported detection results using the probabilistic and standard F1 metrics. Here, we supplement these results with the IDF1 and MOTA (Multi-Object Tracking Accuracy) tracking metrics [6]. To do this, we follow the strategy: We do not penalize tracks that match to visible people, but we reward only tracks that match to occluded people.

4.9.1 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 4.2, 4.3, 4.4, 4.11, 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.

4.9.2 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.

	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 4.11: Analysis of IDF1- and MOTA-occluded for the MOT-17 train ablation experiments. 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).

Perhaps surprisingly, we find in Table 4.11 that the MOTA metric is negative for all ablations. To better understand this, we note that MOTA is a simple combination of TP, FP, and identity switches (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 4.11. 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.

Chapter 5

Discussion

We propose the task of detecting fully-occluded objects from uncalibrated monocular cameras in an online manner. Our experiments show that current detection and tracking approaches struggle to find occluded people, dropping in accuracy from 68% to 28% F1. Our oracle experiments reveal that interpolating across tracklets in an offline setting noticeably improves F1, but the task remains difficult because underlying object detectors do not perform well during large occlusions. We propose an online approach that forecasts the trajectories of occluded people, exploiting depth estimates from a monocular depth estimator to better reason about potential occlusions. Our approach can be applied to the output of existing detectors and trackers, leading to significant accuracy gains of 11% over the baseline, and 5% over state-of-the-art. We hope our problem definition and initial exploration of this safety-critical task encourages others to do so as well.

Chapter 6

Future Directions

6.1 Limitations

Our approach assumes a constant velocity assumption and is thus based on linear dynamics. It can therefore not model non-linear motion of objects especially when such a motion starts at the start of the occlusion phase. We show an example of this failure case and others, such as those resulting from errors in groundtruth, in our qualitative anaylsis video on YouTube.

It should be noted that the depth-based reasoning, in the current approach, is only used as a post-processing step after the short-term forecasting. A better approach would tie and integrate the two together so that depth can guide forecasting and vice-versa. One way to do this would be to reweight the covariances in the Kalman Filter with the truncated gaussians obtained after freespace horizon truncation, resulting in a Particle Filter like approach. Forecasting and depth-reasoning could also be tied together with explicit joint optimization.

Other than this, it might be useful to embrace the existence of metric depth sensors such as LiDARs and modify the approach to operate in 3D world coordinates rather than in the projective space with relative monocular depth maps.

Finally, our approach is based on classical tracking approaches like a smoothing Kalman Filter. We point out that more recent learning based approaches like Tracktor++ and CenterTrack could be explored for incorporating depth-based reasoning for multi-object tracking.

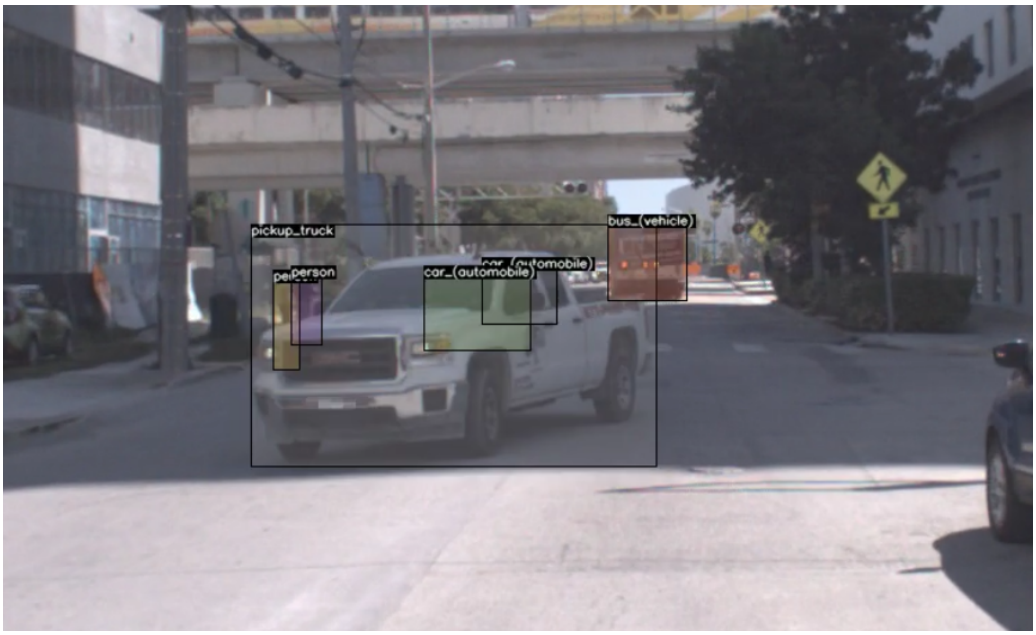

Figure 6.1: TAO-Amodal, an extension of TAO [12], is expected to be the largest in-the-wild amodal object detection dataset that will label objects to their full extent in both in-frame and out-of-frame complete and partial occlusions.

Next, I discuss two major components our pipeline is dependent on and the issues that exist in both the fields.

6.2 Amodal Object Detection

First, our approach is dependent on good *amodal* detectors. In this regard, we explore two directions of future work. First, we note that no video dataset exists that focuses explicitly on amodal object annotations across their tracks, although this is a more reasonable object detection task as objects do not cease to exist where their visual footprint ends. We propose TAO-Amodal, an extension of our older work, TAO [12].

With TAO-Amodal, we hope to label arbitrary objects to their full-extent even in complete occlusions that occur in-frame or out-of-frame. We are currently running a pilot on all the 7 dataset subsets in TAO. Till now, about 30,000 amodal bounding boxes have been labelled, out of which 6% of the boxes correspond to fully-visible objects, 87% of the boxes correspond to partially-visible objects and 7% of the boxes

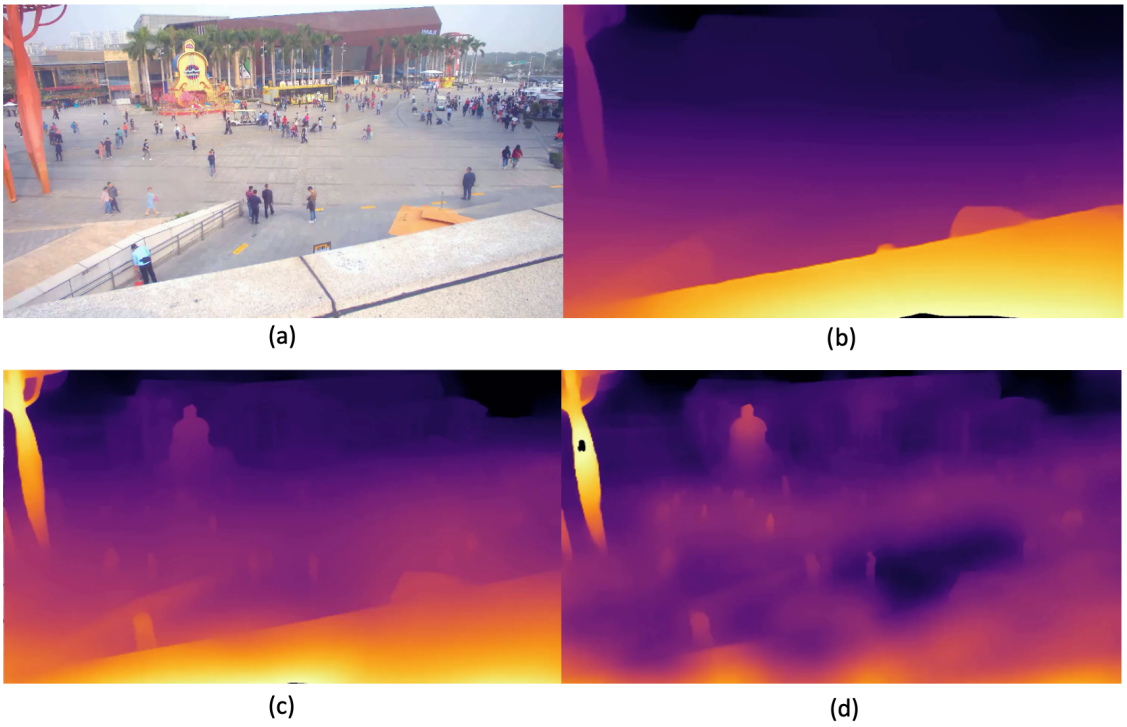

Figure 6.2: Inference at a low-resolution of 384 x 384 from MIDAS [34] results in (b) loss of all high-frequency details in the image, people in this case. When the input resolution is increased by (c) 2x and (d) 4x, most of these details start appearing while the overall geometry of the scene is harmed.

are new boxes (which did not exist in TAO before), corresponding to fully-occluded objects. The latter results in a total of about 2,000 boxes for fully-occluded objects and 93% of the time, our annotators were confident of their labelling location for invisible objects after two rounds of quality control.

TAO-Amodal is expected to be the largest in-the-wild amodal object detection dataset available for public use. A sample frame with amodal annotations is shown in Figure 6.1.

6.3 High-resolution Monocular Depth Estimation

Second, our approach is also dependent on plausible depth estimates. We note that a seemingly harmful protocol in depth estimation inference is to downsample input.

This leads to loss of high-frequency details in images, and important objects like people and vehicles in high-resolution images. An example of this is shown in Figure 6.2. When the input resolution is naively increased, the high-frequency details start to appear but the overall geometry is harmed.

I explored a few test-time hypotheses in this direction, such as the quality of depth being a function of the input resolution proportional to the depth value, but the conclusion was that one must think more broadly about how to correctly design depth estimators that are independent of input resolution; this may either require sending in information about camera intrinsics, or require an architectural shift to representing images implicitly as a set of rays originating from a camera center.

Bibliography

[1] Vitaly Ablavsky and Stan Sclaroff. Layered graphical models for tracking partially occluded objects. *TPAMI*, 33(9):1758–1775, 2011. 2.2

[2] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *CVPR*, 2016. 2.3

[3] Renée Baillargeon and Julie DeVos. Object permanence in young infants: Further evidence. *Child development*, 62(6):1227–1246, 1991. 1

[4] Jerome Berclaz, Francois Fleuret, Engin Turetken, and Pascal Fua. Multiple object tracking using k-shortest paths optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(9):1806–1819, 2011. 2.2

[5] Philipp Bergmann, Tim Meinhardt, and Laura Leal-Taixe. Tracking without bells and whistles. In *ICCV*, 2019. (document), 2.2, 3.3, 4.3, **??**, **??**, 4.6

[6] 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. 4.9

[7] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *ICIP*, 2016. doi: 10.1109/ICIP.2016.7533003. 2.2, 3.3

[8] Ted J Broida, S Chandrashekhar, and Rama Chellappa. Recursive 3-d motion estimation from a monocular image sequence. *IEEE Transactions on Aerospace and Electronic Systems*, 26(4):639–656, 1990. 1.3

[9] Michael Chan, Dimitri Metaxas, and Sven Dickinson. Physics-based tracking of 3d objects in 2d image sequences. In *Proceedings of 12th International Conference on Pattern Recognition*, volume 1, pages 432–436. IEEE, 1994. 2.2

[10] Ming-Fang Chang, John Lambert, Patsorn Sangkloy, Jagjeet Singh, Slawomir Bak, Andrew Hartnett, De Wang, Peter Carr, Simon Lucey, Deva Ramanan, et al. Argoverse: 3d tracking and forecasting with rich maps. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages

8748–8757, 2019. (document), 1.1, 1.1

[11] Javier Civera, Andrew J Davison, and JM Martinez Montiel. Inverse depth parametrization for monocular slam. *IEEE transactions on robotics*, 24(5): 932–945, 2008. 1.3

[12] Achal Dave, Tarasha Khurana, Pavel Tokmakov, Cordelia Schmid, and Deva Ramanan. Tao: A large-scale benchmark for tracking any object. In *European conference on computer vision*, pages 436–454. Springer, 2020. (document), 6.1, 6.2

[13] 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. (document), 1.4, 4.2, 4.2.1, 4.3

[14] Kiana Ehsani, Roozbeh Mottaghi, and Ali Farhadi. Segan: Segmenting and generating the invisible. In *CVPR*, 2018. 2.1

[15] Andreas Ess, Konrad Schindler, Bastian Leibe, and Luc Van Gool. Improved multi-person tracking with active occlusion handling. In *ICRA Workshop on People Detection and Tracking*, 2009. 1.3, 3.3

[16] Georgios D Evangelidis and Emmanouil Z Psarakis. Parametric image alignment using enhanced correlation coefficient maximization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(10):1858–1865, 2008. 3.3

[17] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010. 1, 4.3.1

[18] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645, 2009. (document), 4.3

[19] Francois Fleuret, Jerome Berclaz, Richard Lengagne, and Pascal Fua. Multicamera people tracking with a probabilistic occupancy map. *IEEE transactions on pattern analysis and machine intelligence*, 30(2):267–282, 2007. 2.2

[20] Shan Gao, Zhenjun Han, Ce Li, Qixiang Ye, and Jianbin Jiao. Real-time multipedestrian tracking in traffic scenes via an rgb-d-based layered graph model. *IEEE Transactions on Intelligent Transportation Systems*, 16(5):2814–2825, 2015. 2.2

[21] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32 (11):1231–1237, 2013. 2.1

[22] Helmut Grabner, Jiri Matas, Luc Van Gool, and Philippe Cattin. Tracking the invisible: Learning where the object might be. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1285–1292. IEEE, 2010. 2.2

[23] 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. 4.7.1

[24] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *ICCV*, 2017. 2.1, 4.7, 4.7.3

[25] 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. (document), 4.3, **??**, 4.6

[26] Yan Huang and Irfan Essa. Tracking multiple objects through occlusions. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 1051–1058. IEEE, 2005. 1

[27] 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. 4.7.1

[28] Michael Isard and John MacCormick. Bramble: A bayesian multiple-blob tracker. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, volume 2, pages 34–41. IEEE, 2001. 2.2

[29] Omid Hosseini Jafari, Dennis Mitzel, and Bastian Leibe. Real-time rgb-d based people detection and tracking for mobile robots and head-worn cameras. In *ICRA*, 2014. 2.2

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

[31] Saad M Khan and Mubarak Shah. A multiview approach to tracking people in crowded scenes using a planar homography constraint. In *European Conference on Computer Vision*, pages 133–146. Springer, 2006. 2.2

[32] Kyungnam Kim and Larry S Davis. Multi-camera tracking and segmentation of occluded people on ground plane using search-guided particle filtering. In *European Conference on Computer Vision*, pages 98–109. Springer, 2006. 2.2

[33] Kris M Kitani, Brian D Ziebart, James Andrew Bagnell, and Martial Hebert. Activity forecasting. In *ECCV*. Springer, 2012. 2.3

[34] 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. (document), 3.2, 4.7.2, **??**, 4.8, 6.2

[35] Laura Leal-Taixé, Gerard Pons-Moll, and Bodo Rosenhahn. Everybody needs somebody: Modeling social and grouping behavior on a linear programming multiple people tracker. In *2011 IEEE international conference on computer vision workshops (ICCV workshops)*, pages 120–127. IEEE, 2011. 2.3, 4.5.1

[36] Alon Lerner, Yiorgos Chrysanthou, and Dani Lischinski. Crowds by example. In *Computer graphics forum*, volume 26, pages 655–664. Wiley Online Library, 2007. 4.7.1

[37] Ke Li and Jitendra Malik. Amodal instance segmentation. In *ECCV*. Springer, 2016. 2.1

[38] 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. 3.2, 4.4, 4.7.2, **??**

[39] 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. 4.7.2, **??**

[40] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 1, 4.3.1

[41] 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. **??**

[42] Wei-Chiu Ma, De-An Huang, Namhoon Lee, and Kris M Kitani. Forecasting interactive dynamics of pedestrians with fictitious play. In *CVPR*, 2017. 2.3, 4.5.1

[43] 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. (document), 1, 1.4, 4.1, 4.2, 4.3

[44] Masatoshi Okutomi and Takeo Kanade. A multiple-baseline stereo. In *CVPR*, volume 93, pages 63–69, 1991. 3.2

[45] 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. ??

[46] 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. 2.3, 4.7.1

[47] Hamed Pirsiavash, Deva Ramanan, and Charless C Fowlkes. Globally-optimal greedy algorithms for tracking a variable number of objects. In *CVPR*, 2011. 2.2

[48] Lu Qi, Li Jiang, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Amodal instance segmentation with KINS dataset. In *CVPR*, 2019. 2.1

[49] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. (document), 4.4, 4.5.1, 4.3

[50] John W Roach and JK Aggarwal. Determining the movement of objects from a sequence of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (6):554–562, 1980. 1.3

[51] Alexandre Robicquet, Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. Learning social etiquette: Human trajectory understanding in crowded scenes. In *ECCV*. Springer, 2016. 2.3

[52] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015. 1

[53] Paul Scovanner and Marshall F Tappen. Learning pedestrian dynamics from the real world. In *ICCV*. IEEE, 2009. 2.3, 4.5.1

[54] Davide Spinello and Daniel J Stilwell. Nonlinear estimation with state-dependent gaussian observation noise. *IEEE Transactions on Automatic Control*, 55(6): 1358–1366, 2010. 1.3

[55] Chris Urmson, Joshua Anhalt, Drew Bagnell, Christopher Baker, Robert Bittner, MN Clark, John Dolan, Dave Duggins, Tugrul Galatali, Chris Geyer, et al. Autonomous driving in urban environments: Boss and the urban challenge. *Journal of Field Robotics*, 25(8):425–466, 2008. 1.3

[56] 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. (document), 1.4, 4.2, 4.2.1, 4.3

[57] Yan Wang, Wei-Lun Chao, Divyansh Garg, Bharath Hariharan, Mark Campbell, and Kilian Q Weinberger. Pseudo-lidar from visual depth estimation: Bridging

the gap in 3d object detection for autonomous driving. In *CVPR*, 2019. 1.3

[58] Nicolai Wojke and Alex Bewley. Deep cosine metric learning for person re-identification. In *WACV*. IEEE, 2018. doi: 10.1109/WACV.2018.00087. 2.2, 3.3

[59] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *ICIP*, 2017. 3, 3.1, 3.3, **??**, 4.8

[60] Ning Xu, Linjie Yang, Yuchen Fan, Dingcheng Yue, Yuchen Liang, Jianchao Yang, and Thomas Huang. Youtube-vos: A large-scale video object segmentation benchmark. *arXiv preprint arXiv:1809.03327*, 2018. 1

[61] Kota Yamaguchi, Alexander C Berg, Luis E Ortiz, and Tamara L Berg. Who are you with and where are you going? In *CVPR 2011*. IEEE, 2011. 2.3

[62] Xiaosheng Yan, Feigege Wang, Wenxi Liu, Yuanlong Yu, Shengfeng He, and Jia Pan. Visualizing the invisible: Occluded vehicle segmentation and recovery. In *ICCV*, 2019. 2.1

[63] Fan Yang, Wongun Choi, and Yuanqing Lin. Exploit all the layers: Fast and accurate cnn object detector with scale dependent pooling and cascaded rejection classifiers. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2129–2137, 2016. (document), 4.3

[64] Qian Yu, Gérard Medioni, and Isaac Cohen. Multiple target tracking using spatio-temporal markov chain monte carlo data association. In *ICCV*, 2007. 2.2

[65] Ziheng Zhang, Anpei Chen, Ling Xie, Jingyi Yu, and Shenghua Gao. Learning semantics-aware distance map with semantics layering network for amodal instance segmentation. In *ACM Multimedia*, 2019. 2.1

[66] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking objects as points. *arXiv:2004.01177*, 2020. (document), 4.3

[67] Yan Zhu, Yuandong Tian, Dimitris Mexatas, and Piotr Dollár. Semantic amodal segmentation. *arXiv preprint arXiv:1509.01329*, 2015. 2.1

[68] Yan Zhu, Yuandong Tian, Dimitris Metaxas, and Piotr Dollár. Semantic amodal segmentation. In *CVPR*, 2017. 2.1