2013 IEEE Conference on Computer Vision and Pattern Recognition

Tracking People and Their Objects

Tobias Baumgartner * Dennis Mitzel * Bastian Leibe
Computer Vision Group, RWTH Aachen University
tobias.baumgartner@rwth-aachen.de, {**mitzel,leibe**}@vision.rwth-aachen.de

Abstract

Current pedestrian tracking approaches ignore important aspects of human behavior. Humans are not moving independently, but they closely interact with their environment, which includes not only other persons, but also different scene objects. Typical everyday scenarios include people moving in groups, pushing child strollers, or pulling luggage. In this paper, we propose a probabilistic approach for classifying such person-object interactions, associating objects to persons, and predicting how the interaction will most likely continue. Our approach relies on stereo depth information in order to track all scene objects in 3D, while simultaneously building up their 3D shape models. These models and their relative spatial arrangement are then fed into a probabilistic graphical model which jointly infers pairwise interactions and object classes. The inferred interactions can then be used to support tracking by recovering lost object tracks. We evaluate our approach on a novel dataset containing more than 15,000 frames of person-object interactions in 325 video sequences and demonstrate good performance in challenging real-world scenarios.

1. Introduction

Considerable progress has been made in the development of dynamic scene understanding approaches over the last few years [1, 2, 3, 6, 10, 12, 19]. Still, most current approaches are so far limited to recognizing and tracking a small number of known object categories, such as pedestrians or cars. Recently, tracking approaches have been extended by social walking models [15] and by modeling of group behavior [4, 11, 16, 20]. However, another major factor that influences peoples' behavior and dynamics—their interactions with scene objects—has so far been underrepresented. Such interactions are harder to incorporate, since their analysis requires recognizing the presence of objects whose shape and appearance may as yet be unknown. Consequently, person-object interactions have so far mostly been considered in surveillance settings with fixed cameras (*e.g.*, [5, 17]), where background modeling can be used to segment and track unknown objects.

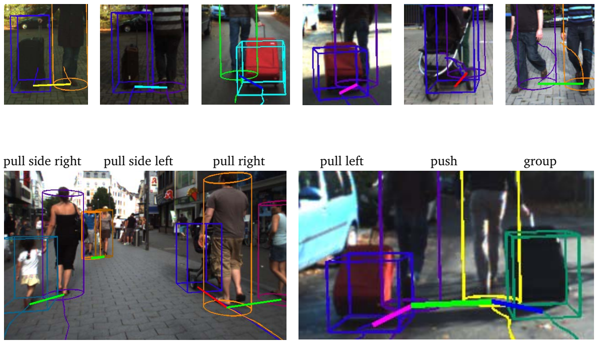


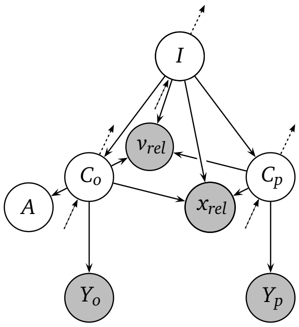
Figure 1: Our proposed approach models pairwise interactions between persons and objects in a probabilistic graphical model, taking into account object shape, relative arrangement, and temporal consistency. Thus, it can infer which objects belong to which persons and predict how the interactions will continue. Recognized interactions are visualized by colored lines linking the foot points of interacting objects (Legend: **pull side right**, **pull side left**, **pull right**, **pull left**, **push**, **group**).

With this paper we present a mobile scene understanding approach for inner-city shopping areas, airports, or train stations. In such scenarios, people often handle luggage items, child strollers, trolleys, etc. Current *tracking-by-detection* approaches cannot track such objects, since (a) there are no generic detectors available for all dynamic objects, and (b) *tracking-by-detection* does not scale to a large number of detector classes. Our approach can track persons and other scene objects from a mobile platform and jointly infer both the object class and the interaction type from observed appearances and dynamics. The core component of our approach is a probabilistic graphical model that relates object appearance and spatial arrangement consistently over time. This model can determine which persons and objects belong together and in what way they interact. Based on the recognized interaction, it can then predict how the interaction will most likely continue and how one object's trajectory will be affected by another object's observed motion.

Realizing such an approach for a mobile platform cannot be done in a standard *tracking-by-detection* framework

[*] Both authors contributed equally to this work.

1063-6919/13 $26.00 © 2013 IEEE
DOI 10.1109/CVPR.2013.469



I	Interaction Type $\in$ {push, group, none , . . .}.
C_o	Object Type $\in$ {person, stroller, 2-wheel bag, . . .}.
C_p	Person $\in$ {Y/N}.
x_{rel}	Relative position.
v_{rel}	Relative velocity.
A	Autonomous $\in$ {Y/N}.
Y_o	Object appearance.
Y_p	Person appearance.

Figure 2: (left) Bayesian Network for object person interaction, dashed lines indicate inference from preceding and to subsequent frames. (right) table of variables in Bayesian Network.

based on pre-trained object detectors [1, 2, 3, 6, 12, 19], since object class inference will only become possible after an object configuration has already been tracked for several frames. We therefore formulate our approach in a *tracking-before-detection* framework based on low-level stereo region segmentation and multi-hypothesis data association.

The benefit of this approach is that it enables us to track a large variability of objects with potentially unknown appearance, while achieving increased robustness to classification failures. For an example, consider the scene shown in Fig. 1. Our approach fails to recognize the child in the bottom left corner of the figure as a *person* (visualized by a cylinder). In a tracking-by-detection approach, this would cause a tracking failure. Instead, our approach treats the child as an *unknown moving object* (visualized by a box) and it can still recognize that this object forms a group with the child's mother (shown by the green connecting line), thus affecting the mother's trajectory.

In detail, our paper makes the following contributions: (1) We propose a probabilistic graphical model for recognizing pairwise person-object interactions taking into account object shape, relative arrangement, and temporal consistency. This model can jointly infer object classes and interaction patterns more robustly than could be done from individual observations. In particular, it can resolve which object belongs to which person, arriving at improved scene understanding. (2) This scene interpretation allows our approach to make improved predictions for the continuation of each tracked object's trajectory with increased robustness to occlusions and detection failures. (3) In order to make this approach feasible on noisy stereo depth data, we propose several detailed contributions spanning the entire tracking pipeline. This includes novel methods for improved region candidate extraction, data association, and multi-hypothesis discrimination. (4) We introduce a novel benchmark dataset for person-object interaction consisting of 325 video sequences with a total of almost 15,000 frames and use it to quantitatively evaluate our approach's performance.

The paper is structured as follows. The following section discusses related work. After that, Sec. 2 presents the proposed graphical model for object and interaction classification. Sec. 3 discusses how model parameters are learned, and Sec. 4 shows how the model is used for inference and prediction. Sec. 5 integrates the model into a tracking pipeline for robust scene interpretation. Finally, Sec. 6 presents experimental results.

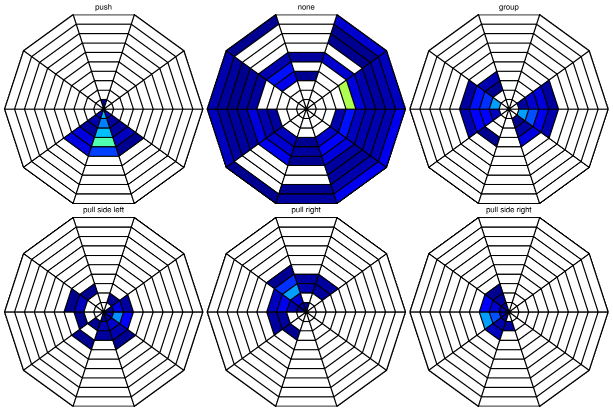


Figure 3: Learned conditional distributions for relative positions in a log-polar grid.

Related Work. Tracking dynamic objects reliably is an important part of scene understanding. In recent years, a number of tracking-by-detection approaches have been proposed for this task [2, 3, 6, 10, 12, 19], achieving good performance. However, most such approaches are restricted to pre-trained classifiers that yield the detections and ignore the impact on individual pedestrian motion by other nearby scene objects.

Incorporating social walking models into modeling the dynamics of individual pedestrians [15, 20] and groups [4, 11, 16] has been shown to yield significant improvement for tracking in crowded scenes. Similarly, [4] have shown that tracking results can be improved by simultaneously tracking multiple people and estimating their collective activities. However, those approaches consider only other pedestrians as possible scene objects and ignore the impact of a large variety of other objects such as bicycles, child strollers, shopping carts, or wheelchairs often present in street scenes. A main reason for this is the lack of reliable classifiers spanning the large variety of scene object classes.

There are several approaches that model person-object interactions in static surveillance camera footage using background modeling. For example, [17] propose to detect abandoned luggage items by analyzing the size and velocity of tracked foreground blobs. [5] propose a more elaborate approach for carried item detection that compares the segmented object area to learned temporal templates of pedestrian shapes. Such approaches are limited by the requirement of background modeling, which makes them not applicable for our scenarios with a moving camera.

Recently, [13] has proposed a tracking-before-detection approach that can track both known and unknown object

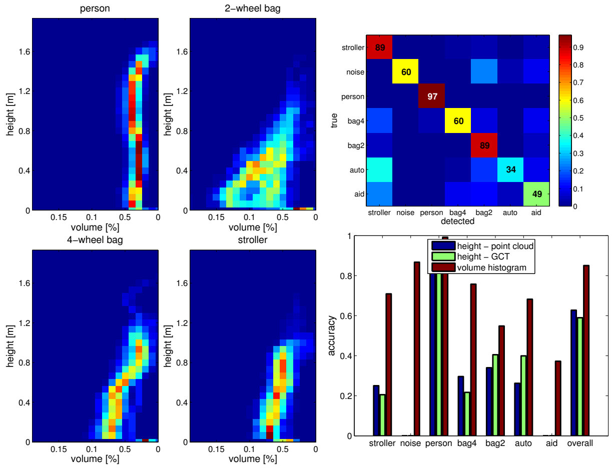


Figure 4: (left) Learned GCT histogram classifiers for person, 2/4-wheel bag and stroller. (right) Performance of classifier.

categories from a mobile platform based on stereo data. Their method relies on stereo region-of-interest (ROI) extraction to extract possible object candidates [2, 3, 9, 13] and to track them over time. We take inspiration from this approach in order to develop our model, but significantly extend it with improved methods for candidate object segmentation, data association, and object interaction handling.

2. Modeling Person-Object Interactions

We model all person-object interactions in the scene in a pairwise manner. This has two important implications: On the one hand, we assume each observable interaction to have exactly two actors. On the other hand, our model becomes easy to handle and learn, and inference can be performed in an efficient way. We try to robustly explain what is happening in the scene under the basic assumption that persons' actions will be the dominant cause of observable object motion, meaning that an object can only move because of a person's impact. Having analyzed a scene and interpreted all interactions, our model can then use this information in a generative way in order to predict future motion and support tracking.

Looking at a scene of various given objects, their past trajectories and current positions, we derive a number of individual and pairwise features to infer the type of interaction. Firstly, we model the appearance of objects and persons and try to assign them to one of the classes: *stroller*, *2-wheel bag*, *4-wheel bag*, *walking aid*, *person*, *autonomous* (*e.g.*, electric wheelchairs), and *none*. For each person-object and person-person pair, we can determine their relative positions in the scene, as well as their relative velocities derived from their trajectories. Together with the object appearances, we use those as features in order to infer the interaction type. In this paper, we consider 6 different interaction classes, as shown in Fig. 1(top), plus the additional class *none*, indicating independence. In our setting of pairwise interaction, the action *group* is defined as true if and only if two persons belong to the same group of people. An intuitive notion of group transitivity will then allow us to robustly identify all persons belonging to the same group.

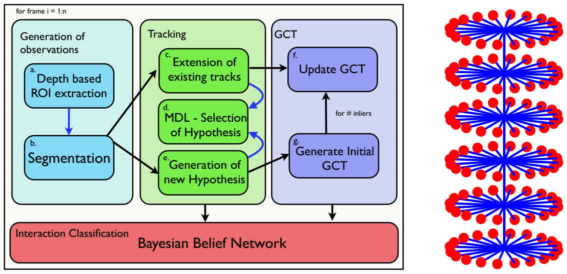

for frame i = 1:n
Generation of observations
a. Depth based ROI extraction
b. Segmentation
Tracking
c. Extension of existing tracks
d. MDL - Selection of Hypothesis
e. Generation of new Hypothesis
GCT
f. Update GCT
for # inliers
g. Generate Initial GCT
Interaction Classification
Bayesian Belief Network

Figure 5: (1) Overview of observation generation for the proposed Bayesian Network. (2) GCT shape representation, which accumulates observed 3D points of the object in distance histograms [13].

In a scene with 3 entities we consider a total of 6 possible interactions, *i.e.*, each pair of entities twice, with either entity as the dominant actor, denoted by "actor". Since we do not know the entity class a priori, we determine for each interaction a probability for the actor to be a person. If this probability is very low, we can tell immediately that we have an instance of the action "none". We also model whether an object acts in an *autonomous* way (as another pedestrian, or electric wheelchair would do).

Fig. 2 illustrates our proposed model and an overview of the used random variables. Using this model, the likelihood of an observed interaction can be decomposed as:

$$p(I, C_o, C_p, v_{rel}, x_{rel}, A, Y_o, Y_p) =$$
$$p(I) \cdot p(C_o|I) \cdot p(C_p|I) \cdot p(x_{rel}|I, C_o, C_p) \cdot$$
$$p(v_{rel}|I, C_o, C_p) \cdot p(Y_o|C_o) \cdot p(Y_p|C_p) \cdot p(A|C_o)$$

Except for $p(Y_*|C_*)$ all of these factors are multinomial distributions learned from frequencies in the training data, as described in Sec. 6. The two conditionals $p(Y_*|C_*)$ will be computed using a new classifier, described in Sec. 3.

At runtime we will then observe the appearances of our actors Y_o and Y_p, as well as their relative positions and velocities, x_{rel} and v_{rel}, respectively (*c.f.* shaded nodes in Fig. 2). To infer an interaction between these two, as well as the object type and person classification, we perform exact Belief Propagation using the junction tree algorithm [14].

The object-type classifier assumes a correct tracking and the input of a 3D point cloud that only contains points belonging to the person to be classified. Later in Sec. 5, we show how to construct these stable inputs from noisy data.

3. Learning

Relative Position and Velocity. We define all relative measures in a log-polar coordinate system. Fig. 3 shows the learned relative positions in our model for 10 bins for each angle and log distances. The intuition for these grids

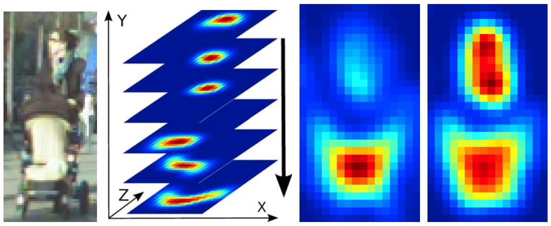


Figure 6: Visualization of the proposed segmentation procedure: (1) image cutout, woman with a stroller, (2) visualization of height layers, (3) ground projection using approach [2], (4) ground projection using our approach.

is that a person is located in the center of the *spider web* facing downwards. An object on her left will hence be represented by the bin to the right of the middle point. Unsurprisingly, these probability distributions correctly reflect spatial arrangements. For example, one would always expect a stroller that is pushed by a person to be located in front of her (*c.f.* Fig. 3(top left)).

Object Classifiers. We use and evaluate two different methods for object classification. The first is based on a simple object height measure. From our training data we learn a multinomial height distribution for the different object classes and use this to predict the class given the observed height. Since we always assume noisy 3D data, the height is smoothed over subsequent frames before classification.

For the second classifier, we use a more complex object shape model based on the volumetric GCT representation from [13]. We determine a volume distribution for each learned object, as described in Sec. 5, and classify by computing per-class posteriors based on the observed volumes. Fig. 4 shows the learned models for person, stroller, 2- and 4-wheel bag.

Given the volume histogram $\mathbf{x}$ for a GCT (*e.g.* Fig. 7), we evaluate the class posterior $p(C_j|\mathbf{x})$ for class C_j. We assume uniform priors $p(C_k)$. Also, we make a naive Bayes assumption and regard the volume distribution in the different height bins as independent, which leaves us with:

$$p(C_j|\mathbf{x}) = \frac{p(\mathbf{x}|C_j) \cdot p(C_j)}{\sum_k p(\mathbf{x}|C_k) \cdot p(C_k)} = \frac{\prod_i p(x_i|C_j)}{\sum_k \prod_i p(x_i|C_k)}$$

4. Inference and Prediction

Inference. In a scene with n entities (persons/dynamic or static objects) there are $n \cdot (n-1)$ pairwise interactions. Despite the complex nature of predicting all interactions in a scene, exact inference is feasible for our model due to its constraining setup. For crowded scenes, we ensure quadratic scalability, whereas a fully connected model would grow exponentially. Using a simply pairwise model does not guarantee *scene consistency* though. This means that an object o might for example be detected to interact with two persons $p_{1/2}$ in a scene, being interpreted as a stroller in the first case and as a suitcase in the second. In reality, it cannot be both at the same time. We incorporate evidence from other interactions in the same scene by marginalizing object types over all pairwise assignments and thus interconnecting all C_o and C_p that belong to the same entity. This is done by iterating over the same frame for a fixed number of times. Each entity e interacts with every other of the n entities in two ways, once as *object* and once as *person*. After one iteration we hence computed $n-1$ many C_o^e and C_p^e that belong to e, respectively. A weighted combination, depending on the certainty of the corresponding action, is used as a prior on C_o^e and C_p^e in subsequent iterations.

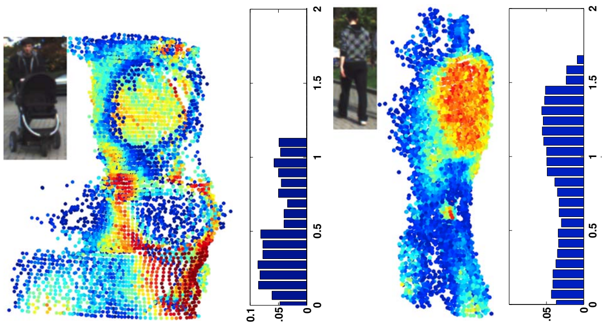


Figure 7: Visualization of the accumulated GCTs for a stroller (left) and a human (right) and the corresponding volumetric features. The color of GCT points corresponds to the significance of the ray represented by the number of accumulated distances.

Another clue we use for prediction is evidence from past frames. The rationale is that an object that has been detected as a person in one frame is likely (but not certain, due to tracking uncertainties) to be a person again in the next one. Again, we set priors on the corresponding distributions from one frame to another (*c.f.* dashed lines in Fig. 2).

Prediction. Having acquired a certain level of semantic scene understanding, we can now use our Bayesian network to also support other tasks. For example, tracking can be facilitated in a setting where objects are occluded or lost. Knowing that a person pushed a stroller s in the past frames raises the suspicion he will do so again in the current frame. Suppose we lost track of this stroller. We can plug this information into our model and infer a probability distribution of the expected location of the lost object. Furthermore, we can infer the relative position of s to all other entities $j \in J$ for the set of all entities J that it interacted with in the past frame. The more interactions were observed before, the more certain we can be when inferring the new position x_s: $p(x_s|J) \sim \prod_{j \in J} \mathcal{L}\left[x_j; p(x_{rel(s,j)}|I_{sj}, C_p^j, C_o^s)\right]$, where $\mathcal{L}\left[\mathbf{x}; \mathbf{p}\right]$ is the probability distribution of positions according to $\mathbf{p}$ (*i.e.*, a log polar grid as in Fig. 3) around the center point $\mathbf{x}$.

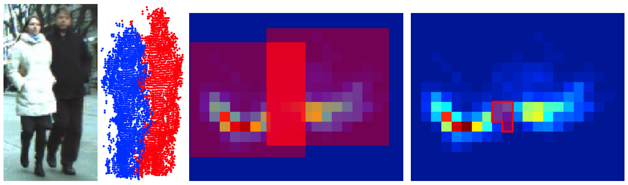


Figure 8: Visualization of the proposed overlap measure: (1) image cutout, two people walking closely together, (2) visualization of corresponding GCTs, (3) standard approach for overlap computation assuming a fixed-size footprint, (4) ground projections of GCT rays and the intersection between GCTs foot prints.

	train		**test**	
Action	# Seq.	# Fra.	# Seq.	# Fra.
none	-	7200	-	9974
push	68	3496	47	1456
group	48	2485	62	2394
pull right	9	408	27	1535
pull side right	10	563	6	297
pull left	11	516	27	1329
pull side left	7	417	3	119
sum (w/o none)	153	7885	172	7130

Table 1: Statistics on number of actions on training and test sets.

5. Robust 3D Data Association and Tracking

Overview. Fig. 5 shows an overview of our tracking system that we use for generating observations (the positions, velocities and 3D object shapes) for the proposed graphical model. Given a stereo pair of images and depth maps, we first generate regions-of-interest (ROIs) by projecting the 3D points onto a ground plane (a). The ROIs are then segmented into individual object areas (b). The center of mass projected onto the plane of the individual object and the 3D points embedded by the segmented area form the input for our multi-hypothesis tracker. In each frame, the newly extracted objects are linked to trajectory hypotheses on the ground plane by starting new trajectories backwards in time (e) and extending already existing tracks with new observations (c). In order to capture the approximate shape of 3D objects, we use a recently proposed 3D-shape representation called *General Christmas Tree* (GCT) [13]. As shown in Fig. 5, the model consists of a center axis (which is initially placed at the center position of each segmented object) and several height layers from which rays are cast in a fixed number of directions up to the height of the object. With each ray, the distance distribution of observed 3D surface points within a certain cylindrical cross-section is captured over time. Thus, for each newly generated hypothesis from the tracker, we produce a GCT starting from the first inliers and updating it by propagating the GCT sequentially over all inliers of the hypothesis (f, g). In case of extending an existing trajectory(f), the GCT is updated by registering it to the point cloud of the new observation using ICP and accumulating the new distance information. With the process so far we obtain an over-complete set of trajectory hypotheses which we prune to a final set mostly consistent with the scene by applying model selection in every frame as proposed by [12]. Finally, positions, velocities and GCTs are passed to the graphical model for classifying person-object interaction.

ROI Extraction and Segmentation. The initial step of tracking is to generate ROIs for potential objects, given the depth information. A common approach for this task is to project the 3D points onto the ground plane to form a 2D histogram accumulating the density of points in each bin, Fig. 6(3). The bins are then thresholded and the remaining bins are grouped into connected components. However, such a simple approach ignores the fact that the target objects we are interested in for tracking need to be connected to the ground plane. As shown in Fig. 6(1), only the torso of the woman pushing the stroller is visible, which means that only these points will contribute to the histogram bins resulting in a very low bin value, as shown in Fig. 6(3), which will be rejected in the thresholding process. Instead, we propose a new procedure that splits the projection process over different height levels, as shown in Fig. 6(2). Starting with the highest level, we project all points above onto this level. In the next steps the points between two layers are projected to the lower layer and for each bin that is empty but was occupied in the layer above, we propagate the value from the layer above. With this process we obtain two distinctive modes for both objects, as shown in Fig. 6(4) and hence compensate for frontal occlusions.

For segmenting the ROIs into individual objects, we use the Quick Shift algorithm [18], which finds modes of a density by shifting from the initial position to a position with a highest density within a certain neighborhood. Each segmented ROI area, representing a potential target object, is passed to the tracker together with its associated 3D points.

3D Shape Representation (GCT). In order to capture the shape of the tracked 3D objects, we use the recently introduced GCT representation [13]. The GCTs are generated for each tracker hypothesis by placing the center of the GCT on the initial inlier (segmented region with the 3D points) of the hypothesis and casting radial rays over a number of discrete height levels. From the 3D points that fall inside a cylinder along the ray, only the distance from the closest point on the ray to the center axis is stored. In each step, when a new inlier is associated to a trajectory, the GCT is updated by new distances. With this we obtain accurate volumetric information for tracked objects, as shown in Fig. 7.

From the GCTs we generate for each trajectory hypothesis a volumetric feature (Fig. 7) which we use in the proposed model in order to classify the objects into different classes. Thus, for each valid trajectory we compute a volumetric histogram over height bins as follows: $|V_i| = \sum_{r_j \in V_i : support(r_j) > \theta} \text{med}(r_j)$, where V_i is the bin, $\text{med}(r_j)$

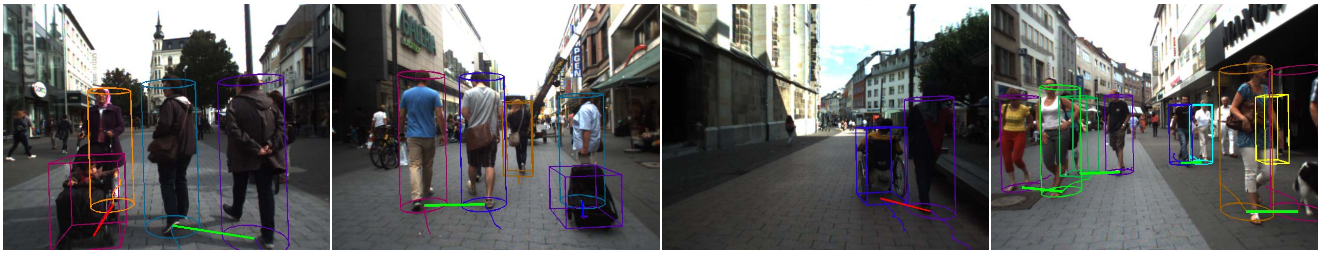


Figure 9: Result images showing tracked persons and their associated objects with correct action inference. Interactions are visualized by linking the footpoints of the interacting objects by a colored line. **Green**-group, **red**-push, **blue**-pull right, **magenta**-pull left.

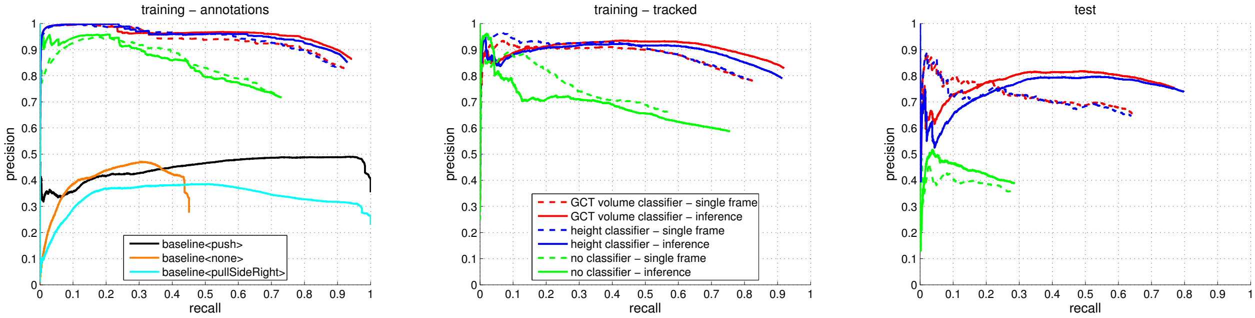


Figure 10: Interaction classification, full pairwise evaluation. (left) using manual point cloud segmentation annotations, (middle) using tracked point cloud data. (right) on dynamic scenes acquired in an inner city.

is the median distance of the ray r_j and $support(r_j) > \theta$ means that we consider only rays that have accumulated at least θ distances already, where θ is interlinked to the lifetime of the GCT. By using the support function we reject rays that originated from noisy outliers.

Measuring Overlap. In addition, we exploit GCTs in the model selection procedure, where we model the interaction between trajectories by considering the intersection between the footprints of individual tracks. A common assumption, used in tracking-by-detection approaches (*e.g.*, [6]), is that two objects cannot occupy the same spot in 3D space at the same time. Modeling object footprints by a fixed rectangular (or circular) shape leads to high interaction costs for close-by objects due to high overlap, as shown in Fig. 8(3), which can cause the rejection of one of the trajectory hypotheses. Instead, we propose an adaptive approach to compute the intersection of two objects based on their GCTs. For that, the reconstructed points of GCTs of both objects are projected onto the ground plane forming a 2D histogram, Fig. 8(4). The projected ray points are weighted by the number of distances of the corresponding ray and thus represent the significance of a ray and the ground projection bin. As shown in Fig. 8(4), the bin intersection between the objects is significantly smaller than in the fixed-footprint case, and using the weighting results in a low intersection value. The final intersection score is obtained by computing the Bhattacharyya distance between the two normalized histograms. This extension makes tracking more robust in our scenarios, since our objects of interest are usually situated close to a person.

Tracker. As our tracking core, we employ an extended version of the robust multi-hypothesis tracking framework presented in [12]. As input, the tracker requires the camera location from Structure from Motion (SfM), a ground plane and the segmented ROIs. From the 3D points of the segmented regions, we generate the footpoint positions of the objects by simply tracking the center of mass of the point cloud and projecting it onto the ground plane. Furthermore, the 3D points are back-projected to the image in order to obtain a color histogram for each object, which is required for the trajectory hypothesis generation process in order to associate the detections. The footpoint positions of the objects are linked to trajectories using a Kalman Filter with a constant-velocity motion model. In each frame, we run two trajectory generation processes: one looking backwards in time in order to generate new trajectories and one looking forward and extending the existing tracks. Using the Kalman Filter allows us to bridge gaps in detection caused by failures of the segmentation procedure. Since the new segmented areas are used for both processes, extension and generation of new hypotheses, each observation is assigned to two hypotheses. For resolving the ambiguity and selecting the hypotheses that are most consistent with the scene, we use model selection [12].

6. Experimental Results

Datasets. In order to train and test the proposed graphical model, we captured a dataset with a Bumblebee2 stereo camera containing 325 sequences with over 15,000 frames. For training, we manually segmented the ROI areas of indi-

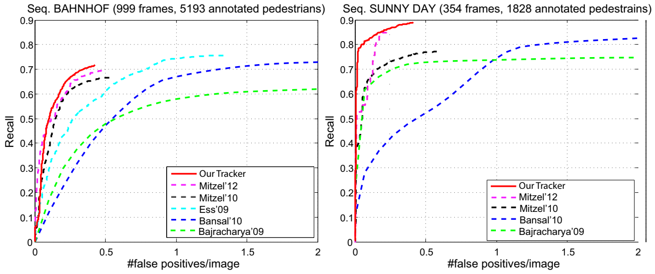


Figure 11: Pedestrian tracking performance on (left) BAHNHOF and (right) SUNNY DAY.

vidual objects and generated tracks (including the GCTs) using the proposed tracker. For each tracked object, we annotated an action and a reference object it is interacting with. The training dataset was captured in a controlled environment from a static setup in order to simplify the annotation process. For the test dataset, we acquired the images in crowded and challenging shopping streets from a moving platform with different object appearances and dynamics. In Tab. 1 we present detailed statistics of the action types in both sets. In total, we have annotated 153 sequences (7885 frames) as training and 172 sequences (7130 frames) as test set in order to asses the performance of our model. For the stereo estimation we used the robust approach from [8].

Tracking Performance. The person-object interaction classification strongly depends on the output of the tracker, since it requires positions, velocities and GCTs of the individual objects. For that reason, we first verify that our tracking approach is sufficiently robust for tracking in complex mobile scenarios. To this end, we experimentally evaluated our approach on two popular sequences, BAHNHOF and SUNNY DAY, courtesy of [6]. The sequences were acquired from a similar capturing platform in busy pedestrian scenes. We apply the evaluation criteria from [6] where the tracked bounding boxes are compared to manually annotated bounding boxes in each frame. Since our approach tracks all objects in the scene, but in this dataset only the pedestrians are annotated, we classify each segmented ROI using the pedestrian classifier from [7] before passing it to the tracker. Fig. 11 presents the performance curves in terms of recall vs. false positives per image. As can be seen, our approach surpasses state-of-the-art performance.

Interaction Classification. We evaluate our action detection framework on our annotated training data, as well as on real-life scenes described above. In order to asses the difficulty of the classification task, we first evaluate several simple baseline classifiers. These baselines follow two easy rules. First, if two objects are close together, they must interact in some way. Secondly, if both of these objects are persons then we just detected a *group*, else the baseline$\langle action \rangle$ detects $action$. In all other cases there is no interaction at all.

Furthermore, we compare the final action detector with

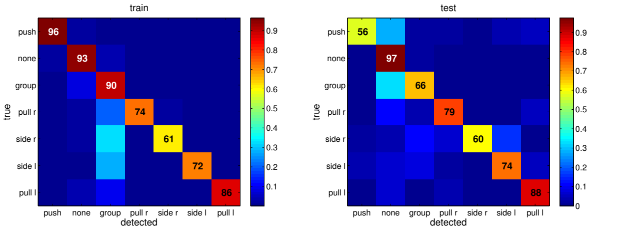


Figure 12: Confusion matrices of our action detection for training (left) and testing (right).

a detector based on a classifier that only takes into account the height of a tracked object as described in Sec. 2. Also, we try our detector without any classifier, *i.e.*, assuming a uniform distribution over object classes. The results for the crossvalidation on the training data are shown in Fig. 10 (left). At the same time, we compare the system without inference between subsequent frames to the integrated approach (*c.f.* dashed *vs.* solid lines). We only show three baselines (*push*, *none* and *pull side right*), since these dominate the other baselines. Clearly, the performance of our approach is above the other presented approaches. We reach a *mean average precision* (mAP) of 0.907 *vs.* an mAP of 0.893 for the runner-up, the full system using a height classifier. In general, the timely inference is better than performing inference in each frame separately: Single-frame mAP for our system is 0.869 (*c.f.* Fig. 10(left)). Just for the system without classifier (*c.f.* Fig 10 (left)) we get a better performance if we do not take into account evidence from past frames. The reason here is that we would only propagate mainly false detections and have a better chance of detection an interaction correctly if we take no priors into account.

Next, we perform the same experiment on our training data, but this time with actual results from our tracking pipeline instead of tracking results based on annotated object segmentations. This is shown in Fig. 10(middle). Because of the competitive performance of our tracking system, we do not lose much against the results in our experiments before. The mAP reduces from 0.907 to still 0.838 for our detector. Finally, we evaluate the performance of our action detector for the test set of challenging scenes with a dynamic camera. We reach an mAP of 0.624 with the full combination of our tracker, object classifier based on GCTs, interaction model and frame inference (*c.f.* Fig. 10(right)).

Taking a deeper look into the failures of our action detector (*c.f.* Fig. 12) reveals that we perform consistently well on the action *none*, which means we have just few false positives. Transitioning from training to test data, we lose most accuracy in the actions *group* and *push*. All other action detection accuracies stay high.

Object Classification. In Fig. 4 we show the classification performance of our new classifier in comparison with a simpler height-based classifier. We also compare to a third

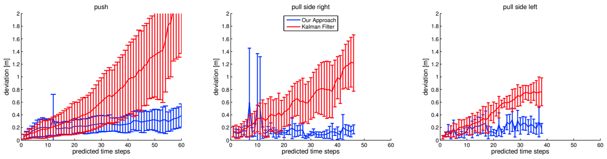


Figure 13: Error Bars of position prediction.

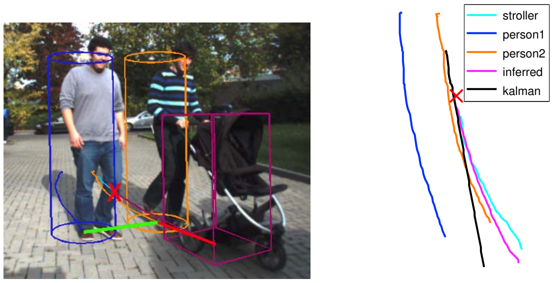


Figure 14: (Left) Tracked observation, lost stroller at the red X. (Right) Prediction results for Kalman Filter and our approach. Color-coded correspondences between left and right.

classifier based on accumulated height information from the GCTs. The performance for all object types is also shown.

Activity Prediction. As mentioned in Sec. 4, we can use our model to also perform a predictive task. In our evaluation we compare this prediction against a linear extrapolation by a *Kalman filter*. We measure success in this test as the closest prediction to the actual path. When we lose track of an object, the Kalman filter will predict future positions based on its underlying motion model. Our inference-based prediction observes the positions of all other entities in the scene and uses the interaction distribution it learned so far to infer the most likely position of the lost object. Fig. 14 illustrates a typical setup. We run these tests on our training data. Tracking is supposed to be lost after 15 frames and all remaining frames are predicted by the Kalman filter and our model. The results of this experiment are shown in Fig. 13. We plot the mean prediction distance including uncertainty *vs*. number of frames looked ahead. With an increasing number of frames, the Kalman filter diverges significantly more than our approach.

7. Conclusion

We have presented a framework that can track both known and unknown objects and simultaneously infer which objects belong together. Furthermore, the proposed model can be used to infer object types and the interaction patterns occurring between associated objects. The action classification has two advantages. On the one side, it can help improve predictions for the continuation of each trajectory in case of detection/tracking failures. On the other side, it can be used for adaptation of dynamic models for certain object-person constellations. For the future, we plan to extend the model to more object and interaction types.

Acknowledgments. This research has been funded by ERC Starting Grant CV-SUPER (ERC-2012-StG-307432).

References

[1] M. Andriluka, S. Roth, and B. Schiele. Monocular 3D Pose Estimation and Tracking by Detection. In *CVPR*, 2010.

[2] M. Bajracharya, B. Moghaddam, A. Howard, S. Brennan, and L. Matthies. A Fast Stereo-based System for Detecting and Tracking Pedestrians from a Moving Vehicle. *IJRS*, 2009.

[3] M. Bansal, S. H. Jung, B. Matei, J. Eledath, and H. S. Sawhney. A real-time pedestrian detection system based on structure and appearance classification. In *ICRA*, 2010.

[4] W. Choi and S. Savarese. A Unified Framework for Multi-Target Tracking and Collective Activity Recognition. In *ECCV*, 2012.

[5] D. Damen and D. Hogg. Detecting Carried Objects in Short Video Sequences. In *ECCV*, 2008.

[6] A. Ess, B. Leibe, K. Schindler, and L. Van Gool. Robust Multi-Person Tracking from a Mobile Platform. *PAMI*, 2009.

[7] P. Felzenszwalb, B. Girshick, D. McAllester, and D. Ramanan. Object Detection with Discriminatively Trained Part-Based Models. *PAMI*, 2010.

[8] A. Geiger, M. Roser, and R. Urtasun. Efficient Large-Scale Stereo Matching. In *ACCV*, 2010.

[9] C. Keller, D. Fernandez-Llorca, and D. Gavrila. Dense Stereo-based ROI Generation for Pedestrian Detection. In *DAGM*, 2009.

[10] C.-H. Kuo, C. Huang, and R. Nevatia. Multi-Target Tracking by On-Line Learned Discriminative Appearance Models. In *CVPR*, 2010.

[11] B. Lau, K. Arras, and W. Burgard. Tracking Groups of People with a Multi-Hypothesis Tracker. In *ICRA*, 2009.

[12] B. Leibe, K. Schindler, and L. Van Gool. Coupled Object Detection and Tracking from Static Cameras and Moving Vehicles. *PAMI*, 2008.

[13] D. Mitzel and B. Leibe. Taking Mobile Multi-Object Tracking to the Next Level: People, Unknown Objects, and Carried Items. In *ECCV*, 2012.

[14] J. Pearl. Fusion, Propagation, and Structuring in Belief Networks. *Art. Intell.*, 1986.

[15] S. Pellegrini, A. Ess, K. Schindler, and L. Van Gool. You'll Never Walk Alone: Modeling Social Behavior for Multi-Target Tracking. In *ICCV*, 2009.

[16] S. Pellegrini, A. Ess, and L. Van Gool. Improving Data Association by Joint Modeling of Pedestrian Trajectories and Groupings. In *ECCV*, 2010.

[17] K. Smith, P. Quelhas, and D. Gatica-Perez. Detecting abandoned luggage items in a public space. In *PETS*, 2006.

[18] A. Vedaldi and S. Soatto. Quick Shift and Kernel Methods for Mode Seeking. In *ECCV*, 2008.

[19] C. Wojek, S. Walk, S. Roth, and B. Schiele. Monocular 3D Scene Understanding with Explicit Occlusion Reasoning. In *CVPR*, 2011.

[20] K. Yamaguchi, A. Berg, L. Ortiz, and T. Berg. Who are you with and where you are going? In *CVPR*, 2011.