A Fixed-Dimensional 3D Shape Representation for Matching Partially Observed Objects in Street Scenes

Dennis Mitzel, Jasper Diesel, Aljosa Osep, Umer Rafi, Bastian Leibe

Abstract— **In this paper, we present an object-centric, fixed-dimensional 3D shape representation for robust matching of partially observed object shapes, which is an important component for object categorization from 3D data. A main problem when working with RGB-D data from stereo, Kinect, or laser sensors is that the 3D information is typically quite noisy. For that reason, we accumulate shape information over time and register it in a common reference frame. Matching the resulting shapes requires a strategy for dealing with partial observations. We therefore investigate several distance functions and kernels that implement different such strategies and compare their matching performance in quantitative experiments. We show that the resulting representation achieves good results for a large variety of vision tasks, such as multi-class classification, person orientation estimation, and articulated body pose estimation, where robust 3D shape matching is essential.**

I. Introduction

In this paper, we address the problem of matching partially observed, potentially articulated 3D shapes. Such matching problems occur in many practical scene understanding scenarios based on 3D sensors, for example in mobile robotics or autonomous driving [1], [2]. When observing a scene with stereo or laser sensors, it is relatively easy to extract object candidates by looking for groups of 3D points that extend beyond the ground surface [3], [4], [5], [6]. However, further classification and detailed analysis of those object candidates faces severe challenges. While a rough division into major object categories such as *pedestrian*, *car* or *bicyclist* can still be achieved based on the 3D region sizes, robust rejection of outliers requires a more detailed shape analysis [6]. The same applies when more detailed results shall be estimated, such as body orientations or even detailed body poses [7], [8].

A main challenge when analyzing and matching 3D shapes from stereo or laser data is that the resulting 3D information is often quite noisy and only contains a partial view of an object's 3D shape, since in each frame only the object surface facing the sensor can be observed. If an object is moving, the visible part of the surface will change with the viewing angle, and non-rigid motions will introduce further distortions. A good representation for matching is therefore hard to find. Point cloud [4] or surfel [12], [13] representations are often used in robotics and automotive applications, but they are hard to match due to the sparse representation and the noisy measurements. Mesh representations, as often used in Computer Graphics (*e.g.*, [9]), are not applicable, since the shape is only partially observed. Part-based representations have therefore been proposed based on 3D features [10], height slices [11], or volume subdivisions [14].

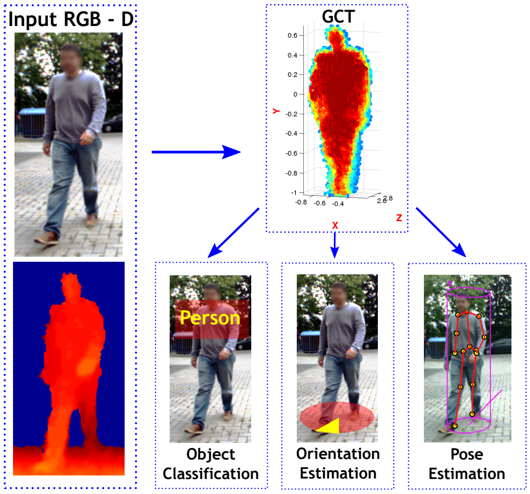


Fig. 1: We propose an object-centric, fixed-dimensional representation and corresponding kernel functions that enable robust matching for generic partial 3D shapes. This new shape representation can be applied to a large variety of vision applications such as multi-class classification, person orientation and articulated body pose estimation.

When looking at other fields, such as object categorization [15] or action recognition, an important lesson from the past is that powerful global descriptors are immensely valuable. A main reason why the HOG descriptor [16] became so popular for all kinds of object detection tasks is that it provides a highly discriminative, global, and fixed-dimensional representation for object appearance that makes it far easier to apply in machine learning approaches than complex part-based models. The same argument can be made for the bag-of-words representation for object categorization [17] and for the Motion History Image [18], [19] for action recognition in surveillance scenarios. All of those representations have in common that they define a fixed-dimensional descriptor that is easy to use and readily applicable for a large variety of machine learning tasks.

In this paper, we take inspiration from the above examples and propose an object centric, fixed-dimensional representation for matching partial 3D object shapes. Our approach integrates 3D measurements from several video frames into a common object-centric, cylindrical coordinate system and then subdivides the individual measurements into a discrete set of bins, spanning shape variation in height, orientation angle, and time. For each bin, we keep a list of observed distances (the "motion history" of the corresponding surface

Computer Vision Group, RWTH Aachen University, {mitzel, osep, rafi, leibe}@vision.rwth-aachen.de

point) that together characterize the observed shape and its evolution over time. The resulting mhGCT representation extends the "Generalized Christmas Tree" (GCT) model that was previously used for tracking [5] by taking into account the detailed motion histories, which allows for more robust matching. We then propose several distance functions and kernels that build upon this representation and show that they can be applied to a large variety of visual analysis tasks, including multi-class object categorization, person orientation classification and articulated body pose estimation.

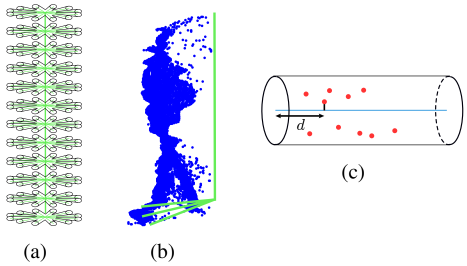
The paper is structured as follows. The next section discusses related work. Sec. III then introduces our shape representation, and Sec. IV proposes several distance functions and kernels. Finally, Sec. V presents experimental results.

II. Related Work

When designing a new feature for classification or regression tasks, the goal is to find a representation that can easily be used with different machine learning techniques, while providing sufficiently discriminative information to support the task. The history of object detection and activity recognition approaches nicely illustrates this point. Simple global feature representations such as Histograms of Oriented Gradients (HOG) [16] or Motion History Images (MHI) [18], [19] have become very popular, since they result in fixed-dimensional feature vectors that can directly be used in, *e.g.*, SVMs. In contrast, interest point based or part-based representations, such as the Constellation model [20] or ISM [21], are relatively unwieldy, since they start from a variable number of input features and have to employ a complex algorithmic pipeline to integrate their contributions into a consistent recognition score. Consequently, progress has been much faster for approaches building upon fixed-dimensional representations – even if those representations are less expressive on their own, the ability to combine them with powerful machine learning techniques more than makes up for this limitation.

Up to now, however, there is no universally accepted representation for matching partially observed 3D surface shapes, as can be obtained from stereo, RGB-D, or LIDAR sensors. Although different variations on the HOG idea have been proposed for indoor object recognition from Kinect RGB-D data, such as Histograms of Oriented Depths (HOD) [22] or Histograms of Oriented Normals (HONV) [23], they are poorly suited for outdoor applications, where sensors can only deliver sparse (in case of LIDAR) or noisy (in case of stereo) depth information. In those applications, a single depth scan is often not sufficient for robust recognition and temporal integration of several measurements becomes desirable, which is problematic for articulated objects. While there are several approaches in the computer graphics community that can robustly align and match articulated 3D shapes [9], [24], [25], they require a full surface mesh to be available, rendering them inapplicable here.

When considering only a single object class, it is often possible to use fixed proxy shapes in order to match object appearances. For example, [26] define a cylindrical proxy shape onto which they map observed surface pixel colors for 3D people tracking and re-identification in multi-view surveillance applications. However, such representations cannot adapt to different object shapes.



(a) (b) (c)

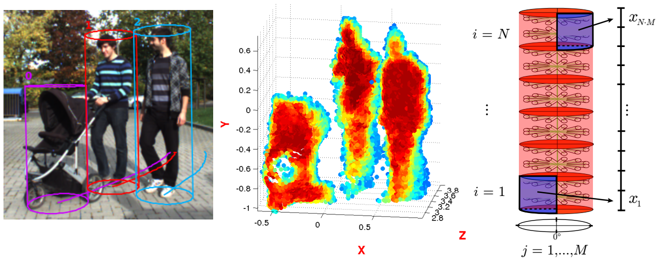
Fig. 2: The GCT generation process. A GCT consists of a vertical center axis and several height layers with associated uniformly spaced rays (a) that together form a non-parametric representation of the object's surface shape. When a GCT receives new measurements (b), each ray is updated with the distance from the center axis to the 3D surface point closest to the ray (c).

In the robotics community, a number of methods have been proposed for multi-class classification using 3D laser scans [27], [28], [6]. However, those approaches rely on local features or part representation in order to describe the different object classes, resulting again in complex matching procedures.

In this work, we propose a fixed-dimensional object representation that is suitable for capturing the surface shape of partially observed, possibly articulated objects, as well as its evolution over time. We build upon the GCT shape representation proposed by Mitzel *et al.* [5], [29], which accumulates shape information of generic tracked objects over time. In the original papers [5], [29], this representation has been shown to be beneficial for precise frame-to-frame alignment in tracking, but its potential has not been explored for recognition tasks yet. Our main contribution in this paper is to demonstrate how a motion-history augmented GCT representation (mhGCT) can be used for more general multi-class classification, orientation and pose estimation tasks. For this, we propose suitable distance functions and kernels and show their usefulness on several benchmark datasets.

III. Generalized Christmas Tree (GCT) Representation

General GCT Representation. Fig. 2(a) visualizes the GCT representation [5] we use as the basis for our feature representation. A GCT represents an object by a number of regularly sampled surface measurements ("rays") in a cylindrical coordinate system that encode the distance from the surface to the cylinder's upright center axis. Given a point cloud for a candidate object (*e.g.*, obtained from a stereo depth map) and an estimate for the object center, the GCT is built up by casting rays over a fixed number of height and angle levels (*c.f*. Fig. 2(b)) and collecting all surface points that fall within a small volume around the ray. For each ray, the point closest to the ray is selected and its distance to the central axis is taken as a measurement. When tracking an



(a) (b) (c)

Fig. 3: mhGCT extraction procedure. (a) Object tracking results. (b) Reconstructed GCTs of the scene in (a) where the color of GCT points corresponds to the significance of the ray represented by the number of accumulated distances. (c) Binning procedure to create the mhGCT feature vector.

object over time, the GCT is aligned to the next frame's point cloud using upright ICP [30] (*i.e.*, restricting the transformation to a 2D translation on the ground plane and a 1D rotation around the ground plane normal), and each ray receives a new distance measurement. Over time, each ray thus builds up a distribution over observed surface distances, from which a robust median estimate can be extracted. In [5], Mitzel *et al.* propose to use this median estimate, weighted by the inverse variance of the ray's distance distribution, for ICP alignment. They show that this results in more precise point alignment for articulated objects and thus more robust tracking.

mhGCT as Feature. In this paper, we propose to create a new slice-wise discretization of the GCT that additionally captures each ray's motion history over a fixed temporal window. As illustrated in Fig. 3, our proposed mhGCT coarsely discretizes the surface information contained in the GCT rays into a fixed number of vertical, angular, and temporal bins. For each bin, we keep either only the median distance (mhGCT-med) or a histogram over the contained distances (mhGCT-hist). In both cases, this results in a fixed-dimensional feature vector for the partially observed object shape. Matching object shapes now boils down to comparing mhGCT vectors. However, as we argue in the following, several complications still need to be overcome in order to make this matching effective.

Most importantly, relying on depth information we always obtain only a partial view of the object. The GCT accumulates surface information over time and can thus complete object shapes whenever objects turn. However, we need to ensure that we always compare corresponding mhGCT bins during the matching procedure, *e.g.*, by aligning the extracted GCTs to a canonical orientation. When handling moving objects, we can use the trajectories obtained from the tracker in order to estimate each object's moving direction and rotate the GCT to a fixed orientation facing the camera. For static objects, this is unfortunately not possible; depending on the task, it may thus be necessary to compare mhGCTs at several different rotations.

In addition, since the surface information stored in mhGCTs depends on both the object's orientation and on its position relative to the depth sensor, different views of the same object will only partially overlap. Also, depending on the size of an object, bins belonging to high height layers might be empty. Therefore, when we compare two mhGCTs bin-by-bin, it will often happen that we compare empty with full bins. This raises the question how we should deal with bins for which we have no information. In the following, we will explore different distance functions for our mhGCT feature that handle empty bins explicitly.

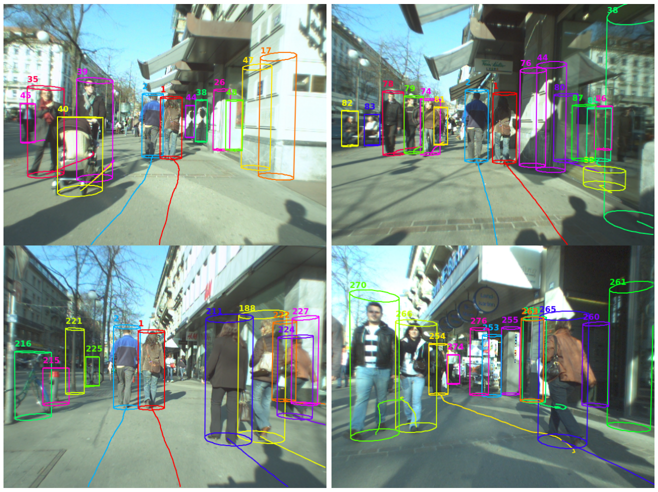


Fig. 4: Four example images from the SUNNY DAY sequence with tracking boxes for `person` and `non-person` class generated by our tracker.

IV. DISTANCE FUNCTIONS

When comparing mhGCT bins, three cases can occur. In the first case, both bins are empty. We do not want to penalize this case and hence their distance should always be set to 0. The second case occurs if exactly one of the two bins is empty. In this case, we want to enforce a penalty in order to discourage misalignments. In the last case, both bins contain valid distances, which we can compare using a variety of distance functions. In the following, we present several possible distance functions and evaluate their suitability for matching.

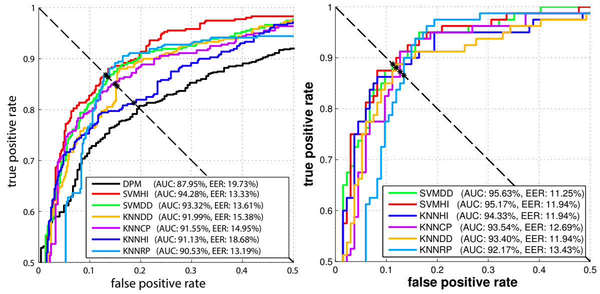
Constant Penalty. A simple solution is to use Euclidean distances between bin medians and to assign a constant, learned penalty p whenever an empty bin is matched with a full one. This can be done as follows:

$$d_{CP}(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n} b(x_i, y_i) \tag{1}$$

where

$$b(x_i, y_i) = \begin{cases} 0, & \text{if } x_i = \emptyset \wedge y_i = \emptyset \\ p, & \text{if } (x_i = \emptyset \wedge y_i \neq \emptyset) \\ & \quad \vee (x_i \neq \emptyset \wedge y_i = \emptyset) \\ (\mathrm{med}_{x_i} - \mathrm{med}_{y_i})^2, & \text{otherwise} \end{cases} \tag{2}$$

Relative Penalty. In order to reduce the sensitivity to noise, the bin penalties can be weighted by the proportion of valid rays in the specific bin with respect to the total number of



(a)

(b)

	height	**angle**	**penalty**	**histBins**	**k**
KNNCP	30	17	0.31	-	4
KNNRP	9	2	1.41	-	4
KNNDD	20	21	0.02	-	6
KNNHI	26	17	-	6	6
SVMDD	26	17	0.02	-	-
SVMHI	29	9	-	2	-

(c)

Fig. 5: Two-class (person, non-person) classification performance on the BAHNHOF test sequence using our four different distance function combined with KNN and SVM classifiers, compared to a DPM baseline [31]. (a) Performance when the trajectories are split into 10-frames tracklets and the mhGCTs for classification are extracted from the 10-frames tracklets. (b) Performance when using the full trajectories and the corresponding extracted mhGCTs. (c) Optimal parameter values obtained by cross-validation on the SUNNY DAY sequence for each of the six best classifiers. Note: for the KNNDD and SVMDD classifiers, the *penalty* corresponds to the learned default distance.

valid rays in the respective mhGCT. We formally define the corresponding distance as follows:

$$d_{RP}(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{n} p \left| \frac{r_i}{r} - \frac{r'_i}{r'} \right| + \sum_{\substack{j=1\ldots n,\\ x_j \neq \emptyset \wedge y_j \neq \emptyset}} (\mathrm{med}_{x_j} - \mathrm{med}_{y_j})^2, \quad (3)$$

where r_i and r'_i are the number of rays in the corresponding bins and r and r' the total number of rays in the corresponding mhGCT.

The two previously described distance functions do not fulfill the triangle inequality and hence are not metrics. This is mainly due to the fixed penalty p, regardless of whether it is weighted or not. In the following, we therefore present two measures which are explicitly designed to be metrics, allowing us to employ them in a kernel.

Default Distance. In this measure, we drop the case differentiation whether a bin contains rays or not and just assign to each empty bin a default distance. Then we can treat this default value completely the same as any other median distances from bins containing valid rays. For comparing the resulting mhGCT feature, we again use Euclidean distances.

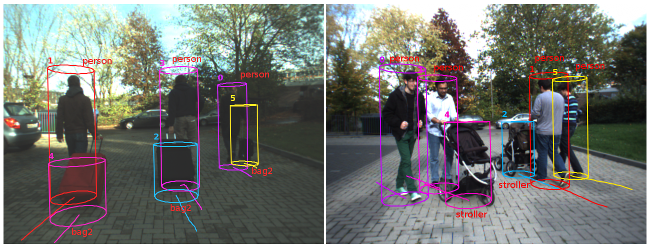


Fig. 6: Qualitative results in the multi-class classification problem using mhGCT features with an SVMHI-classifier.

$$d_{DD}(\mathbf{x},\mathbf{y}) = \sqrt{\sum_{i=1}^{n} (d_{x_i} - d_{y_i})^2} \quad (4)$$

$$\text{where } d_{x_i} = \begin{cases} d_{default}, & \text{if } x_i = \emptyset \\ \mathrm{med}_{x_i}, & \text{otherwise} \end{cases}$$

Histogram Intersection. The distance functions presented so far compare mhGCTs using the median distances of all valid rays. This way, we are able to represent an entire bin by just a single number. While this results in efficient computation, it also loses information, since the original distance distributions are discarded. We therefore propose another distance metric that uses histogram intersection to compare (normalized) distance distributions of corresponding bins. Similar to the *Default Distance*, we represent empty bins by a fixed default histogram. We define this default histogram to be the inverted mean histogram of the corresponding bin across a training set of mhGCTs (*i.e.*, each default histogram cell contains one minus the value of the corresponding mean histogram cell). This way, we maximize the distances to empty bins, while staying within a metric histogram comparison framework.

$$d_{HI}(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{n} d_{\cap}(x'_i, y'_i) \quad (5)$$

$$\text{where } x'_i = \begin{cases} x_{default}, & \text{if } x_i = \emptyset \\ x_i, & \text{otherwise} \end{cases}$$

From Distances to Classifiers. In the following, we apply all four distance functions together with a KNN classifier. Since d_{DD} and d_{HI} are metrics, we can additionally define kernels for them for use in a kernel SVM classifier.

$$k_{DD}(\mathbf{x},\mathbf{y}) = \exp\{-g \cdot d_{DD}(\mathbf{x},\mathbf{y})\} \quad (6)$$

$$k_{HI}(\mathbf{x},\mathbf{y}) = \exp\{-g \cdot d_{HI}(\mathbf{x},\mathbf{y})\} \quad (7)$$

We have thus defined four different distance measures and two kernels for comparing mhGCT features, corresponding to different strategies of handling empty bins. Especially the kernels are a valuable contribution, since they allow us to use mhGCT features with a large variety of powerful machine learning techniques, such as kernel SVMs, kernel PCA, or Gaussian Processes. Note that this has only become possible through our object-centric, fixed-dimensional mhGCT representation and our definition of suitable default handling strategies for empty bins. In the following section, we compare the different distance functions experimentally.

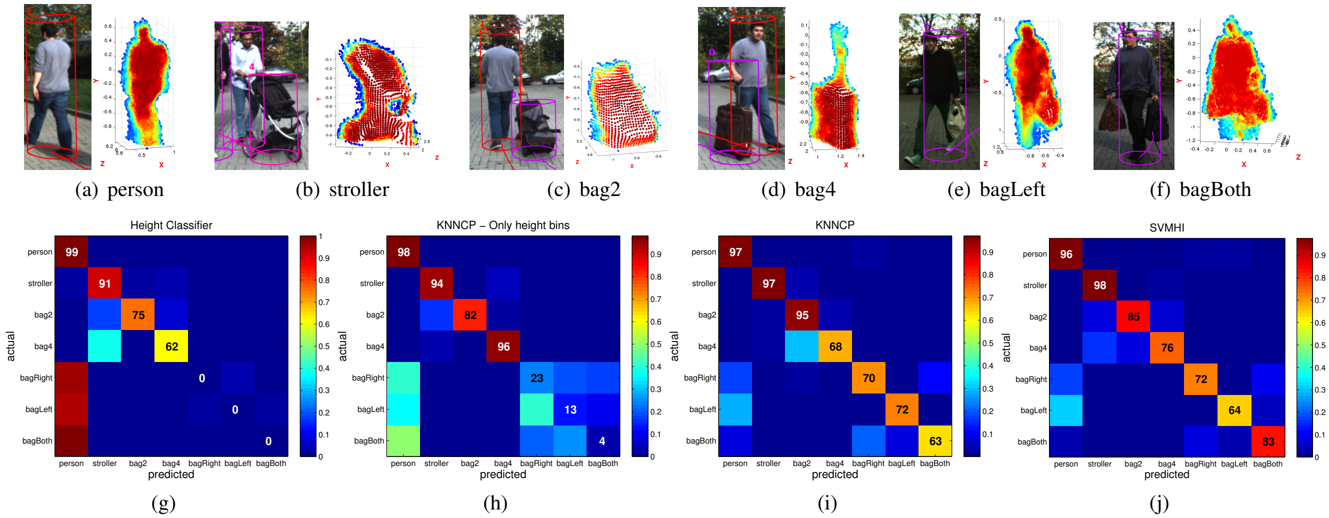


Fig. 7: (top row) Object classes used in our multi-class experiments together with example GCTs. (bottom row) Multi-class classification performance presented as confusion matrix for two baseline approaches and the KNNCP and SVMHI classifiers using our mhGCT feature: (g) Baseline results with a simple KNN height classifier. All person with bag classes are confused with the person class. (h) Baseline performance when using the volume histogram classifier, as proposed by [29]. (i) Results when using *constant penalty* distance function with our mhGCT feature and KNN as classifier. (j) Results when using SVMs with *histogram intersection* kernel with our mhGCT feature.

V. Evaluation

In order to demonstrate the effectiveness of our new mhGCT shape representation, we apply it to a variety of matching problems, including pedestrian classification, multi-class classification, pedestrian orientation estimation and articulated body pose estimation. In the following, we present an extensive evaluation for all those problems, underlining the generality of our new shape representation.

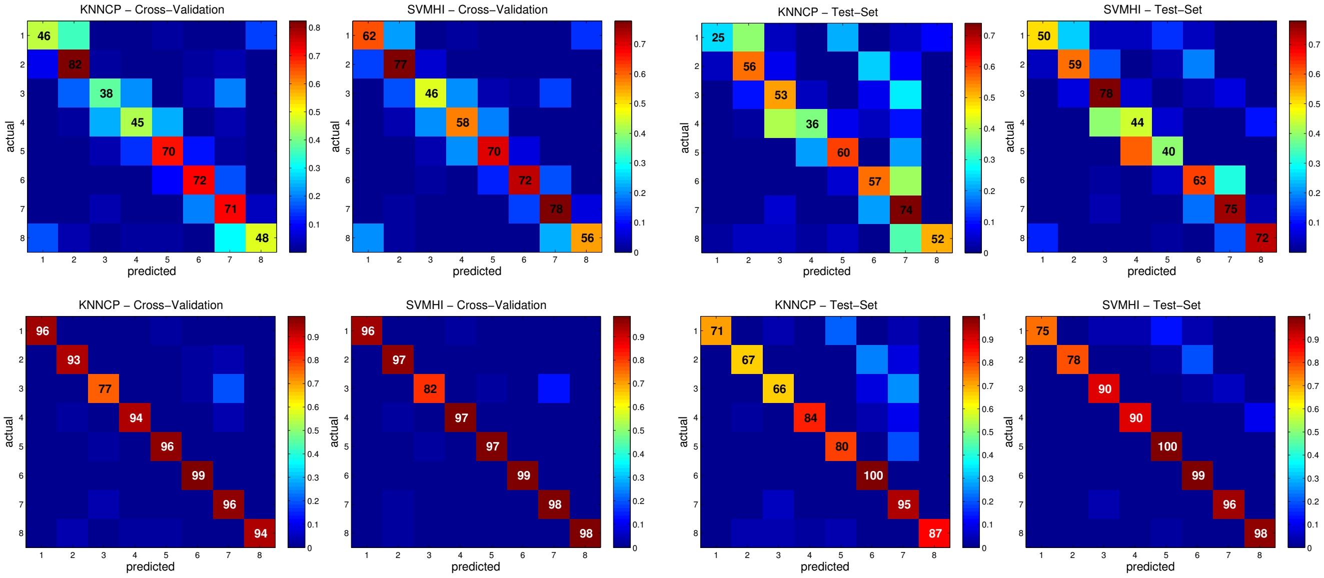
Pedestrian Classification. We first analyze the effect of each distance function by applying them to a 2-class classification problem person *vs.* non-person. We combine all four distance functions with a KNN classifier, resulting in the variants KNNCP, KNNRP, KNNDD, and KNNHI. In addition, we apply kernel SVMs with k_{DD} and k_{HI}, resulting in the variants SVMDD and SVMHI.

For training and testing, we use two popular sequences, Bahnhof and Sunny Day from the Zurich Mobile Pedestrian Corpus, courtesy of [32]. Both sequences were acquired from a stereo rig mounted on a child-stroller moving on crowded sidewalks. We extracted mhGCTs using the *tracking-before-detection* framework from [5], [29], which is based on low-level stereo region segmentation and multi-hypothesis data association. This approach enables us to track a large variability of objects that are present in the scene, including pedestrians, child strollers, trash bins, buildings, or traffic signs, but it also generates many spurious responses from scene clutter and facade structures, as shown in Fig. 4. The goal of this experiment is to find out how well the proposed mhGCTs can separate true pedestrians from this clutter and how the different classifier variants compare for this task.

We labeled all extracted trajectories and the corresponding mhGCTs for all objects belonging either to the person or non-person class for both sequences and used the Sunny Day sequence for training and the Bahnhof sequence for testing. As all distance functions depend on a number of parameters, we perform cross-validation embedded into a grid search on the annotations of Sunny Day to determine the optimal parameters (*c.f.* Fig. 5(c)). It should be noted that constructing GCTs that are sufficiently well-aligned for matching requires a robust estimate for each GCT's object center. We therefore employ the procedure proposed in [5] and re-estimate the object center from the stored GCT distances after several frames have been observed.

Fig. 5 shows the resulting ROC curves for all 6 classifiers. We compare both the classification performance based on full trajectories (Fig. 5(b)) and the performance when splitting each trajectory into 10-frame tracklets and generating mhGCTs for each tracklet (Fig. 5(a)). The latter is motivated by an online application, where classification would then be based only on a 10-frame time window. As can be seen from those plots, the mhGCTs achieve good classification performance, with the two SVM variants consistently outperforming the KNN versions. When splitting the trajectories into 10-frame mhGCTs, the performance drops only slightly (95.63% AUC *vs.* 94.28% AUC), which means our shape representation becomes robust already with few frames.

In order to put those results into perspective, we compare the 10-frame tracklets results to the baseline classification performance obtained by applying an appearance based DPM detector [31] to the last frame of each 10-frame tracklet (in a region-of-interest corresponding to 1.5 times the tracked person's bounding box). As can be seen from Fig. 5, the DPM achieves a lower performance with 87.95% AUC, showing that our shape representation is competitive and can provide a viable complement to appearance-based classifiers. Based on the above results, we only report performances


KNNCP – Cross–Validation

SVMHI – Cross–Validation

KNNCP – Test–Set

SVMHI – Test–Set

KNNCP – Cross–Validation

SVMHI – Cross–Validation

KNNCP – Test–Set

SVMHI – Test–Set

(a) (b) (c) (d)

Fig. 8: (top row) Confusion matrices of orientation estimation using our mhGCT in combination with different classifiers for 8 different orientation classes (a) Cross-validation with the KNNCP-classifier. (b) Cross-validation with the SVNHI-classifier. (c) Performance on the independent test set using KNNCP-classifier. (d) Performance on the independent test set using SVMHI-classifier. (bottom row) Obtained results if we count the prediction in the two nearest neighbor classes as true as well. As can be seen, most confusions occur only with the two closest neighboring orientation classes.

for the KNNCP and SVMHI classifiers in the following experiments.

Multi-Class Classification. In order to assess the performance in a multi-class problem, we use the dataset from [29], which contains 325 sequences with more than 15,000 frames recorded with a Bumblebee2 stereo camera. The dataset contains four different classes (*c.f*. Fig. 7(a-d)) such as `person`, `child stroller`, `bag2` (a 2-wheel bag that can be pulled behind a person) and `bag4` (a 4-wheel bag that can be pushed). In order to get more variability, we augmented the dataset by three additional classes, namely persons carrying a bag in the left hand (`bagLeft`), the right hand (`bagRight`), or both hands (`bagBoth`), as shown in Fig. 7(e-f). We again apply our SVMHI and KNNCP classifiers. In addition, we include two baseline classifiers: a simple height classifier (Fig. 7(g)) and a KNNCP classifier with a mhGCT representation that only considers height bins and a single angular bin per height level (Fig. 7(h)). This simulates the volume histogram classifier proposed by [29].

Obviously, these classes cannot be distinguished by a simple height classifier as shown in Fig. 7(g), where classes such as `bagBoth, bagLeft, bagRight` are always confused with the `person` class. However, when using mhGCT with the *Constant Penalty* distance function and KNN as classifier we obtain remarkable performance, over 95% for classes `person, stroller, bag2` and over 63% for the remaining classes (*c.f*. Fig. 7(i)). SVMHI performs similarly well, as shown in Fig. 7(j). In Fig. 7(h) we present the performance when simulating the volume histogram classifier as proposed by [29]. Here we set the number of angle bins to one and the number of height bins to 20, as proposed in [29]. For the four original classes as introduced by [29], we obtain similar performance as reported in [29]; however, for the three new additional classes the performance is quite low. This result is expected since it is difficult to distinguish between classes such as `person` and `bagBoth, bagLeft, bagRight` just using one angle bin. However, when employing mhGCT with 20 angle bins and KNNCP or SVMHI, the performance for the new classes increases significantly.

Orientation Estimation. Next, we evaluate our new mhGCT representation on a pedestrian orientation estimation task. For this, we used the pedestrians from the dataset proposed by [29] as our training set. Each mhGCT obtained from a 10-frames tracklet was annotated with the orientation obtained from the tracker, binned in an 8-bin orientation histogram whose bins are evenly spread over 0 to 360 degrees. It is important to note that for this experiment we do not rotate the mhGCT in order to face the camera, since we assume that the walking direction is unknown at test time. In total we obtained over 700 labeled mhGCT vectors for training. The results of cross-validation are presented in Fig. 8(a top) using KNNCP and Fig. 8(b top) when using SVMHI. The obtained results are quite promising, especially if we consider the confusions of each class which are mostly the two nearest neighbors (left and right). To illustrate this, we plot in Fig 8(a,b bottom) also the confusion matrices when we count an orientation estimate in the directly adjoining orientation bins to be acceptable as well. As expected, the optimal parameter settings for this experiment show a larger number of angular bins (22 for KNNCP and 30 for SVMHI) than for the 2-class problem (*c.f*. Fig. 5(c)).

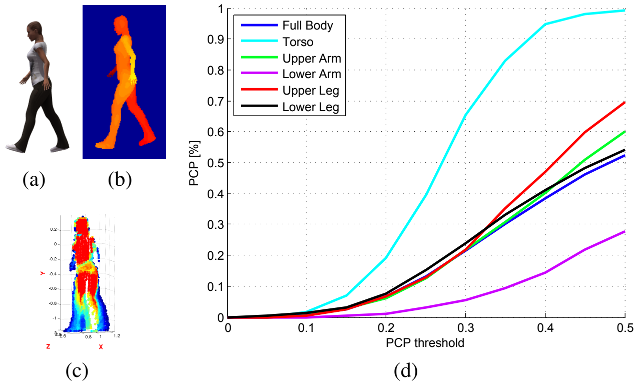


Fig. 9: (left) Synthetic data used for creating a database of mhGCTs annotated with exact pose: (a) RGB image and (b) corresponding depth image of one of the models; (c) extracted GCT using a 12-frames window. (d) Quantitative pose estimation results using the PCP evaluation criterion (percentage of correct body parts) introduced by [33].

Fig. 8(c,d) present the results on an independent test set which contains mhGCTs from pedestrians from BAHNHOF, SUNNY DAY and test sequences from the dataset of [29]. The test set contains around 700 mhGCTs, similar to the training set. The results are slightly worse, but confusions again mostly occur with the two closest neighboring classes, as Figs. 8(c,d right) illustrate. The obtained results are remarkable considering the fact that we use only noisy depth information.

Pose Estimation. Finally, we apply our mhGCT feature to the challenging problem of body posture estimation in outdoor settings. The goal here is to infer the position of a person's body parts (such as torso, (left/right) shoulder, upper-arm, lower-arm, *etc*.). In order to train our classifiers we make use of synthetic training data (generated using the software Poser [34]), as shown in Fig. 9, allowing us to obtain precise ground-truth joint locations. We used three different person models that were rendered from a number of different camera locations while being animated with walking motions from MoCap data. In each frame we obtained a point cloud of the person and the corresponding joint locations in 3D. For this task, we split a person's trajectory now into 12-frames tracklets and each tracklet is split into three temporal bins from which we generate mhGCT features as before. These three mhGCTs are then simply concatenated forming a new feature which captures information about a person's gait phase. Each training example is labeled with the pose from the last frame of the 12-frame tracklet.

In order to test the performance of our approach for this task, we recorded a new dataset using a stereo camera setup where persons were walking in front of the camera, as shown in Fig. 10. For each frame, we annotated the visible person's joints in 2D. In total we annotated over 520 frames. For each annotated frame, we extract 12-frame mhGCTs as before. Then we apply the KNNCP classifier to estimate the body pose from the nearest neighbor in mhGCT space. For visualization, we align the retrieved skeleton of the nearest neighbor to the GCT's 3D center position and backproject each joint to the image. Some qualitative results are shown in Fig. 10.

The results are promising, showing that we often obtain a precise joint location estimate. In order to evaluate the performance quantitatively, we use the well-established PCP measure for pose estimation proposed by [33], which assesses the **P**ercentage of **C**orrectly labeled body **P**arts. An inferred body part is assumed as being correct if its joints (endpoints) lie within a fraction of the length (PCP-threshold) of the ground-truth segment from their annotated location. In Fig. 9 we show the performance while varying the PCP-threshold for the full-body pose, as well as for each single body part. Considering the fact that we can only partially observe the objects, where in many cases several parts are not visible at all, we obtain very good performance. Over 50% of the body part locations can be estimated correctly (with PCP-threshold 0.5). Expectedly, we have some problems classifying lower legs and lower arms, which can be easily explained by the fact that these are the parts which undergo the strongest articulations. Since we accumulate shape information over time, relying on median distances over a time window, we slightly underestimate the articulation of these body parts.

VI. CONCLUSION

We have presented a novel object centric, fixed-dimensional 3D shape representation which enables a robust matching of partially observed 3D shapes observed from RGB-D data. We have proposed different distance functions and kernels for the matching task and have compared their matching performance in a series of quantitative experiments. In order to demonstrate the wide applicability of our proposed 3D representation, we applied it to a variety of challenging vision tasks such as multi-class classification, person orientation estimation, and body posture estimation and showed that it can achieve good performance. We therefore foresee that it can find broad applicability in a variety of scenarios, providing a shape-based alternative to complement current appearance-based object descriptors such as DPM [31].

Acknowledgments: This work was funded, in parts, by ERC Starting Grant project CV-SUPER (ERC-2012-StG-307432) and EU project SPENCER (ICT-2011-600877).

REFERENCES

[1] D. Geronimo, A. Lopez, A. Sappa, and T. Graf, "Survey of Pedestrian Detection for Advanced Driver Assistance Systems," *PAMI*, vol. 32, no. 7, 2010.

[2] A. Geiger, P. Lenz, and R. Urtasun, "Are We Ready for Autonomous Driving? The KITTI Vision Benchmark Suite," in *CVPR*, 2012.

[3] M. Bajracharya, B. Moghaddam, A. Howard, S. Brennan, and L. Matthies, "Results from a Real-time Stereo-based Pedestrian Detection System on a Moving Vehicle," in *ICRA*, 2009.

[4] A. Teichman and S. Thrun, "Tracking-Based Semi-Supervised Learning," in *RSS*, 2011.

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

[6] D. Wang, I. Posner, and P. Newman, "What could move? finding cars, pedestrians and bicyclists in 3d laser data," in *ICRA*, 2012.

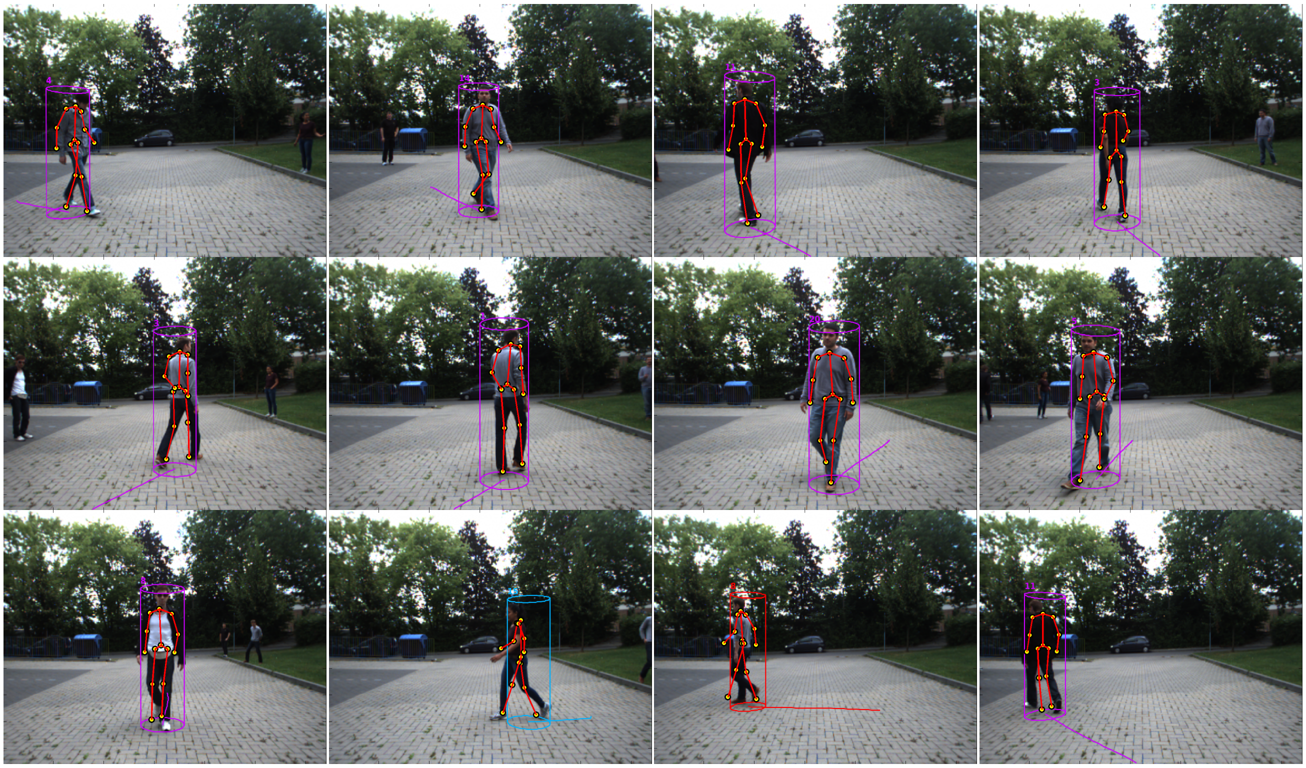


Fig. 10: Qualitative results for pose estimation using mhGCT features with a KNNCP classifier. As can be seen, the estimated pose is often correct. The last two images in the last row show failure cases, where the KNNCP classifier yields an imprecise pose estimate.

[7] S. Gammeter, A. Ess, T. Jaeggli, K. Schindler, B. Leibe, and L. Van Gool, "Articulated Multi-Body Tracking Under Egomotion," in *ECCV*, 2008.

[8] L. Pishchulin, A. Jain, M. Andriluka, T. Thormaehlen, and B. Schiele, "Articulated People Detection and Pose Estimation: Reshaping the Future," in *CVPR*, 2012.

[9] M. Bronstein and I. Kokkinos, "Scale-invariant heat kernel signatures for non-rigid shape recognition," in *CVPR*, 2010.

[10] A. Johnson and M. Hebert, "Using Spin Images for Efficient Object Recognition in Cluttered 3D Scenes," *PAMI*, vol. 21, no. 5, pp. 433–449, 1999.

[11] L. Spinello, K. O. Arras, R. Triebel, and R. Siegwart, "A Layered Approach to People Detection in 3D Range Data," in *AAAI*, 2010.

[12] T. Weise, T. Wismer, B. Leibe, and L. Van Gool, "Online Loop Closure for Real-time Interactive 3D Scanning," *CVIU*, vol. 115, no. 5, pp. 635–648, 2011.

[13] M. Krainin, P. Henry, X. Ren, and D. Fox, "Manipulator and Object Tracking for In-Hand 3D Object Modeling," *International Journal of Robotics Research (IJRR)*, vol. 30, no. 9, pp. 1311–1327, 2011.

[14] L. Spinello, M. Luber, and K. O. Arras, "Tracking People in 3D Using a Bottom-Up Top-Down Detector," in *ICRA*, 2011.

[15] M. Everingham, L. V. Gool, C. Williams, J. Winn, and A. Zisserman., "The PASCAL Visual Object Classes (VOC) challenge," *IJCV*, vol. 88, no. 14, 2010.

[16] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *CVPR*, 2005.

[17] G. Csurka, C. Dance, L. Fan, J. Willarnowski, and C. Bray, "Visual Categorization with Bags of Keypoints," in *ECCV'04 Workshop on Stat. Learn. in Comp. Vis.*, 2004.

[18] A. F. Bobick and J. W. Davis, "The recognition of human movement using temporal templates," *PAMI*, vol. 23, no. 3, 2001.

[19] D. Weinland, R. Ronfard, and E. Boyer, "Free viewpoint action recognition using motion history volumes," *CVIU*, vol. 104, no. 2, 2006.

[20] R. Fergus, A. Zisserman, and P. Perona, "Object class recognition by unsupervised scale-invariant learning," in *CVPR*, 2003.

[21] B. Leibe, A. Leonardis, and B. Schiele, "Robust Object Detection with Interleaved Categorization and Segmentation," *IJCV*, vol. 77, no. 1-3, 2008.

[22] L. Spinello and K. O. Arras, "People Detection in RGB-D Data," in *IROS*, 2011.

[23] S. Tang, X. Wang, T. Han, J. Keller, Z. He, M. Skubic, and S. Lao, "Histogram of Oriented Normal Vectors for Object Recognition with a Depth Sensor," in *ACCV*, 2012.

[24] T. Windheuser, U. Schlickewei, F. R. Schmidt, and D. Cremers, "Geometrically consistent elastic matching of 3d shapes: A linear programming solution," in *ICCV*, 2011.

[25] E. Rodola, T. Harada, Y. Kuniyoshi, and D. Cremers, "Efficient shape matching using vector extrapolation," in *BMVC*, 2013.

[26] D. Baltieri, R. Vezzani, R. Cucchiara, A. Utasi, C. Benedek, and T. Szirani, "Multi-View People Surveillance Using 3D Information," in *ICCV'11 Workshop on Visual Surveillance*, 2011.

[27] A. Teichman, J. Levinson, and S. Thrun, "Towards 3D Object Recognition via Classification of Arbitrary Object Tracks," in *ICRA*, 2011.

[28] J. Shin, R. Triebel, and R. Siegwart, "Unsupervised 3d object discovery and categorization for mobile robots," in *ISRR*, 2011.

[29] T. Baumgartner, D. Mitzel, and B. Leibe, "Tracking People and Their Objects," in *CVPR*, 2013.

[30] P. J. Besl and H. D. Mckay, "A Method for Registration of 3-D Shapes," *PAMI*, vol. 14, no. 2, 1992.

[31] P. Felzenszwalb, B. Girshick, D. McAllester, and D. Ramanan, "Object Detection with Discriminatively Trained Part-Based Models," *PAMI*, vol. 32, no. 9, 2010.

[32] A. Ess, B. Leibe, K. Schindler, and L. Van Gool, "Robust Multi-Person Tracking from a Mobile Platform," *PAMI*, vol. 31, no. 10, 2009.

[33] M. Eichner and V. Ferrari, "Human pose co-estimation and applications," *PAMI*, vol. 34, no. 11, 2012.

[34] SmithMicro, "Poser Pro 2012," http://my.smithmicro.com/poser-3d-animation-software.html.