Probability Occupancy Maps for Occluded Depth Images

Timur Bagautdinov[1], François Fleuret[2], and Pascal Fua[1]

[1]École Polytechnique Fédérale de Lausanne (EPFL), Switzerland
[2]IDIAP Research Institute, Switzerland
{timur.bagautdinov, pascal.fua}@epfl.ch, francois.fleuret@idiap.ch

Abstract

We propose a novel approach to computing the probabilities of presence of multiple and potentially occluding objects in a scene from a single depth map. To this end, we use a generative model that predicts the distribution of depth images that would be produced if the probabilities of presence were known and then to optimize them so that this distribution explains observed evidence as closely as possible.

This allows us to exploit very effectively the available evidence and outperform state-of-the-art methods without requiring large amounts of data, or without using the RGB signal that modern RGB-D sensors also provide.

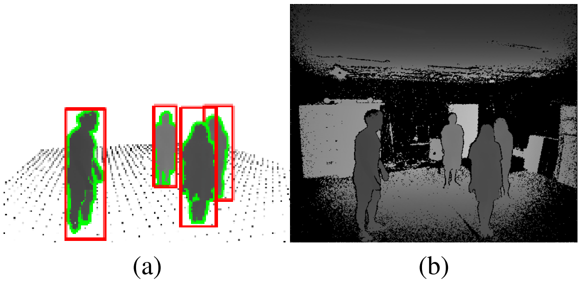


Figure 1. DPOM: generative model for depth maps. (a) Objects can be thought of as boxes, and images of objects are outlines within their rectangular projections. (b) Background is modeled explicitly: for each pixel there is a probability distribution, whose parameters are estimated from a set of background images.

1. Introduction

The advent of the original Kinect camera [12] and its sucessors has sparked a tremendous regain of interest for RGB-D imagers, which were formerly perceived as being either cumbersone or expensive. They have been used with great success for motion capture [17, 18] and are becoming increasingly popular for people detection in robotics applications [14, 19, 15, 10]. However, the former requires the algorithms to be trained on very large training databases, which may not always be easy to create, to achieve the desired level of performance while the latter usually do not make provisions for the fact that people may occlude each other. This results in failures such as those depicted by Fig. 2.

In this paper, we propose an approach that relies on a generative model to evaluate the probability of target objects being present in the scene while explicitly accounting for occlusions, which prevents such failures. It is inspired by an earlier approach to estimating these probabilities from background subtraction results from multiple cameras with overlapping fields of view [9]. Here, we use instead a single depth-map and approximate probabilities of occupancy at separate locations by choosing these probabilities so that the lower bound on the model likelihood is maximized. In contrast to many other approaches, ours does statistical reasoning jointly, *i.e.* knowledge about one piece of image evidence helps us to reason about the rest. This allows in particular to properly infer the presence of a severely occluded target from the presence of a small fragment. This process is depicted by Fig. 1.

We will demonstrate that our approach outperforms [15, 19, 18] for people detection purposes while using only the depth image, whereas these other approaches also require either the use of the RGB image and an additional classifier or extensive training. Furthermore, because we do not require training, it took us very little additional effort to also detect a completely different kind of objects, that is, flying drones potentially occluding each other.

2. Related work

There is an immense body of literature on people detection from regular images such as [2, 8, 4] to name but a few. However, most algorithms rely on features that are not necessarily present in depth images and are only rarely

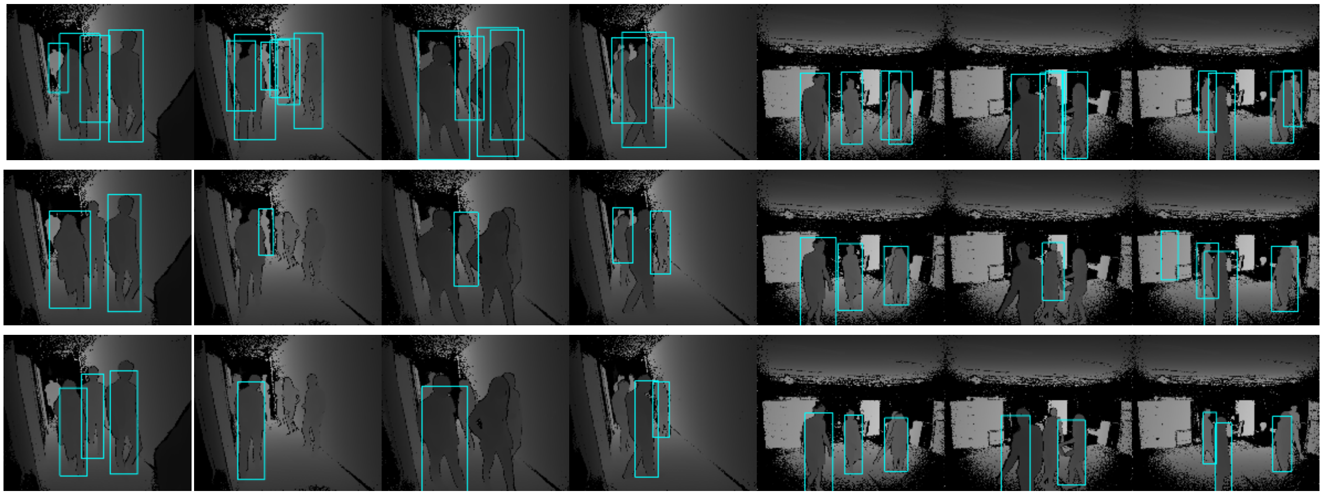


Figure 2. Situations in which we outperform state-of-the-art methods. Our approach (top) correctly detects most of the people including those that are severely occluded, whereas [15] (middle) and [13] (bottom) fail to do so.

designed to be robust to occlusions.

As far as using depth images is concerned, an impressive success was the original Kinect algorithm [17, 18] that could not only detect people but also estimate their 3D poses. It has since been improved and is included in the latest Kinect for Windows SDK [13]. This constitutes an extremely strong baseline against which we will compare our algorithm to show that our approach to occlusion handling does boost performance when people hide each other.

One of the reasons why the Kinect algorithm [17, 18] works so well is that it relies on decision forests that have been trained on huge datasets of synthetically generated data, which would make it non trivial to extend it to other categories of objects as we do for drones in this paper.

Among the recent approaches that do not require such extensive training databases are those proposed in [19, 15, 10], which we briefly describe below and will also use as baselines in our result section.

In [19], the authors introduce a descriptor called the Histogram of Oriented Depths (HOD) that extends the HOG descriptor [2]. They train two separate SVMs, one for HOG features for RGB data and the other on HOD features for depth images, and combine their scores.

In [15], a complete framework for tracking people in RGB-D data is described. Detection comprises two steps: hierarchical-clustering of the depth maps and HOG-based RGB detection. The clustering step involves finding top-level clusters in a depth image and then applying heuristics to detect people's heads to produce more fine-grained sub-clusters. The RGB detector, which is based on an improved version of the HOG features [5] and trained on the INRIA Person dataset [2], is then applied to the corresponding parts of the RGB image. The code is available in the Point Cloud Library [16] and we used it in our experiments.

In [10], two detectors are also used, a depth-based one for people at close range and a color-based one for those farther afield. The depth detector relies on template-matching followed by non-maxima suppression applied to regions of interests which are extracted using 3D point cloud clustering. Specifically, a 2D histogram is built by projecting all 3D points that could belong to objects on the ground plane and then finding clusters in that histogram. The RGB detector is a HOG-based detector with additional geometric constraints to reduce the search-space. This approach is very similar to that of [15], the main differences being the way RGB data is handled. Since this is not the main focus of our work and since the code that implements [15] runs on standard hardware, whereas that of [10] requires a modern GPU to use the complete RGB-D signal, we used the former as a representative of this class of techniques in our experimental evaluation.

To summarize: approaches discussed above typically do not perform occlusion reasoning explicitly, and mostly rely on heuristics when handling depth signals, which in many cases can provide reasonable results, but sometimes can lead to failures that are hard to interpret and predict, such as ones depicted in Fig. 2.

There is also a number of approaches related to ours [11, 7] in that they also apply generative modeling and variational inference to vision problems. However, to the best of our knowledge, they are focusing on very different problems, such as e.g. learning medium-level representations of images [11] or learning natural scene categories [7], whereas our goal is to estimate location of multiple occluding objects in the environment.

3. Approach

As discussed in the previous section, given a depth map of a scene featuring several people who are not occluding each other, state-of-the-art methods [15, 19, 13] do a good job of detecting them. However, these techniques do not perform the detection of all the targets jointly. Consequently, they can not re-assess properly the presence of a certain target, given evidence *and* the presence of occluding targets. More simply: a fragment of a target T in an empty room is a poor evidence of the presence of T. However, the presence of the same fragment when it is known that another target T' is present and hides the rest of T is a good evidence of the presence of T. Moreover, some of these methods [15, 10] rely on heuristics that sometimes result in failures even in simple cases. Fig. 2 depicts both situations.

A similar problem arises when attempting to detect people on the basis of background subtraction results from multiple cameras with overlapping fields of view. It was addressed in [9] by using a generative model for background subtraction images. Namely, people were represented as boxes that project to rectangles in individual views. The algorithm was then estimating people locations on the discretized ground plane, such that the image synthesized according to the generative model matched the background subtraction results as well as possible in all the views. We will refer to this approach as POM and it is depicted by the top row of Fig. 1. The strength of POM is that occlusions are naturally handled by the fact that rectangles corresponding to people further away from the camera are hidden by those corresponding to people that are closer.

Here, we also advocate the use of such a generative model to handle occlusions, but one designed to synthesize depth maps instead of binary images, as illustrated by the bottom row of Fig. 1. We will refer to this approach as DPOM.

In our model, we consider a finite number of locations on the ground. An object of interest, located at one of these, is represented by a flat free-shape inside a rectangular bounding box, as demonstrated in the Fig. 1(d). In practice, with each location k we therefore associate two random variables. The first is a Boolean X_k that denotes the presence or absence of the object at location k. The second, a Boolean mask $\boldsymbol{M}_k$, represents the 2D contour of that object and is intended to improve the fit of the generative model to the data. We model the measured depths at each pixel in the image as conditionally independent given these variables, and distributed around the depth of the closest object, or according to the background distribution if no object is present.

Given this model, we estimate the $\boldsymbol{M}_k$ through a segmentation procedure, and turn the estimation of the probabilities of presence $P(X_k|\boldsymbol{Z}, \boldsymbol{M})$ into a Bayesian inference problem as described formally in Section 4. Intuitively, what it allows us to do is predict the distribution of depth images that would be produced if the probabilities of presence were known and then to optimize them so that this distribution is centered around the observed one.

The introduction of the shape latent variables $\boldsymbol{M}_k$ leads to a better fit between the observed signal and the model, which is critical given that we exploit a single camera view. The standard POM algorithm achieves target localization through triangulation, using two or more cameras: even if the correct location of a target does not correspond to the best match in a individual view – in particular along the axis toward the camera – it is enforced through consistency in the other views which have non-parallel camera axis. In DPOM, since we use a single view signal, the accuracy along the axis of view is entirely due to the precision of the model.

4. Formulation

In this section, we formally describe our generative model, explain how we do inference on it to get the probabilities of presence, and then describe some implementation aspects.

4.1. Generative model

We introduce first some notations, which are summarized with those of other sections in Table 1.

Let $\boldsymbol{Z} \in \mathcal{Z}^{|\mathcal{L}|}$ denote the depth map, with $\mathcal{L} = \{1, \ldots, N\}$ being the set of all pixels, and $\mathcal{Z}$ being a set of all possible depth values. Let $z^{\infty} \in \mathcal{Z}$ be a special value of depth encoding situation when no depth is observed (depicted in black in Fig. 1(c)).

Let us assume we have discretized the ground plane into possible object locations $\mathcal{K} = \{1, \ldots, K\}$, as depicted in Fig.1(d). We introduce hidden binary variables $\boldsymbol{X} = \{X_k, \forall k \in \mathcal{K}\}$ with $X_k = 1$ if location k is occupied, 0 otherwise. Furthermore, for each location k, we have a corresponding crude rectangular representation of an object, which we call *silhouette* $\mathcal{S}_k \subset \mathcal{L}$. For each pixel of a silhouette we specify a corresponding depth distribution over $z \in \mathcal{Z}$:

$$\theta_{ki}(z), \forall i \in \mathcal{S}_k \ , \qquad (1)$$

where the specific shape and parameterization of the distribution θ_{ki} depends very much on the sensor used for depth acquisition. We will introduce a more detailed model in Section 4.3.

The silhouette specifies a very simplistic rectangular shape, whereas most of the objects have much more complex outlines. To encounter for that, we introduce *segmentation masks* $\boldsymbol{M}_k \subset \mathcal{L}, \forall k \in \mathcal{K}$. If $i \in \boldsymbol{M}_k$, it means that the pixel $i \in \mathcal{S}_k$ actually belongs to the object outline at k-th location (see Fig. 1(d)).

Finally, some pixels belong to the background, rather than objects. In particular, when there are no objects in the

$\mathcal{K}$	set of all locations
$\mathcal{L}$	set of all pixels
$\mathcal{Z}$	set of all possible depth values
X_k	binary occupancy variable for a location k
Z_i	observed depth value variable at pixel i
z^∞	special value for when no depth is observed
M_{ki}	segmentation mask at pixel i for location k
$\mathcal{S}_k$	object silhouette for k-th location
$\lvert\mathcal{S}_k\rvert$	number of pixels in the k-th silhouette
$\langle\cdot\rangle_p$	expectation w.r.t. a distribution p
$\sigma(x)$	sigmoid function $(1+e^{-x})^{-1}$
ρ_k	approximate posterior $Q(X_k = 1)$
π^∞	probability of observing z^∞
π°	probability of observing an outlier
$\Delta_{l,i}$	$\log\theta_{l,i}(z)$, $l \in \mathcal{K} \cup \{\text{bg}\}$

Table 1. Notations used in this paper

scene, we observe only background. Thus, for each pixel of the depth map we have a corresponding background distribution over $z \in \mathcal{Z}$:

$$\theta_{\text{bg},i}(z), \forall i \in \mathcal{L} \; . \tag{2}$$

Our ultimate goal is to estimate the posterior distribution $P(\boldsymbol{X}|\boldsymbol{Z},\boldsymbol{M})$ given the depth image $\boldsymbol{Z}$ and segmentation masks $\boldsymbol{M}$. To do that, we introduce a generative model $P(\boldsymbol{Z}|\boldsymbol{X},\boldsymbol{M})$ and then apply Bayes' rule.

First, we assume that the prior occupancies $\boldsymbol{X}$ are independent from each other, *i.e.*:

$$P(\boldsymbol{X}) = \prod_{k\in\mathcal{K}} P(X_k) \; , \tag{3}$$

which intuitively means that objects occupy locations regardless of the presence of other objects.

Second, we assume that the observations for individual pixels $Z_i, \forall i \in \mathcal{L}$ are *conditionally* independent, *i.e.* given $\boldsymbol{X}$ and $\boldsymbol{M}$.

We can now synthesize depth Z_i for each pixel $i \in \mathcal{L}$ of the depth image. We select the model corresponding to a silhouette which is present ($X_k = 1$), contains the pixel ($i \in \mathcal{S}_k$), belongs to the object segmentation mask ($i \in \boldsymbol{M}_k$), and is the closest to the camera. It is, of course, also possible that we observe background at a specific pixel. This happens either if no silhouettes are present that has a model for this pixel ($i \notin \boldsymbol{M}_k, \forall k \in \mathcal{K}$), or if all of the silhouettes are further away from the camera (object is occluded by a part of the background). Note, that we assume that all the depth distributions $\theta_{l,i}(z), \forall l \in \mathcal{K} \cup \{\text{bg}\}$ are ordered w.r.t. the distance to the camera. In practice, we order them by their mean value $\langle\theta_{l,i}(z|z \neq z^\infty)\rangle$. More formally, $\forall i \in \mathcal{L}$:

$$\begin{aligned} l^* &= \operatorname*{arg\,min}_{l:\{X_l=1, i\in\boldsymbol{M}_l, l\in\mathcal{K}\}\cup\{\text{bg}\}} \langle\theta_{l,i}(z|z \neq z^\infty)\rangle, && (4) \\ Z_i &\sim \theta_{l^*,i}(z). && (5) \end{aligned}$$

4.2. Inference

Even under the assumptions of our generative model, computing $P(\boldsymbol{X}|\boldsymbol{Z},\boldsymbol{M})$ directly is still computationally untractable, due to the dimensionality of $\boldsymbol{X}, \boldsymbol{M}$ and $\boldsymbol{Z}$. To solve this, we first assume that $\boldsymbol{M}$ is given, by computing it as described in Section 4.4, and then derive a variational approximation for $P(\boldsymbol{X}|\boldsymbol{Z},\boldsymbol{M})$. Let us introduce the following approximate posterior distribution over hidden variables $\boldsymbol{X}$:

$$Q(\boldsymbol{X}) = \prod_{k\in\mathcal{K}} Q(X_k) \tag{6}$$

where each $Q(X_k)$ is a Bernoulli distribution.

We then minimize the KL-divergence between $Q(\boldsymbol{X})$ and $P(\boldsymbol{X}|\boldsymbol{Z},\boldsymbol{M})$, which has been shown [20] to be equivalent to getting an updated approximate posterior $Q^*(X_k)$ for each X_k:

$$Q^*(X_k) \propto \exp\langle\log P(\boldsymbol{Z},\boldsymbol{M},\boldsymbol{X})\rangle_{Q(\boldsymbol{X}\setminus X_k)} \tag{7}$$

where $\langle\cdot\rangle_{Q(\boldsymbol{X}\setminus X_k)}$ denote an expectation w.r.t. all the variables *except* for X_k.

Knowing that X_k is a Bernoulli variable, we can get the following update rule for $\rho_k = Q(X_k = 1)$:

$$\begin{aligned} \rho_k = \sigma(\quad & \langle\log P(\boldsymbol{Z},\boldsymbol{X},\boldsymbol{M}|X_k = 0)\rangle_{Q(\boldsymbol{X}/X_k)} - \\ & \langle\log P(\boldsymbol{Z},\boldsymbol{X},\boldsymbol{M}|X_k = 1)\rangle_{Q(\boldsymbol{X}/X_k)}) \; , \end{aligned} \tag{8}$$

where $\sigma(x) = (1+e^{-x})^{-1}$ is a sigmoid function.

We want to substitute our generative model $P(\boldsymbol{Z}|\boldsymbol{X},\boldsymbol{M})$ into (8). Let's first introduce some notations. Let $\Delta_{k,i} = \log\theta_{ki}(z)$, and $\Delta_{\text{bg},i} = \log\theta_{\text{bg},i}(z)$. Let us denote the prior of occupancy $\epsilon = P(X_k = 1)$, assuming it is identical $\forall k \in \mathcal{K}$. If we assume, without loss of generality, that silhouettes are sorted w.r.t. the distance to the sensor, then the probability of all silhouettes $\mathcal{S}_l : l < k$ being absent at a pixel $i \in \mathcal{L}$ will be:

$$\tau_{ki} = \prod_{l<k, i\in\boldsymbol{M}_l} (1 - \rho_l) \tag{9}$$

which can be considered as a transparency at a certain pixel.

If we now substitute our model into (7), and evaluate expectation w.r.t. the current estimate of $Q(\boldsymbol{X})$ we will get the following update for $\rho_k, \forall k \in \mathcal{K}$:

$$\begin{aligned} & \rho_k = \sigma(\log \tfrac{1-\epsilon}{\epsilon} - \\ & \textstyle\sum_{i\in\boldsymbol{M}_k} \tau_{k-1,i}\Delta_{k,i} - \\ & \textstyle\sum_{i\in\boldsymbol{M}_k} \tfrac{1}{1-\rho_k} (\sum_{l>k, i\in\boldsymbol{M}_l} \tau_{l-1,i}\rho_l\Delta_{l,i} + \tau_{|\mathcal{K}|,i}\Delta_{\text{bg},i})) \end{aligned} \tag{10}$$

4.3. Numerical model

Up until now, we have not specified exactly our pixel depth distributions $\theta_{ki}(z)$ and $\theta_{\text{bg},i}(z)$. The shape of those distributions can vary depending on the type of the sensor, but here we describe one that worked well for both versions of Kinect.

We assume that the distribution has two components: a Dirac and a robust Gaussian. The former is necessary, since in some cases sensor is incapable of producing a reasonable estimate of the depth, and reports a special value, z^{∞}. The robust Gaussian component is simply a mixture of a Gaussian ($\mathcal{N}$) and a uniform distribution ($\mathcal{U}$), which takes care of possible outliers. Thus, each $\theta_{l,i}$ has four parameters: π^{∞}_{li}, a probability of observing z^{∞}, π°_{li}, a probability of observing an outlier, and μ_{li}, σ^2_{li}, which are the mean and the variance of the Gaussian component.

Finally, $\forall l \in \mathcal{K} \cup \{\text{bg}\}$:

$$\theta_{li}(z|z \neq z^{\infty}) = \pi^{\circ}_{li}\mathcal{U}(z) + (1 - \pi^{\circ}_{li})\mathcal{N}(z|\mu_{li}, \sigma^2_{li}) \tag{11}$$

Particularly, the mean for object pixel distributions $\theta_{ki}(z), \forall k \in \mathcal{K}$ is a value one would observe if object was a flat surface facing the camera. The variance is fixed for an object type, e.g. for people we use a fixed value $\sigma^2_{ki} = 100, \forall k \in \mathcal{K}$. For background pixel distributions, we estimate all the parameters from a set of background frames.

4.4. Computing M

As already mentioned, in theory we also could have obtained an approximation to posterior $P(\boldsymbol{M}|\boldsymbol{Z})$, but it would be rather expensive computationally. Thus, given the observed depth map $\boldsymbol{z} \in \mathcal{Z}^{|\mathcal{L}|}$, we apply the following simple procedure to obtain a point estimate for $\boldsymbol{M}_k, \forall k \in \mathcal{K}$:

$$\boldsymbol{M}_k = \{i : \pi^{\circ}\mathcal{U}(z_i) > (1 - \pi^{\circ})\mathcal{N}(z_i|\mu_{ki}, \sigma^2), z_i \neq z^{\infty}\} \tag{12}$$

which in words means that we consider a pixel i to be a part of the object if depth value is observed and not considered an outlier under the model (11).

4.5. Implementation details

In reality, we have noticed that the update (10) is not very robust. Namely, the predicted ρ_k are very peaky, and sometimes for a relatively small amount of evidence, the confidence of occupancy is very high. Since the depth maps are relatively noisy, it can lead to a large number of false positives. To avoid that, we use soft thresholding based on the amount of pixel evidence:

$$\rho^*_k = \sigma\left(\alpha\frac{\sum_{i \in \boldsymbol{M}_k} \tau_{ki}}{|\mathcal{S}_k|} + \beta\right) \cdot \rho_k\ , \tag{13}$$

where, α and β are sigmoid parameters. They were set to disable those estimates that have very little evidence w.r.t. the size of our crude rectangular silhouette (we are using $\alpha = -100$, $\beta = 8$).

5. Experimental evaluation

In this section, we first report our people detection results and compare them against those of the baseline methods introduced in Section 2. We then show that our approach can be easily adapted to a very different detection problem, namely detecting flying drones that may occlude each other. We supply the corresponding videos as supplementary material.

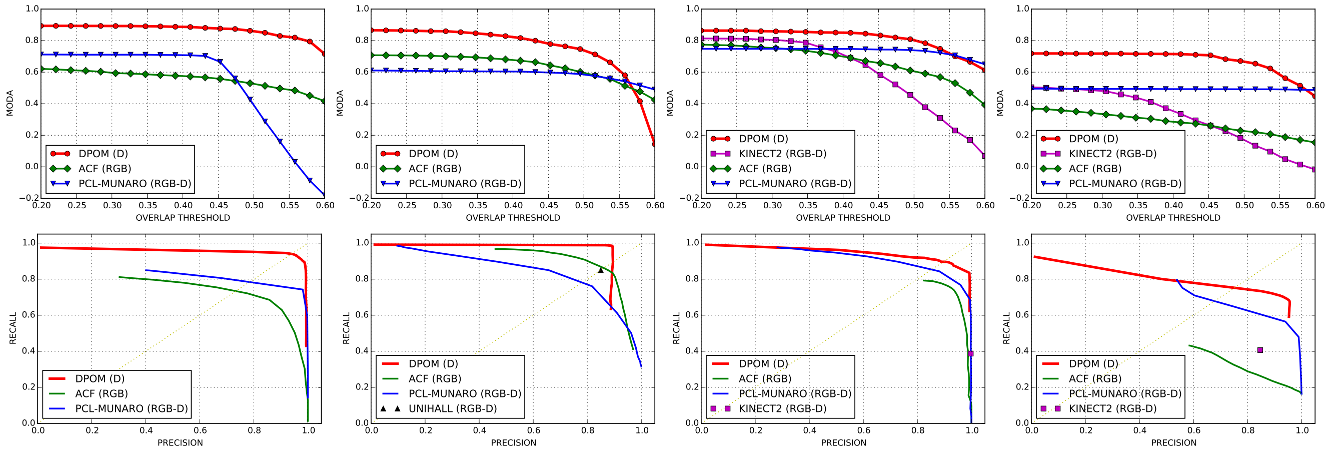
5.1. Datasets

There are many well-known and publicly available RGB datasets for testing pedestrian detection algorithms, such as those of [2, 6]. For RGB-D data, there are far fewer.

The Kinect Tracking Precision dataset (`KTP`) presented in [15] contains several sequences of at most 5 people walking in a small lab environment. They were recorded by a depth camera mounted on a robot platform and we use here the only one that was filmed while the camera was static. Authors provide ground truth locations of the individuals both on the image plane, and on the ground plane. Unfortunately, the quality of the ground truth for the ground plane is limited, due to the poor quality registration of the depth sensor location to the environment. In order to fix this, we made an effort and manually specified points corresponding to individuals on the depth maps, then projected them on the ground plane, and took an average to get a single point representing person location. This introduces a small bias as we only observe the outer surface of the person but any motion capture system would have similar issues.

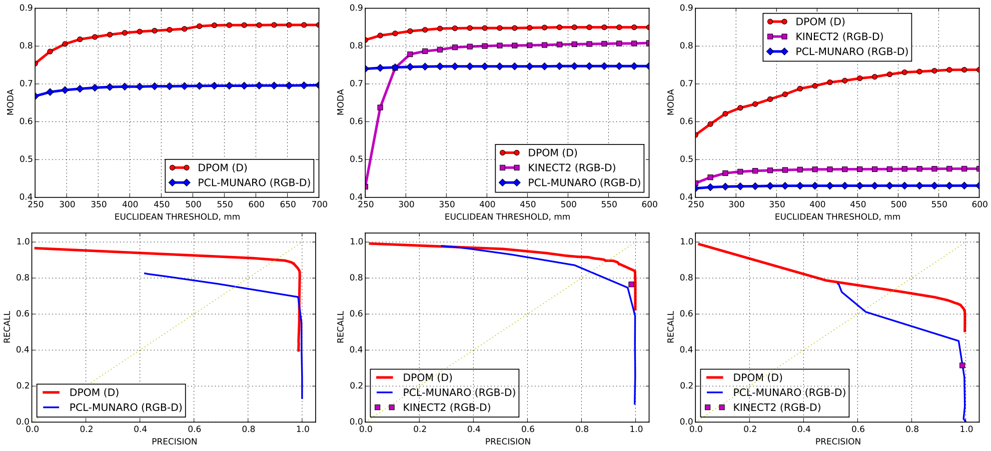
In [19, 14], the authors report their results in a dataset containing about 4500 RGB-D images recorded in a university hall from a three statically mounted Kinects (`UNIHALL`). Unfortunately, there is no ground plane ground truth available, thus we only report results for image plane. To compare to their results, we follow evaluation procedure described in [19], that is, without penalizing approaches for not detecting occluded or hidden people. We also report our performance for the full dataset separately.

There are no publicly available datasets for multiple people tracking using the latest Kinect SDK [13] and we therefore created two of them ourselves. The first one (`OURS-LAB`) contains around 1000 RGB-D frames with around 3000 annotated people instances. There are at most 4 people who are mostly facing the camera, presumably the scenario for which the Kinect software was fine-tuned. The second one (`OURS-CORRIDOR`) was recorded in a more realistic environment, a corridor in a university building. It contains over 3000 frames with approximately 20000 individual people instances. Sample frames together with our detection results are shown in Fig. 6.



(a) KTP (b) UNIHALL (c) OURS-LAB (d) OURS-CORRIDOR

Figure 3. MODA (top) and precision-recall (bottom) for image plane ground truth. For each algorithm, the label indicates what type of information it uses: D - depth only, RGB - color only, RGB-D - both depth and color.



(a) KTP (b) OURS-LAB (c) OURS-CORRIDOR

Figure 4. MODA (top) and precision-recall (bottom) for ground plane ground truth. For each algorithm we mention the type of information that it uses: D - only depth, RGB - only color, RGB-D - both.

The ground truth for the ground plane locations was obtained similarly to what has been described for KTP dataset, that is, for each person instance, we specified the points on the depth maps, projected them on the ground plane, and computed the average to get a location. In order to get individuals' bounding boxes in the image plane, for every target we compute the average of the projections of the marked pixels onto the image plane, and add a bounding box centered on it and sized according to their average depth.

Some approaches, including ours, require knowing both extrinsic and intrinsic camera parameters. The intrinsics were fixed for the specific Kinect we used. To compute the extrinsics, we manually specified the region of the depth map corresponding to the ground plane and then estimated the transformation from camera space to that plane.

5.2. Baselines

We use the following baselines for comparison purposes:

- KINECT2 - the results obtained from the human pose estimation of the latest Kinect for Windows SDK [13]. It is not publicly known what specific algorithm is used. However in [17], the authors report that their algorithm is at the core of the human pose estimation for the older version of the Kinect software. For undisclosed reasons, the framework supports tracking up until six people, with the working depth range limited to 4.5 meters. To ensure fairness, we kept this restrictions in mind when using the OURS-LAB and

OURS-CORRIDOR datasets. We do not penalize algorithms for not detecting more than 6 people or people who are further than 4.5 meters away.

- UNIHALL - RGB-D detector[19] based on HOG and HOD features. The code is not available and we therefore report only a single point on the precision-recall curve.

- PCL-MUNARO - RGB-D detector [15]. It uses modified HOG features [5] on regions extracted by depth segmentation. We used the implementation from the PCL library [16].

- ACF - RGB detector from [3], based on AdaBoost and aggregate channel features [4] to give a sense of what a state-of-the-art detector that does not use depth can do on these sequences.

5.3. Overall Performance

In Fig. 3, we report overall performance comparisons on the four datasets introduced in Section 5.1. For each one, we report results in two different ways. We plot both Multiple Objects Detection Accuracy (MODA) [1], as a function of bounding box overlap in the image plane, and also precision-recall curves. We made this choice to compare ourselves to authors who report image plane-only results. Here, precision-recall curves are shown for an overlap threshold of 0.4 as in [19], and additional curves are provided in the supplementary material. The MODA curves are computed for fixed detector confidence threshold, the best-performing one for each approach, except in the case of KINECT2 for which we have no way to set any threshold. Results for other detection thresholds are also available as supplementary material.

DPOM clearly outperforms all other approaches. This is true even though, as the KINECT2, we use *only* the depth information whereas the other algorithms also use the RGB information. Only for UNIHALL dataset, at overlap threshold above 0.55, does DPOM become slightly worse. This can be ascribed to the fact that we use a fixed-sized object model, and for this particular dataset and ground truth this size happens to be too small. This is not very crucial though, since at those values of overlap threshold, absolute performance of all the evaluated methods is rather low.

Note that KINECT2 performs much worse on the OURS-CORRIDOR sequence that in OURS-LAB one. It is very hard to know why exactly, because the specific algorithm being used is a trade-secret. Our best guess is that in OURS-CORRIDOR, the camera is slightly tilted and people do not appear to as being strictly vertical. If this interpretation were correct, it would illustrate the dangers of training an algorithm under specific assumptions that may prevent generalization. Another possible explanation is that some sequences start with people already present in the field of view, thus making it hard to use any kind of background subtraction. Whatever the case, the OURS-LAB sequence presents neither of these difficulties and DPOM still performs better.

In Fig. 4, we report MODA and precision-recall values computed in the ground plane instead of the image plane for the three methods for which it can be done. In ground-plane settings, we consider detection a match to the ground truth if it is within a certain Euclidean distance to it. The values of MODA are shown as a function of Euclidean thresholds, for a single best detection threshold for each algorithm. Precision-recall curves are plotted for a fixed Euclidean distance threshold of 500mm. The performance ordering stays essentially the same.

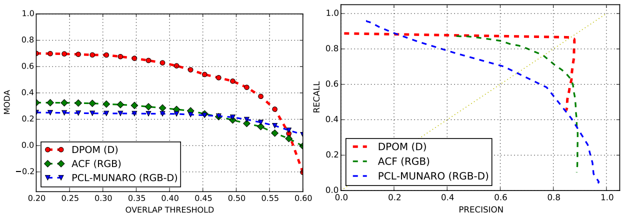


Figure 5. MODA (left) and precision-recall (right) for image plane ground truth for UNIHALL dataset. Approaches *were* penalized for not detecting occluded or hidden people.

Recall that the evaluation procedure we have used so far is that of [19] in which not detecting occluded or hidden people is *not penalized*. To demonstrate that this choice does not have a major impact on our conclusions, we plot in Fig. 5 equivalent precision-recall curves when they are penalized. As expected, the performance numbers are worse than those shown in Fig. 3 for all methods. However, the ranking is preserved and the performance drop is smaller for DPOM. This highlights once more its ability to deal with occlusions.

5.4. Drones

To demonstrate the versatility of our approach, we have also applied it to a completely different type of objects, that is, drones. Note that, for people, we estimated their locations on the discretized ground plane. For drones, we instead use a discretized 3D space, and our algorithm thus estimates occupancy probabilities for each discrete 3D location in that space.

We filmed two drones flying in a room, sometimes occluding each other and sometimes being hidden by furniture. As in our people sequences, we obtained ground truth by manually specifying points on the drones, and then computing the bounding cube. To determine whether a detection is a match, we use overlap in bounding cubes.

Since there are no canonical baseline approaches, we

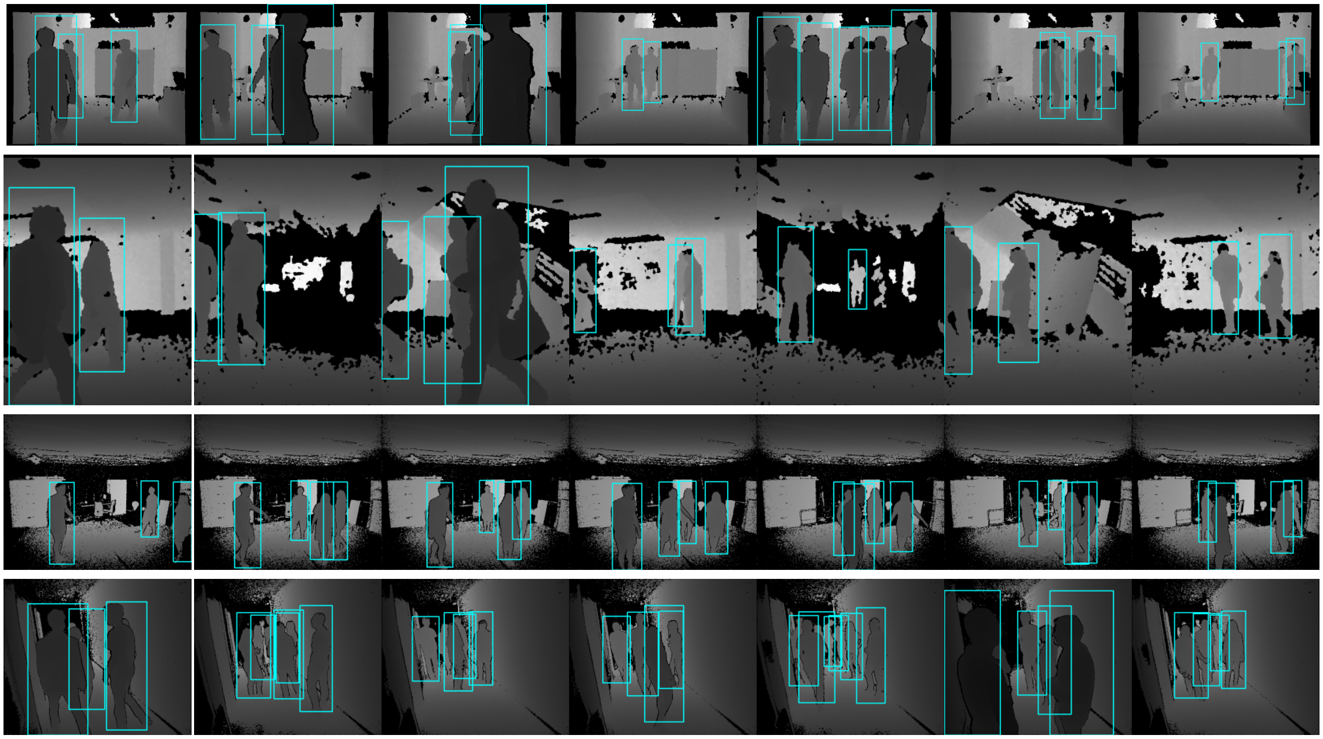
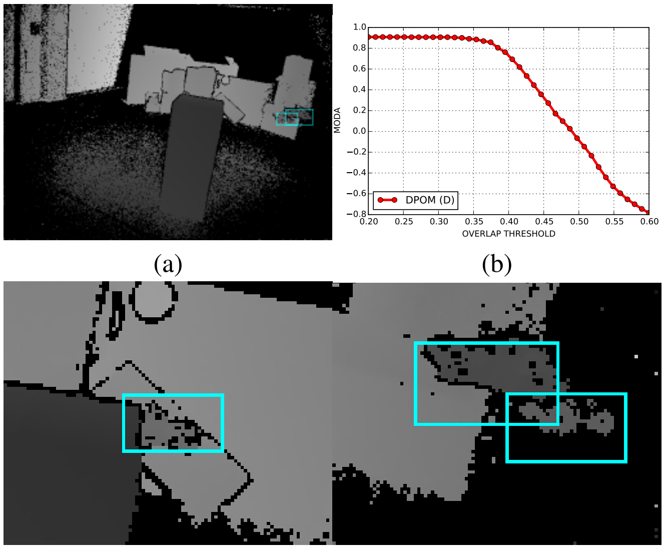


Figure 6. Sample detections for selected frames of our test datasets. From top to bottom, we show `KTP`, `UNIHALL`, `OURS-LAB`, and `OURS-CORRIDOR`.



(a) (b)

(c) (d)

Figure 7. Detection results for drones. (a) Sample detections. (b) MODA score. (c) Detection with background occlusions. (d) Detection with occlusions.

only report our own MODA values in Fig. 7. For overlap thresholds below 0.4, we obtain reasonable performance. For larger thresholds, the drop in performance is attributable to the fact that we discretize the 3D space, which means a relatively large localization error compared to the small size of the drones.

6. Discussion and future work

We have introduced a probabilistic approach to estimating occupancy maps given depth images. We have shown that it outperforms state-of-the-art approaches both on publicly available datasets and our own challenging sequences. Moreover, the approach is generic enough to be easily adapted to a completely different object type, which we demonstrated by using it for detecting drones.

However, one weak point of our approach is speed: currently, our implementation is not real-time, and takes several seconds to process a single depth frame on a 2.3GHz Intel CPU. This problem can be targeted using GPUs, since the bottleneck of our algorithm is iterating through the pixels. Another limitation, which is a consequence of using a rough generative model, is the lack of discriminative power. Our approach requires no training data but cannot distinguish between different object types as long as they fit our model well enough. Therefore, the next step would be to provide means to either combine our occupancy maps with the output of a discriminative classifier or to make object models more sophisticated, possibly by learning them from the data.

Acknowledgment

This work was supported in part by a Swiss National Science Foundation grant.

References

[1] K. Bernardin and R. Stiefelhagen. Evaluating Multiple Object Tracking Performance: the Clear Mot Metrics. *EURASIP Journal on Image and Video Processing*, 2008, 2008. 7

[2] N. Dalal and B. Triggs. Histograms of Oriented Gradients for Human Detection. In *Conference on Computer Vision and Pattern Recognition*, 2005. 1, 2, 5

[3] P. Dollár. Piotr's Computer Vision Matlab Toolbox (PMT). http://vision.ucsd.edu/~pdollar/toolbox/doc/index.html. 7

[4] P. Dollár, R. Appel, and W. Kienzle. Crosstalk Cascades for Frame-Rate Pedestrian Detection. In *European Conference on Computer Vision*, 2012. 1, 7

[5] P. Dollár, Z. Tu, P. Perona, and S. Belongie. Integral Channel Features. In *British Machine Vision Conference*, 2009. 2, 7

[6] P. Dollar, C. Wojek, B. Schiele, and P. Perona. Pedestrian Detection: A Benchmark. In *Conference on Computer Vision and Pattern Recognition*, June 2009. 5

[7] L. Fei-Fei and P. Perona. A Bayesian Hierarchical Model for Learning Natural Scene Categories. In *Conference on Computer Vision and Pattern Recognition*, 2005. 2

[8] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object Detection with Discriminatively Trained Part Based Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9), 2010. 1

[9] F. Fleuret, J. Berclaz, R. Lengagne, and P. Fua. Multi-Camera People Tracking with a Probabilistic Occupancy Map. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):267–282, February 2008. 1, 3

[10] O. Jafari, D. Mitzel, and B. Leibe. Real-Time RGB-D Based People Detection and Tracking for Mobile Robots and Head-Worn Cameras. In *International Conference on Robotics and Automation*, pages 5636–5643, 2014. 1, 2, 3

[11] N. Jojic and B. J. Frey. Learning flexible sprites in video layers. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–199. IEEE, 2001. 2

[12] Kinect Camera, 2010. http://www.xbox.com:80/en-us/kinect/. 1

[13] Kinect for Windows SDK 2.0, 2014. http://www.microsoft.com/en-us/kinectforwindows/. 2, 3, 5, 6

[14] M. Luber, L. Spinello, and K. Arras. People Tracking in Rgb-D Data with On-Line Boosted Target Models. In *International Conference on Intelligent Robots and Systems*, pages 3844–3849, 2011. 1, 5

[15] M. Munaro and E. Menegatti. Fast RGB-D People Tracking for Service Robots. *Autonomous Robots*, 37(3):227–242, 2014. 1, 2, 3, 5, 7

[16] R. B. Rusu and S. Cousins. 3D is here: Point Cloud Library (PCL). In *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 9-13 2011. 2, 7

[17] J. Shotton, A. Fitzgibbon, M. Cook, and A. Blake. Real-Time Human Pose Recognition in Parts from a Single Depth Image. In *Conference on Computer Vision and Pattern Recognition*, 2011. 1, 2, 6

[18] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, and R. Moore. Real-Time Human Pose Recognition in Parts from Single Depth Images. *Communications of the ACM*, 56(1):116–124, 2013. 1, 2

[19] L. Spinello and K. Arras. People Detection in RGB-D Data. In *International Conference on Intelligent Robots and Systems*, pages 3838–3843, 2011. 1, 2, 3, 5, 7

[20] J. M. Winn and C. M. Bishop. Variational message passing. In *Journal of Machine Learning Research*, pages 661–694, 2005. 4