Human Detection and Segmentation via Multi-view Consensus

Isinsu Katircioglu[1] * Helge Rhodin[2] Jörg Spörri[3] Mathieu Salzmann[1,4] Pascal Fua[1]
[1]EPFL, Lausanne, Switzerland [2]UBC, Vancouver, Canada
[3]Balgrist University Hospital, Zurich, Switzerland [4]ClearSpace SA, Lausanne, Switzerland
{firstname.lastname}@epfl.ch, rhodin@cs.ubc.ca, joerg.spoerri@balgrist.ch

Koh et al. [22] Yang et al. [53] Katircioglu et al. [20] Rhodin et al. [37] + [20] Ours

Figure 1: Leveraging multi-view consistency at training time to segment the salient object from *single* images at inference time and to outperform baselines exploiting temporal consistency [22], optical flow [53, 20] and novel view synthesis [37].

Abstract

Self-supervised detection and segmentation of foreground objects aims for accuracy without annotated training data. However, existing approaches predominantly rely on restrictive assumptions on appearance and motion.

For scenes with dynamic activities and camera motion, we propose a multi-camera framework in which geometric constraints are embedded in the form of multi-view consistency during training via coarse 3D localization in a voxel grid and fine-grained offset regression. In this manner, we learn a joint distribution of proposals over multiple views. At inference time, our method operates on single RGB images. We outperform state-of-the-art techniques both on images that visually depart from those of standard benchmarks and on those of the classical Human3.6M dataset.

1. Introduction

Robust detection and segmentation of moving people can now be achieved reliably in scenarios for which large amounts of annotated data are available. However, for less common activities, such as skiing, it remains challenging, because the required training databases do not exist. Self-supervised approaches [10, 22, 5, 6, 8, 9, 37, 53, 28, 3, 31] promise to address this problem. However, most of them depend on strong constraints, such as the target objects being seen against a static background, or rely on object localization and object-boundary detection networks pre-trained with supervision, which limits their applicability.

In this paper, we propose to remove these limitations by using a multi-camera setup for training purposes and explicitly encoding the 3D geometry of the scene. At inference time, our trained network can then handle single images and outperforms earlier techniques, as shown in Fig. 1. Our algorithm can be applied to any object as long as the two assumptions from [24] hold: foreground and background are distinguishable by color or texture; every part of the background must be visible more often than not.

Using several cameras complicates data acquisition but only in a limited way because both synchronization and calibration are well understood tasks for which off-the-shelf

*Work supported in part by the Swiss National Science Foundation.

solutions exist. In practice, for static cameras, this has to be dealt with only once before a filming session using well-known techniques [15, 12] and requires far less effort than manually annotating images. For moving cameras, SLAM methods are now robust enough to perform the calibration automatically and fast in the wild [58, 51]. Hence, there are many applications in which training with multiple cameras makes perfect sense, especially those with unusual activities for which large training databases are not available.

To leverage multi-view training data as weak supervision, we introduce the object proposal strategy depicted by Fig. 2. Candidate 2D bounding boxes are produced by a network that can be trained in an unsupervised fashion. They are used to vote into a 3D proposal grid, and multi-view geometry constraints are then imposed to align proposals from different views in a differentiable manner. To train the resulting network, we sample a 3D proposal, deconstruct and reconstruct the image in each view using the corresponding 2D bounding box, and compare the resulting resynthesized images to the original ones.

While our self-supervised learning strategy leverages multiple views during training, the resulting model can be used for detection and segmentation in monocular images acquired by moving cameras and featuring unknown backgrounds. Our contributions can be summarized as follows.

- We introduce a self-supervised end-to-end trainable object detection and segmentation approach that explicitly leverages 3D multi-view geometry as weak supervision during training.
- It comprises a 3D object proposal framework that enables to enforce prediction consistency across views without having to introduce additional loss terms.

To show that our approach can handle unusual activities and fast motion, we demonstrate it on the skiing dataset depicted by Fig. 1, captured by moving cameras, on a small dataset acquired using hand-held cameras, as well as on the more standard H36M dataset [18] acquired using fixed cameras. Note that our multi-view supervision differs from weak supervision in video object segmentation as it does not require any segmentation annotation. Hence, our method relates to self-supervised approaches. We show that the proposed multi-view training increases single-image accuracy performance at inference time, which allows us to outperform state-of-the-art single-view [22, 53, 9, 31, 20] and multi-view [37] approaches. Our code is publicly available at https://github.com/isinsukatircioglu/mvc.

2. Related Work

Salient object detection and segmentation is a long-standing problem in computer vision. In this section, we review the monocular and multi-view approaches that have been proposed to solve this task.

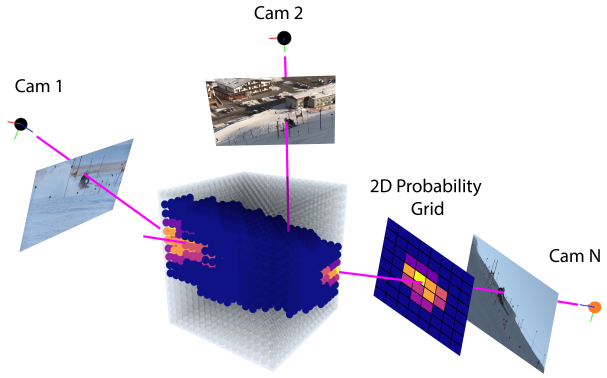


Figure 2: **3D Proposal Grid.** The consensus between individual views is found on a 3D voxel grid (black) as a combination of 2D probabilities projected on the voxels (rainbow colors). Once a coarse grid location is found, a fine offset is found via offset prediction and 3D triangulation (purple lines).

Single View Approaches. Most salient object detection and segmentation algorithms are fully-supervised [7, 16, 36, 42, 4, 27, 29, 40, 41, 55] and require large annotated datasets containing pairs of images and labels. Our goal is to train a purely self-supervised method without either segmentation or object bounding box annotations. Note that this differs from the so-called *unsupervised object segmentation* methods that leverage either domain-specific annotations during training but not at test time [35, 17, 19, 25, 26, 30, 47, 54, 49, 56, 57], or the label of the first frame at inference time [50]. We focus our discussion on self- and weakly-supervised methods with regard to the type of training data used.

As conventional methods relying on hand-crafted features, recent methods train deep neural networks for object detection [19, 52], optical flow estimation [45, 46], and object saliency [25] using motion and appearance related cues.

Motion-based methods [23, 33, 11, 21, 48, 14, 22, 43, 53] define the foreground object based on the region that moves differently from the rest of the scene, and they integrate this supervision through optical flow images and temporal consistency. [22] combines the flow information with the recurrence property of the primary object in an image sequence and identifies the matching segment tracks across frames by extracting ultrametric contour maps. Similar to our approach, [43] assumes that the foreground is harder to model than the background and while modeling the background by a low-dimensional linear basis, the image parts this model fails to explain are identified as the salient object. In contrast to [43], our method relies on the predictability of image patches from their spatial neighborhood using deep neural

networks, can handle complex background motion and does not require videos. Built upon [43], [9] trains an ensemble of networks, which comes at the cost of requiring significant amounts of additional data. In [20], an inpainting network is trained to identify the regions that are harder to reconstruct from the surrounding image patches and encodes and decodes the content of this region to learn the scene decomposition. [53] employs a similar inpainting network but on flow fields obtained by [44] and aims to generate the mask of a moving object in the region where the inpainting network yields poor reconstruction. Methods based on deep optical flow are not strictly self-supervised and can yield degenerate solutions when applied to still images with no or little movement. Recently, temporal information at different granularities has also been used via forward-backward patch tracking [31]. Note that these methods can only operate on video streams and exploit a strong temporal dependency, which our model doesn't.

Recent self-supervised methods that operate on single RGB images employ generative models to detect the regions that can be exposed to certain transformations without changing the realism of the image [5, 6, 1, 3]. However, these methods can easily be deceived by other background objects whose random displacement or texture change can still yield realistic images. In contrast to all of these techniques, our approach works with single images acquired using a moving camera and with an arbitrary background.

Multi-View Self-Supervised Approaches. Other relevant approaches include the generative unsupervised multi-person detection and tracking methods proposed in [13, 2]. The former localizes and matches persons across several cameras with overlapping fields of view using a grid of candidate positions on the ground plane. The latter uses a joint CNN-CRF architecture and Mean-Field inference to produce a Probabilistic Occupancy Map (POM) as in [13] but leverages discriminative features extracted by a CNN. Both require background subtraction images as input and can therefore only work with static cameras. Furthermore, they exploit multiple views at inference time, whereas we aim to perform monocular person segmentation.

Multi-View Self-Supervised Training for Single View Inference. Our work is closely related to [38, 37] in that we do not use any segmentation annotation to learn the foreground region. In [38, 37], novel view synthesis is used in conjunction with multi-view synchronized videos of human motions captured by calibrated cameras to learn a geometry-aware embedding. In contrast to our approach, it requires a known background to decompose the scene into foreground and background regions. Hence, it cannot handle scenes filmed by moving cameras. Here, we introduce a method that works with a changing background. To this end, we do not rely on novel view synthesis but instead exploit multi-view consistency by relating the 2D detections of the multiple views to a common 3D capture volume.

3. Method

Our goal is to develop a self-supervised algorithm that generates a bounding box and the corresponding segmentation mask from a single image. However, whereas earlier methods use videos from a single camera for training purposes, we want to demonstrate that using calibrated and synchronized cameras for training purposes increases performance. Therefore, let us assume that we have videos acquired by $C > 1$ calibrated and synchronized projective cameras. For each c between 1 and C, camera c captures image $\mathbf{I}_c$ and its behavior is modeled by a 3×4 projection matrix $\mathbf{P}_c$.

3.1. Multi-View Self-Supervised Training

Let us now turn to the task of exploiting such multi-view data to train our detection and segmentation network. Because we ultimately aim to perform single-view 2D detection and segmentation, our approach produces bounding boxes and segmentation masks for each individual view. Nevertheless, we exploit multi-view geometry to better constrain the training process and enforce consistency across the views. Furthermore, we do this without requiring additional loss terms that would make the process more complex and force us to carefully weigh these additional terms against the original ones. To this end, our training algorithm goes through the following steps

1. We use a network $\mathcal{F}$ to compute a probability map for 2D bounding boxes over an image grid for each view c. These probability maps are used to vote in a 3D grid for potential 3D locations of these bounding boxes.
2. We sample individual 3D voxels in that 3D grid according to the resulting probability density. This corresponds to one 2D bounding box for each view.
3. We compute the 3D center and object height that best agree with these 2D bounding boxes in a least-square sense.
4. We project the resulting 3D center and height in each view to define new 2D bounding boxes, keeping the original width of the sampled boxes.
5. These boxes are then used to evaluate the loss function associated to $\mathcal{F}$ in each image.

Multi-view consistency is achieved both by sampling the 3D proposal grid and adjusting the 2D bounding boxes. Hence, we do not require additional losses to enforce consistency. This is a central element of our approach because, as observed in [39], such loss terms tend to favor degenerate solutions that are consistent but wrong. This is something our ablation study confirms. In the remainder of this section, we describe these steps in more detail.

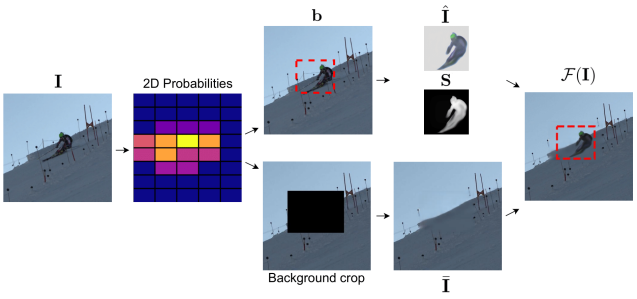


Figure 3: **Single-view self-supervised segmentation.** This figure summarizes our starting point, the single view approach. It predicts 2D occupancy probabilities, an associated bounding box, and a foreground mask within this window. It is trained to reconstruct the input image by pasting the foreground region underneath the mask on a background image obtained by inpainting the predicted bounding box.

3.1.1 Bounding Boxes in Individual Views

Let us consider the network $\mathcal{F}$ of [20], which we use as the backbone of our approach. It takes an image $\mathbf{I} \in \mathbb{R}^{W\times H\times 3}$ as input and resynthesizes it. In the process, it produces a probability map over a grid, encoding for each cell i the probability p_i that a bounding box $\mathbf{b}_i$ at this location contains a person. As depicted by Fig. 3, resynthesis is achieved by sampling a candidate bounding box, cropping the corresponding image patch, and, in parallel, predicting a foreground image $\hat{\mathbf{I}} \in \mathbb{R}^{128\times 128\times 3}$ and a segmentation mask $\mathbf{S} \in \mathbb{R}^{128\times 128}$ from the crop, while inpainting the cropped region to generate a background image $\bar{\mathbf{I}} \in \mathbb{R}^{W\times H\times 3}$. We then re-compose the foreground crop and the background image according to the segmentation mask. Formally, this can be written as

$$\mathcal{F}(\mathbf{I}) = \mathcal{T}^{-1}(\hat{\mathbf{I}} \circ \mathbf{S}) + \bar{\mathbf{I}} \circ (1 - \mathcal{T}^{-1}(\mathbf{S})), \qquad (1)$$

where $\mathcal{T}$ is the spatial transformer corresponding to the selected bounding box, and $\circ$ is the element-wise multiplication. This allows one to train $\mathcal{F}$ in a self-supervised fashion, by comparing the reconstructed image to the input one.

3.1.2 Consistent Sampling using a 3D Proposal Grid

To link 2D detections across views, we construct a 3D proposal grid with V voxels centered at the point nearest to the optical axes of all cameras in the 3D world coordinate system, as shown in Fig. 2. For each voxel j of that grid, we compute its center $\mathbf{v}_j \in \mathbb{R}^3$, together with a probability of occupancy q_j, discussed below.

Since we know the camera matrix $\mathbf{P}_c$ for each image $\mathbf{I}_c$, we can project the center $\mathbf{v}_j$ of each 3D voxel into it. The projected center will fall into image grid cell $i^c(j)$ to which $\mathcal{F}$ has associated a probability $p^c_{i^c(j)}$, as discussed at the beginning of Section 3.1.1. We repeat this operation over all images and all voxels and sum the resulting log probabilities for each voxel. We then normalize the resulting probability density over the 3D grid so that it integrates to one. Formally, this can be written as

$$q_j = \frac{1}{Z} \exp\left(\sum_c \log(p^c_{i^c(j)})\right) , \qquad (2)$$

where Z is a normalization constant easily computed on a discrete grid of finite dimensions.

To train our network in a self-supervised fashion, we then sample one voxel location j according to the distribution in Eq. 2. The sampled voxel then corresponds to one bounding box candidate in each view, inherently encouraging consistency across the views as illustrated in Fig. 4(a). This consistency, however, is only a partial one because each view still predicts the precise location and dimensions of its own bounding box. Hence, the final bounding boxes may still disagree. To prevent this, we explicitly enforce geometric consistency as discussed below.

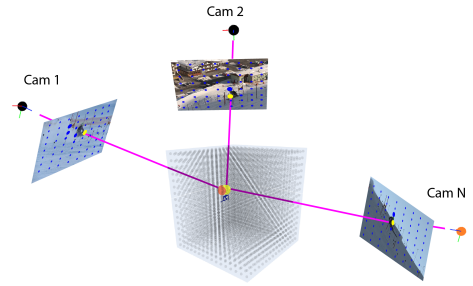
3.1.3 Enforcing Geometric Bounding Box Consistency

To enforce geometric consistency between the bounding boxes from different views, we want to ensure that their 2D centers all match the same point in 3D and that their 2D heights correspond to the same 3D size. In other words, we want to modify the bounding box locations so that the new ones have consistent 2D centers and heights and we want to achieve this with as little displacement as possible. Since the cameras are often set in a rough circle pointing at the subject, enforcing height consistency makes sense because the camera up directions are aligned. Only when the camera angle varies, as in drone footage taken from arbitrary angles, should the height constraint be replaced. We do not constrain the bounding-box width because the left-right direction of cameras is not aligned unless the cameras are parallel. This makes the width view dependent, as in Fig. 1 where the skier's projection is wider in some views.

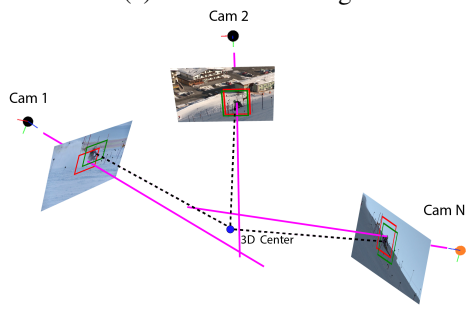
In essence, we seek to *project* the bounding boxes to new ones that satisfy the center and height constraints and that will be used by the network to evaluate its objective function during its forward pass. It is therefore essential that this projection be differentiable such that the backward pass can be carried out during training.

Adjusting Bounding Box Centers. As shown in Fig. 4(b), we use the lines of sights defined by the 2D centers of the bounding boxes, find the 3D point closest to all of them, and use its re-projection into the images as the modified center for the bounding boxes. For each view c, the line of sight $\mathbf{l}_c$ in image $\mathbf{I}_c$ can be expressed as

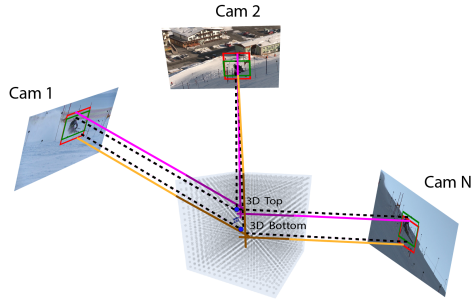
$$\mathbf{l}_c = \mathbf{M}_c^{-1} \left[u_c, v_c, 1\right]^T , \qquad (3)$$



(a) Multi-view voting.



(b) Making the bounding box centers consistent.



(c) Making the bounding box heights consistent.

Figure 4: **Finding bounding boxes that are view consistent.** (a) The blue dots overlaid on each view represent the initial 2D probabilities and vote in the 3D grid along their respective lines of sight. As a result, the yellow 3D voxel becomes very likely to be sampled. (b) The red bounding box drawn in each view is the initial prediction and the purple line of sight is going through the bounding box center. The 3D center is the point closest to all these lines and its re-projection in the images becomes the center of the new bounding boxes, shown in green. (c) The red bounding boxes represent the initial prediction and the purple and orange lines indicate the line of sight going through the bounding box top and bottom points. The 3D top and bottom locations are taken to be the point closest to purple and orange lines respectively. Their re-projection in the images become the top and bottom middle points of the new bounding boxes, shown in green.

where $\mathbf{M}_c$ is the 3×3 matrix formed by the first 3 columns of $\mathbf{P}_c$ and u_c, v_c are the 2D pixel coordinates of the bounding box center in $\mathbf{I}_c$. Hence, finding the point closest to all the $\mathbf{l}_c$ amounts to solving a least-squares problem, which can itself be achieved by solving a linear system of equations and is therefore differentiable. In practice, we use a differentiable least-squares implementation for this purpose and provide its details in the supplementary material.

Adjusting Bounding Box Heights. As shown in Fig. 4(c), we similarly use the midpoints of the top and bottom parts of the bounding boxes in each view to predict two new intersection points, one for the top and one for the bottom of the bounding box in 3D. We then take the distance between the re-projections of these points into the image to be the new height of the bounding boxes. As before, this is a differentiable operation.

3.1.4 Training

Because our 2D bounding boxes are made to be consistent, we can train our network by minimizing the same loss as in the single view approach of [20], except for the fact that we jointly compute it over several images, and do not require to introduce an additional loss to enforce consistency.

More specifically, we minimize the weighted sum of two loss functions $G(\mathbf{I}_1, \ldots, \mathbf{I}_C)$ and $O(\mathbf{I}_1, \ldots, \mathbf{I}_C)$. G accounts for the fact that a region containing a moving foreground object is unlikely to be well re-synthesized by the inpainter and is critical to train the network to place the bounding box at the right location in each image. O gauges how well $\mathcal{F}$ resynthesizes the complete original images and is minimized when the segmentation mask fits the salient object as well as possible within the sampled bounding box. In practice, they are taken to be

$$G(\mathbf{I}_1, \ldots, \mathbf{I}_C) = -\sum_{c=1}^{C} r_j \frac{\|\bar{\mathbf{I}}_c - \mathbf{I}_c\|^2}{area(\mathbf{b}^c_{i^c(j)})} , \quad (4)$$

$$O(\mathbf{I}_1, \ldots, \mathbf{I}_C) = \sum_{c=1}^{C} r_j \|\mathcal{F}(\mathbf{I}_c) - \mathbf{I}_c\|^2 , \quad (5)$$

where $area(\mathbf{b}^c_{i^c(j)}) \in \mathbb{N}_0$ is the area of the bounding box obtained by sampling voxel j and enforcing geometric consistency. As in [20], the sampled voxel is obtained by importance sampling, and r_j is the ratio of the probability q_j, from Eq. 2, by its importance sampling probability. In addition to these loss terms, as [20], we use an L_1 prior on $\mathbf{S}$ to favor a crisp segmentation, and compute Eq. 5 not only on pixel color but also on learned features. Additional details on the sampling, hyper-parameters, training and network architectures are provided in the supplementary material.

3.2. Single-View Inference

Once trained using multiple views, our model can detect and segment the salient object from single RGB images at inference time without any further changes. We run our network on the image and simply choose the 2D grid cell with

the highest occupancy probability. Its bounding box parameter estimations are fed into the spatial transformer $\mathcal{T}$ to crop the region of interest, which is encoded into the corresponding segmentation mask and foreground, and decoded into the reconstructed image as illustrated in Fig. 3.

4. Experiments

Unlike that of [37], our self-supervised approach is designed to work using multiple-cameras that can move. In this section, we show that it does, yet outperforms [37] even when the background is static. Furthermore, we show that using multiple cameras for training purposes delivers the hoped-for performance boost over state-of-the-art monocular approaches [22, 53, 9, 31, 20].

4.1. Images and Metrics

We first describe the image datasets we work with and then the metrics we use for comparison purposes.

Images acquired using moving cameras. The **Ski-PTZ** dataset of [39] features six skiers on a slalom course. We use the official training/validation/test sets that split the 12 videos of six skiers as four/one/one, with, respectively, 7800, 1818 and 1908 frames. The pan-tilt-zoom cameras constantly adjust to follow the skier. Nothing remains static, the background changes quickly, and there are additional people standing in the background. The cameras were calibrated using static scene markers *without* any markers or keypoints on the skier's body. We use the full image as input and evaluate detection accuracy using the available 2D pose annotations and segmentation accuracy of the 300 labeled frames in the test sequences. To pick the hyperparameters, we use 36 labeled validation frames (3 frames each from six cameras and two sequences). Due to the large distance between cameras and subject, the 3D proposal grid has 16^3 voxels with cuboid side length of 8 meters.

To demonstrate the applicability of our method to scenes without an initial camera calibration, we use the **Handheld190k** dataset [20] captured by three hand-held cameras that translate and rotate in an unscripted fashion. It comprises three training, one validation, and one test sequences. They all feature one person performing actions mimicking the human motions in an outdoor environment with a changing background. We used OpenSFM [1] to calibrate 4200 frames from the training set using and tested on the same images as [20]. The 3D proposal grid has 16^3 voxels with cuboid side length of 12 meters.

Images acquired using Static Cameras. To compare against algorithms requiring a static background, we evaluate our approach in the more controlled environment of the **H36M** dataset [18]. It was acquired using four static cameras and comprises 3.6 million frames and 15 motion classes. It features 5 subjects for training and 2 for validation, seen from different viewpoints against a static background and with good illumination. The 3D proposal grid consists of 10^3 voxels, with cuboid side length of 4 meters.

	Ski-PTZ		
Method	J Score	F Score	Run-time (sec)
Chen et al. [6]	0.37	0.42	0.11
Stretcu et al. [43]	0.51	0.56	0.02
Lu et al. [31]	0.51	0.60	0.60
Katircioglu et al. [20]	0.61	0.67	0.24
Rhodin et al. [37] + [20]	0.61	0.70	0.23
Croitoru et al. [9]	0.62	0.72	0.15
Yang et al. [53] w/o CRF	0.61	0.71	0.32
Yang et al. [53]	0.67	0.77	1.12
Katircioglu et al. [20] w/ flow	0.69	0.79	0.24
Koh et al. [22]	0.70	0.80	107.4
Ours	**0.71**	**0.83**	0.17

Table 1: **Segmentation results on the Ski-PTZ.** We compare against the state-of-the-art single-view approaches and a modified version of the multi-view approach of [37].

Metrics. We report our segmentation scores in J- and F-measure as defined in [34]. The former is defined as the intersection-over-union (IoU) between the ground-truth segmentation mask and the prediction, while the latter is the harmonic average between the precision and the recall at the mask boundaries. The detection scores are calculated in terms of $\text{mAP}_{0.5}$, the mean probability of having an IoU of more than 50%. Different segmentation algorithms set the foreground-background threshold differently. Hence, to allow a fair comparison, we perform a line search from 0 to 1 with a step-size of 0.05, selecting the optimal value for all baselines and variants for each individual dataset.

4.2. Comparative Results with Moving Cameras

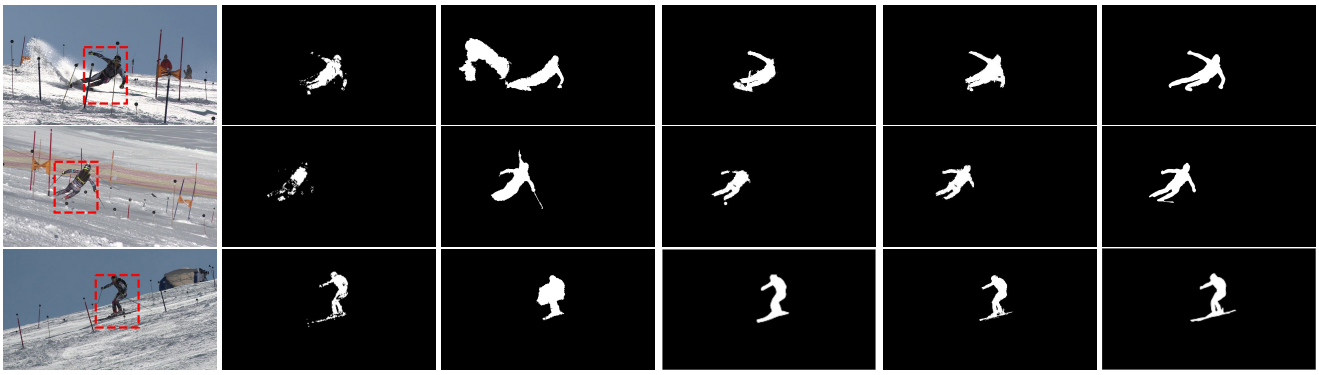
Fig. 5 depicts qualitative results on the **Ski-PTZ** dataset and we report the corresponding quantitative results using 4 cameras in Table 1, in which we use the scores reported in [20] for the baselines. [2]

We outperform all existing single-view self-supervised segmentation approaches [22, 53, 9, 20, 43, 6, 31] while being comparatively fast. For completeness, we also report results for [53] without CRF post-processing. This shows that a great deal of the method's performance comes from such post-processing, which we do not require. Note that, in contrast to [20] with flow and [22], our approach does not require computing optical flow. Unlike DAVIS [34], our datasets feature large camera motions with quick background changes, which causes methods such as [31] to often merge portions of the background and the human.

The only other self-supervised multi-view approach for which a public implementation is available is that of [37]. Unfortunately, it requires background images as an input,

[1] https://www.opensfm.org/

[2] The implementations of [20] and [31] were provided by the authors.



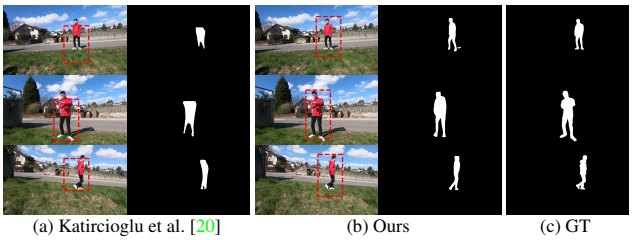
(a)Input/Ours detection (b) Yang et al. [53] (c) Koh et al. [22] (d) Katircioglu et al. [20] (e) Ours (f) GT

Figure 5: **Qualitative results on the Ski-PTZ dataset.** (a) Input images with our predicted bounding box overlaid in red. (b,c,d) Segmentation masks predicted by three of our baselines. (e) Our segmentation mask prediction. (f) Ground truth segmentation mask. Note the quality of our predicted masks even though, unlike the methods of [22] and [53], we do not use explicit temporal cues at inference time.

which are not given in this case and are not trivial to create because the cameras rotate and zoom. To do so anyway, we use the single-view approach of [20] to produce background images that we can feed to the network of [37] for multi-view training. As can be seen in Table 1, this modified version of [37] does slightly better than [20] in F score terms but remains far behind our method. The method that comes closest to ours is that of [22], which operates on the whole sequence and is therefore prohibitively slow as discussed below. By contrast our approach operates on a single image and does not require motion information.

The inference times for each method are shown in the last column of Table 1 and computed using code that is either publicly available or that the authors made available to us privately. All except those of [43, 22] were obtained using a single NVIDIA TITAN X Pascal GPU. Since [43, 22] are designed to run on CPU, the inference for them is computed on Intel(R) Xeon(R) Gold 6240 CPUs. The tailored optimization approach of [22] that comes closest to our results is three orders of magnitude slower than our approach because it tracks several patches over time. Unlike [53], our method does not require optical flow computation or CRF post-processing which brings a five-fold speedup. Our computational complexity is similar to that of [20, 37] since the triangulation time is negligible. The training time of our model on the **Ski-PTZ** is approximately 8 hours whereas that of [37] and [20] are 14 and 7.5 hours, respectively.

We also evaluate our method on **Handheld190k** trained using 4200 images from multiple views and compare against the network of [20] trained using the same 4200 images. We obtain a J-score of 0.66 instead of 0.64 and an F-score of 0.77 instead of 0.71, again showing the importance of multi-view consistency. Our method benefits from multi-view information obtained in an automated off-the-shelf manner, particularly in tightly fitting to the subject, as shown in Fig. 6. In short, the improvement demonstrated here highlights the previously untapped potential of multi-view constraints for self-supervised segmentation.



(a) Katircioglu et al. [20] (b) Ours (c) GT

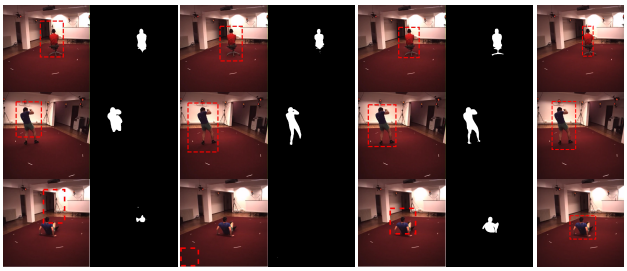
Figure 6: **Qualitative results on the Handheld190k dataset.** (a) The detection and segmentation mask results of [20] trained and tested on single images. (b) The predictions of our model trained using 3-camera multi-view consistency and tested on single images. (c) Ground truth. Our results are generally more accurate, which justifies the effort invested in calibrating the cameras.

			H36M
Method	Training Type	Background Assumption	mAP
Katircioglu et al. [20]	single-view	dynamic	0.57
Rhodin et al. [37]	multi-view	static	0.71
Ours	multi-view	dynamic	**0.85**

Table 2: **Comparative results on the H36M dataset.** Our detection accuracy improves in terms of $mAP_{0.5}$.

4.3. Comparative Results with Static Cameras

In the previous example, we had to modify the multi-view self-supervised algorithm of [37] to make it work on images with a moving background. To evaluate the original version instead, we compare on the **H36M** dataset and report the results using again 4 cameras in Table 2. As in the **Ski-PTZ** case, we outperform it and, this time, the difference cannot be caused by any background modification we made. This is somewhat surprising because the method



(a) Katircioglu et al. [20] (b) Rhodin et al. [37] (c) Ours (d) GT

Figure 7: **Qualitative results on the H36M dataset.** (a) The detection and segmentation results of [20] trained and tested on single images. (b) The results of [37] trained with a pair of camera views and tested on single images. (c) Our predictions obtained from the model trained with the 4-cam multi-view consistency and tested on single images. (d) Ground truth. Our method consistently detects the person whereas [20, 37] occasionally produce inconsistent results, such as the failed detections in the last row.

	# Cam	*Ours w/o VC*	*Ours w/o HC*	*Ours w/ TC*	*Ours w/ WC*	*Ours*
J Score	2	0.66	0.67	0.66	0.61	0.66
	3	0.68	0.70	0.68	0.68	**0.71**
	4	0.68	0.70	0.67	0.68	**0.71**
	5	0.67	0.67	0.67	0.68	0.69
	6	0.66	0.70	0.67	0.67	0.68
F Score	2	0.73	0.73	0.73	0.65	0.75
	3	0.75	0.77	0.75	0.74	0.81
	4	0.75	0.79	0.75	0.77	**0.83**
	5	0.74	0.74	0.74	0.75	0.78
	6	0.73	0.78	0.73	0.74	0.76

Table 3: **Ablation study on the Ski-PTZ**. We test variants of our approach while using varying numbers of cameras.

	# Cam	*Ours w/o VC*	*Ours w/o HC*	*Ours w/ TC*	*Ours w/ WC*	*Ours*
mAP	2	0.73	0.74	0.74	0.73	0.75
	3	0.78	0.80	0.79	0.79	0.82
	4	0.79	0.83	0.82	0.84	**0.85**

Table 4: **Ablation study on the H36M dataset.** We test variants of our approach while using varying numbers of cameras.

of [37] assumes a constant static background, which is the case here, whereas ours is learned without any such constraint. We attribute this result to the explicit consistency of bounding box positions in 3D and the background inpainting constraint. The latter triggers when part of the subject is outside the bounding box leading to correctly segmented legs while the method of [37] has trouble distinguishing the skin and floor color when in shadow, as depicted in Fig. 7. See additional qualitative results in the supplementary.

In Table 2, we also report the result of [20], that is, our backbone network run on single views. The performance drops, which once again highlights the usefulness to exploit multiple views for training when they are available.

4.4. Ablation Study

We compare the following variants of the multi-view constraints of Section 3.1: *Ours* denotes the full model that employs all the steps shown in Fig. 4. *Ours w/o HC* excludes the bounding box height consistency depicted by Fig. 4 (c). *Ours w/o VC* leaves out both the center and height adjustment of Fig. 4 (b,c) and enforces only consistent sampling. *Ours w/ WC* imposes a bounding box width consistency in addition to the full model. Finally, *Ours w/ TC* is a baseline that replaces the view consistency with a triangulation loss minimizing the distance between the lines joining the centers of the camera and predicted 2D bounding box.

3D Grid Size	**Ski-PTZ** J-Score	3D Grid Size	**H36M** $mAP_{0.5}$
$[10 \times 10 \times 10]$	0.64	$[6 \times 6 \times 6]$	0.76
$[16 \times 16 \times 16]$	**0.68**	$[10 \times 10 \times 10]$	**0.79**
$[24 \times 24 \times 24]$	0.66	$[16 \times 16 \times 16]$	0.76

Table 5: **Influence of voxel resolution.** The numbers in square brackets indicate the number of voxels in the 3D proposal grid and we use 4 cameras.

In Table 3 and Table 4 we report results as a function of the number of cameras we used. We can use only 2 cameras but the best results are obtained for 3 or 4. Beyond that, additional cameras add little new information while taking more space in the training batches, resulting in less diverse batches and lower performance. The numbers for the different variants in Fig. 4 show that all the elements we have incorporated into our approach contribute positively and that the one we have purposely ignored—constraining the width—would degrade performance. Crucially *Ours w/ TC* also performs worse, hence substantiating our claim that imposing consistency constraints using the projection mechanism of Section 3.1.3 is crucial to our success.

We also analyzed the influence of the voxel resolution on the reconstruction accuracy. Table 5 shows that a 10^3 cube is more accurate than a 6^3 cube while going to a 16^3 does not bring further improvements in **H36M** dataset. The 0.01 lower mAP may indicate that learning a discrete distribution on the 3D grid may be less efficient on larger spaces. However, as the ski footage covers a wider area, a 16^3 cube yields the best performance on the **Ski-PTZ**.

5. Conclusion

We have presented a self-supervised detection and segmentation technique that exploits multi-view geometry during training to accurately separate foreground from background in single RGB images at inference time. It outperforms the state-of-the-art on the challenging **Ski-PTZ**, depicting unusual activities captured with moving cameras, and on **H36M**, acquired with static cameras. We have focused on scenes with a single salient object. However, our method has the potential to handle multiple objects by sampling more than one proposal as long as they are not overlapping. Our future work will be in this direction.

References

[1] R. Arandjelovic and A. Zisserman. Object Discovery with a Copy-Pasting GAN. In *arXiv Preprint*, 2019. 3

[2] P. Baqué, F. Fleuret, and P. Fua. Deep Occlusion Reasoning for Multi-Camera Multi-Target Detection. In *International Conference on Computer Vision*, 2017. 3

[3] Y. Benny and L. Wolf. OneGAN: Simultaneous Unsupervised Learning of Conditional Image Generation, Foreground Segmentation, and Fine-Grained Clustering. In *European Conference on Computer Vision*, 2020. 1, 3

[4] G. Bhat, F. J. Lawin, M. Danelljan, A. Robinson, M. Felsberg, L. V. Gool, and R. Timofte. Learning What to Learn for Video Object Segmentation. In *European Conference on Computer Vision*, 2020. 2

[5] A. Bielski and P. Favaro. Emergence of Object Segmentation in Perturbed Generative Models. In *Advances in Neural Information Processing Systems*, 2019. 1, 3

[6] M. Chen, T. Artieres, and L. Denoyer. Unsupervised Object Segmentation by Redrawing. In *Advances in Neural Information Processing Systems*, 2019. 1, 3, 6

[7] J. Cheng, Y. H. Tsai, S. Wang, and M. H. Yang. Segflow: Joint Learning for Video Object Segmentation and Optical Flow. In *International Conference on Computer Vision*, 2017. 2

[8] E. Crawford and J. Pineau. Spatially Invariant Unsupervised Object Detection with Convolutional Neural Networks. In *Conference on Artificial Intelligence*, 2019. 1

[9] I. Croitoru, S. V. Bogolin, and M. Leordeanu. Unsupervised Learning of Foreground Object Segmentation. *International Journal of Computer Vision*, 127:1279–1302, 2019. 1, 2, 3, 6

[10] S.M.A. Eslami, N. Heess, T. Weber, Y. Tassa, D. Szepesvari, K. Kavukcuoglu, and G. Hinton. Attend, Infer, Repeat: Fast Scene Understanding with Generative Models. In *Advances in Neural Information Processing Systems*, 2016. 1

[11] A. Faktor and M. Irani. Video Segmentation by Non-Local Consensus Voting. In *British Machine Vision Conference*, 2014. 2

[12] O.D. Faugeras and Q.T. Luong. *The Geometry of Multiple Images*. MIT Press, 2001. 2

[13] 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. 3

[14] E. Haller and M. Leordeanu. Unsupervised Object Segmentation in Video by Efficient Selection of Highly Probable Positive Features. In *International Conference on Computer Vision*, 2017. 2

[15] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000. 2

[16] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick. Mask R-CNN. In *International Conference on Computer Vision*, 2017. 2

[17] Y. T. Hu, J. B. Huang, and A. G. Schwing. Unsupervised Video Object Segmentation Using Motion Saliency-Guided Spatio-Temporal Propagation. In *European Conference on Computer Vision*, 2018. 2

[18] C. Ionescu, I. Papava, V. Olaru, and C. Sminchisescu. Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing in Natural Environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2014. 2, 6

[19] S. D. Jain, B. Xiong, and K. Grauman. Fusionseg: Learning to Combine Motion and Appearance for Fully Automatic Segmentation of Generic Objects in Videos. In *Conference on Computer Vision and Pattern Recognition*, 2017. 2

[20] I. Katircioglu, H. Rhodin, V. Constantin, J. Spörri, M. Salzmann, and P. Fua. Self-Supervised Segmentation via Background Inpainting. In *arXiv Preprint*, 2020. 1, 2, 3, 4, 5, 6, 7, 8

[21] M. Keuper, B. Andres, and T. Brox. Motion Trajectory Segmentation via Minimum Cost Multicuts. In *International Conference on Computer Vision*, 2015. 2

[22] Y. J. Koh and C.-S. Kim. Primary Object Segmentation in Videos Based on Region Augmentation and Reduction. In *Conference on Computer Vision and Pattern Recognition*, 2017. 1, 2, 6, 7

[23] Y.J. Lee, J. Kim, and A. K. Grauman. Key-Segments for Video Object Segmentation. In *International Conference on Computer Vision*, 2011. 2

[24] M. Leordeanu. *Unsupervised Learning in Space and Time*. Springer, 2020. 1

[25] S. Li, B. Seybold, A. Vorobyov, A. Fathi, Q. Huang, and C.-C. J. Kuo. Instance Embedding Transfer to Unsupervised Video Object Segmentation. In *Conference on Computer Vision and Pattern Recognition*, 2018. 2

[26] S. Li, B. Seybold, A. Vorobyov, X. Lei, and C.-C. J. Kuo. Unsupervised Video Object Segmentation with Motion-Based Bilateral Networks. In *European Conference on Computer Vision*, 2018. 2

[27] Y. Li, Z. Shen, and Y. Shan. Fast Video Object Segmentation using the Global Context Module. In *European Conference on Computer Vision*, 2020. 2

[28] Z. Lin, Y-F. Wu, S. Vi. Peri, W. Sun, G. Singh, F. Deng, J. Jiang, and S. Ahn. SPACE: Unsupervised Object-Oriented Scene Representation via Spatial Attention and Decomposition. In *International Conference on Learning Representations*, 2020. 1

[29] X. Lu, W. Wang, M. Danelljan, T. Zhou, J. Shen, and L. V. Gool. Video Object Segmentation with Episodic Graph Memory Networks. In *European Conference on Computer Vision*, 2020. 2

[30] X. Lu, W. Wang, C. Ma, J. Shen, L. Shao, and F. Porikli. See More, Know More: Unsupervised Video Object Segmentation with Co-Attention Siamese Networks. In *Conference on Computer Vision and Pattern Recognition*, 2019. 2

[31] X. Lu, W. Wang, J. Shen, Y. Tai, D. Crandall, and S.C.H. Hoi. Learning Video Object Segmentation from Unlabeled Videos. In *Conference on Computer Vision and Pattern Recognition*, 2020. 1, 2, 3, 6

[32] Mapillary. OpenSFM. Available at https://www.opensfm.org/.

[33] A. Papazoglou and V. Ferrari. Fast Object Segmentation in Unconstrained Video. In *International Conference on Computer Vision*, pages 1777–1784, 2013. 2

[34] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung. A Benchmark Dataset and Evaluation Methodology for Video Object Segmentation. In *Conference on Computer Vision and Pattern Recognition*, 2016. 6

[35] F. Perazzi, O. Wang, M. Gross, and A. S.-Hornung. Fully Connected Object Proposals for Video Segmentation. In *International Conference on Computer Vision*, 2015. 2

[36] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You Only Look Once: Unified, Real-Time Object Detection. In *Conference on Computer Vision and Pattern Recognition*, 2016. 2

[37] H. Rhodin, V. Constantin, I. Katircioglu, M. Salzmann, and P. Fua. Neural Scene Decomposition for Human Motion Capture. In *Conference on Computer Vision and Pattern Recognition*, 2019. 1, 2, 3, 6, 7, 8

[38] H. Rhodin, M. Salzmann, and P. Fua. Unsupervised Geometry-Aware Representation for 3D Human Pose Estimation. In *European Conference on Computer Vision*, 2018. 3

[39] H. Rhodin, J. Spoerri, I. Katircioglu, V. Constantin, F. Meyer, E. Moeller, M. Salzmann, and P. Fua. Learning Monocular 3D Human Pose Estimation from Multi-View Images. In *Conference on Computer Vision and Pattern Recognition*, 2018. 3, 6

[40] S. Seo, J.-Y. Lee, and B. Han. URVOS: Unified Referring Video Object Segmentation Network with a Large-Scale Benchmark. In *European Conference on Computer Vision*, 2020. 2

[41] H. Seong, J. Hyun, and E. Kim. Kernelized Memory Network for Video Object Segmentation. In *European Conference on Computer Vision*, 2020. 2

[42] H. Song, W. Wang, S. Zhao, J. Shen, and K.-M. Lam. Pyramid Dilated Deeper ConvLSTM for Video Salient Object Detection. In *European Conference on Computer Vision*, 2018. 2

[43] O. Stretcu and M. Leordeanu. Multiple Frames Matching for Object Discovery in Video. In *British Machine Vision Conference*, 2015. 2, 3, 6, 7

[44] D. Sun, X. Yang, M. Liu, and J. Kautz. Pwc-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume. In *Conference on Computer Vision and Pattern Recognition*, 2018. 3

[45] P. Tokmakov, K. Alahari, and C. Schmid. Learning Motion Patterns in Videos. In *Conference on Computer Vision and Pattern Recognition*, 2017. 2

[46] P. Tokmakov, K. Alahari, and C. Schmid. Learning Video Object Segmentation with Visual Memory. In *International Conference on Computer Vision*, 2017. 2

[47] C. Wang, D. Xu, Y. Zhu, R. Martín-Martín, C. Lu, L. Fei-Fei, and S. Savarese. Densefusion: 6D Object Pose Estimation by Iterative Dense Fusion. In *Conference on Computer Vision and Pattern Recognition*, 2019. 2

[48] W. Wang, J. Shen, and F. Porikli. Saliency-Aware Geodesic Video Object Segmentation. In *Conference on Computer Vision and Pattern Recognition*, 2015. 2

[49] W. Wang, H. Song, S. Zhao, J. Shen, S. Zhao, S.C. Hoi, and H. Ling. Learning Unsupervised Video Object Segmentation Through Visual Attention. In *Conference on Computer Vision and Pattern Recognition*, 2019. 2

[50] X. Wang, A. Jabri, and A.A. Efros. Learning Correspondence from the Cycle-Consistency of Time. In *Conference on Computer Vision and Pattern Recognition*, 2019. 2

[51] Y. Wang, Y. Liu, X. Tong, Q. Dai, and P. Tan. Outdoor Markerless Motion Capture with Sparse Handheld Video Cameras. *IEEE Transactions on Visualization and Computer Graphics*, 24:1856–1866, 2018. 2

[52] X.-S. Wei, C.-L. Zhang, J. Wu, C. Shen, and Z.-H. Zhou. Unsupervised Object Discovery and Co-Localization by Deep Descriptor Transforming. In *arXiv Preprint*, 2017. 2

[53] Y. Yang, A. Loquercio, D. Scaramuzza, and S. Soatto. Unsupervised Moving Object Detection via Contextual Information Separation. In *Conference on Computer Vision and Pattern Recognition*, 2019. 1, 2, 3, 6, 7

[54] Z. Yang, Q. Wang, L. Bertinetto, W. Hu, S. Bai, and P. H. S. Torr. Anchor Diffusion for Unsupervised Video Object Segmentation. In *International Conference on Computer Vision*, 2019. 2

[55] Z. Yang, Y. Wei, and Y. Yang. Collaborative Video Object Segmentation by Foreground-Background Integration. In *European Conference on Computer Vision*, 2020. 2

[56] L. Zhang, J. Zhang, Z. Lin, R. Měch, H. Lu, and Y. He. Unsupervised Video Object Segmentation with Joint Hotspot Tracking. In *European Conference on Computer Vision*, 2020. 2

[57] M. Zhen, S. Li, L. Zhou, J. Shang, H. Feng, T. Fang, and L. Quan. Learning Discriminative Feature with CRF for Unsupervised Video Object Segmentation. In *European Conference on Computer Vision*, 2020. 2

[58] D. Zou and P. Tan. CoSLAM: Collaborative Visual SLAM in Dynamic Environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(2):354–366, 2013. 2