TU Delft

Delft University of Technology

Privacy protection in street-view panoramas using depth and multi-view imagery

Uittenbogaard, Ries; Sebastian, Clint; Vijverberg, Julien; Boom, Bas; Gavrila, Dariu; De With, Peter H.N.

DOI
10.1109/CVPR.2019.01083

Publication date
2019

Document Version
Final published version

Published in
Proceedings IEEE Computer Vision and Pattern Recognition (CVPR 2019)

Citation (APA)
Uittenbogaard, R., Sebastian, C., Vijverberg, J., Boom, B., Gavrila, D., & De With, P. H. N. (2019). Privacy protection in street-view panoramas using depth and multi-view imagery. In *Proceedings IEEE Computer Vision and Pattern Recognition (CVPR 2019)* (pp. 10573-10582). IEEE. https://doi.org/10.1109/CVPR.2019.01083

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright
Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy
Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

This work is downloaded from Delft University of Technology.
For technical reasons the number of authors shown on this cover page is limited to a maximum of 10.

Privacy Protection in Street-View Panoramas using Depth and Multi-View Imagery

Ries Uittenbogaard[1], *Clint Sebastian[2], Julien Vijverberg[3],
Bas Boom[3], Dariu M. Gavrila[1], Peter H.N. de With[2]

[1]Intelligent Vehicles Group, TU Delft, [2]VCA Group, TU Eindhoven, [3]Cyclomedia B.V

c.sebastian@tue.nl

*corresponding author

Abstract

The current paradigm in privacy protection in street-view images is to detect and blur sensitive information. In this paper, we propose a framework that is an alternative to blurring, which automatically removes and inpaints moving objects (e.g. pedestrians, vehicles) in street-view imagery. We propose a novel moving object segmentation algorithm exploiting consistencies in depth across multiple street-view images that are later combined with the results of a segmentation network. The detected moving objects are removed and inpainted with information from other views, to obtain a realistic output image such that the moving object is not visible anymore. We evaluate our results on a dataset of 1000 images to obtain a peak noise-to-signal ratio (PSNR) and L_1 loss of 27.2 dB *and 2.5%, respectively. To assess overall quality, we also report the results of a survey conducted on 35 professionals, asked to visually inspect the images whether object removal and inpainting had taken place. The inpainting dataset will be made publicly available for scientific benchmarking purposes at* https://research.cyclomedia.com/.

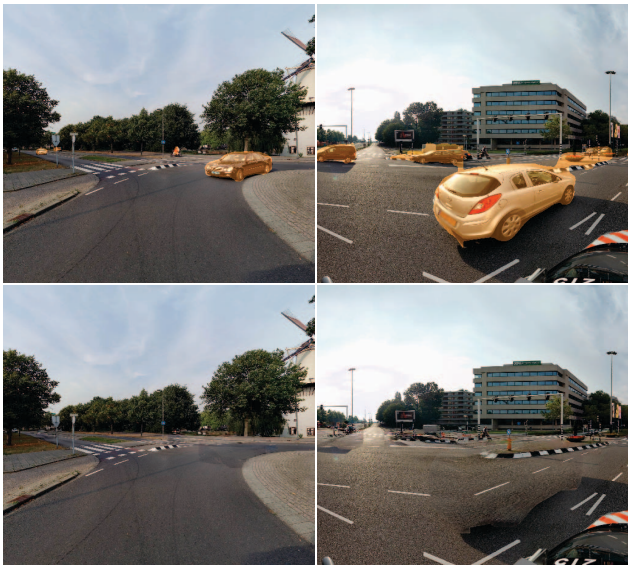

Figure 1: Example of moving object segmentation (top) and the results after inpainting (bottom). The regions that are highlighted in orange are removed and inpainted.

1. Introduction

In recent years, street-view services such as Google Street View, Bing Maps Streetside, Mapillary have systematically collected and hosted millions of images. Although these services are useful, they have been withdrawn or not updated in certain countries [1, 2], due to serious privacy concerns. The conventional way of enforcing privacy in street-view images is by blurring sensitive information such as faces and license plates. However, this has several drawbacks. First, the blurring of an object like a face might not ensure that the privacy of the person is sufficiently protected. The clothing, body structure, location and several other aspects can lead to the identity of the person, even if the face is hidden. Second, blurring objects creates artifacts that is undesirable in application when consistent view of the infrastructure is required. For commercial purposes such as change detection and localization of objects from street-view imagery, blurring limits the scope of these applications.

It is therefore desirable to use a method that completely removes the identity-related information. In this paper, we present a method that automatically segments and replaces moving objects from sequences of panoramic street-view images by inserting a realistic background. Moving objects are of primary interest because authentic information about the background is present in the other views (we assume that most moving objects are either vehicles or pedestrians).

978-1-7281-3293-8/19/$31.00 ©2019 IEEE
DOI 10.1109/CVPR.2019.01083

IEEE computer society

Method	Input data	Image matching	Detection	Class/No. of objects	Inpainting method
Flores *et al.*	Grayscale, Multi-view	Homography computation using SIFT & RANSAC	Leibe's detector (no moving object detection, manual)	One pedestrian per image	Homography-based warping + Davis compositing
Ours	RGB-D, Multi-view	Real world positions from GPS, IMU with camera intrinsics	Novel deep learning based moving object segmentation	Any number of objects and classes per image	Reprojection from multiple views + multiview inpainting GAN

Table 1: Comparison of the proposed vs. closest related method.

It is risky to inpaint static objects after removing them, as they may remove important or introduce new information from context. We do not focus on inpainting static objects such as parked cars, standing pedestrians etc. as no authentic information of the background can be derived. However, inpainting from spatial context is viable solution for non-commercial applications. Using a segmentation network to detect a moving object is a challenging task, since it needs to learn to distinguish moving from static objects. To simplify this problem, we generate a prior segmentation mask, exploiting the consistencies in depth. The proposed moving object detection algorithm, combines with results from a standard segmentation algorithm to obtain segmentation masks for moving objects. Finally, to achieve an authentic completion of the removed moving objects, we use inpainting Generative Adversarial Network (GAN) that utilizes multi-view information. While multi-view GAN has been explored for synthesizing an image to another view [3, 4], to the best of our knowledge, this is the first work that exploits multi-view information for an inpainting GAN network.

2. Related Work

Privacy protection Several approaches have been proposed for privacy protection in street-view imagery [5, 6, 7, 8, 9]. The most common way to hide sensitive information is to detect the objects of interest and blur them [10]. However, few works have explored the removal of privacy-sensitive information from street-view imagery for privacy protection. Flores and Belongie [6] detects and inpaints pedestrians from another view (details in Table 1). Similarly, Nodari *et al*. [8] also focuses on pedestrians removal. However, they remove the pedestrian with a coarse inpainting of the background. This is followed by replacement of the inpainted region with a pedestrian obtained from a controlled and authorized dataset. Although this method ensures the privacy, the replaced pedestrians tend to appear unrealistic with respect to the background.

Object detection Due to the progress in deep learning, there have been significant improvements in object detection [11, 12, 13]. Detecting objects of interest provides a reliable way to localize faces and license plates. Similarly, for precise localization, semantic segmentation offers a better alternative to bounding boxes [14, 15, 16]. Hence, we rely on semantic segmentation approaching pixel accuracy, as it requires fewer pixels to be replaced during inpainting. We obtain our segmentation masks through a combination of the proposed moving object segmentation and segmentation from a fully convolutional deep neural network.

In the recent years, LiDAR systems has become ubiquitous for applications like self-driving cars and 3D scene reconstruction. Several moving object detection methods rely on LiDAR as it provides rich information about the surroundings [17, 18, 19, 20]. Few approaches convert LiDAR-based point-clouds into 3D occupancy grids or voxelize them [17, 18]. These are later segregated into occupied and non-occupied building blocks. The occupied building blocks are grouped into objects and are tracked over time to determine moving objects [21]. Fusion of both LiDAR and camera data has also been applied for object detection [20, 19, 21]. In this case, consistency across both image and depth data (or other modalities) in several frames are checked to distinguish static and moving objects.

Inpainting Prior works have tried to produce a realistic inpainting by propagating known structures at the boundary into the region that is to be inpainted [22]. However, in street-view imagery, this is a challenging task especially when it has large holes and require complex inpainting. Therefore, few results relied on an exemplar or multi-view based methods [23, 24]. State-of-the-art of inpainting methods adopt Generative Adversarial Networks (GANs) to produce high-quality inpainted images [25]. GANs are often applied for problems such as image inpainting [26, 27, 28], image-to-image translation [29, 30], conditional image translation [31, 32] and super-resolution [33, 34].

Different approaches have been proposed for inpainting images using deep neural networks. Pathak *et al*. proposed one of the first methods that utilized a deep neural network [26]. They applied a combination of both reconstruction and adversarial losses to improve the quality of the inpainted images. This was improved in [27], using dilated convolutions and an additional local discriminator. To improve the quality of details in the output image, Zhao *et al*. proposes to use a cascade of deep neural networks [35]. The network first inpaints with a coarse result, followed by a deblurring-denoising network to refine the output. A multi-stage approach for inpainting is also proposed in [28]. Yu *et*

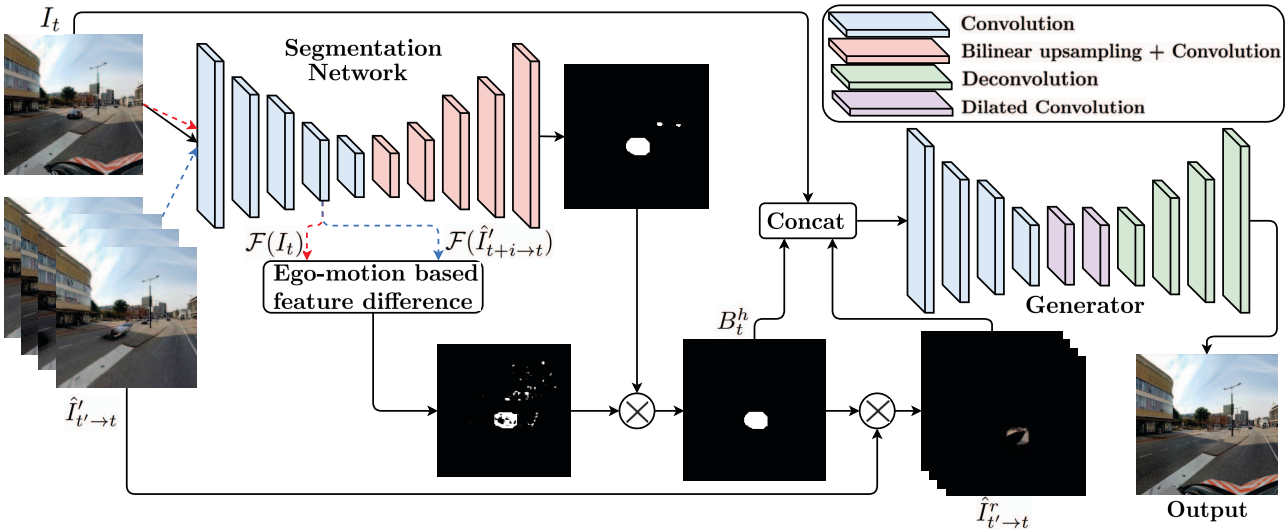

Figure 2: Overview of the proposed method to segment moving objects and inpaint them from other views. The input image is first fed to segmentation network to produce the segmentation mask of both moving and static objects. The difference of the convolution features of the reprojected images and input image is used to find the moving objects in the segmentation mask. The original input image (I_t), moving object segmentation mask (B_t^h) and the reprojected images with regions active in the segmentation mask ($\hat{I}^r_{t'\to t}$) is fed to the generator. Note that the discriminators networks are not shown for simplicity.

al. introduces an attention based inpainting layer that learns to inpaint by copying features from context. They also introduce a spatially discounted loss function in conjunction with improved Wasserstein GAN objective function [36] to improve the inpainting quality.

Although inpainting based on context produces plausible outputs for accurate image completion, GANs may introduce information that is not present in reality. This is undesirable, especially in commercial applications, where objects of interest are present or accurate localization are required. A reasonable idea here is to utilize information from other views as a prior. Other views provide a better alternative than inpainted information from scratch. An example would be the case when an object of interest (e.g. traffic-sign, bill-board) is occluded by a car or person. After moving object detection, using a GAN to inpaint the hole from context information would remove the object of interest. However, multi-view information could alleviate this problem as the object of interest is visible in the other views. Our main paper contributions are:

- We propose a new multi-view framework for detecting and inpainting moving objects, as an alternative to blurring in street-view imagery
- We introduce a new moving object detection algorithm based on convolutional features that exploits depth consistencies from a set of consecutive images.
- We train an inpainting GAN that utilizes multi-view information to obtain authentic and plausible results.

3. Method

First, we construct a method that combines standard segmentation with a novel moving object segmentation, which segments the moving objects from a consecutive set of images that have a large baseline. The moving object is estimated using an ego-motion based difference of convolutional features. Second, we use a multi-view inpainting GAN to fill-in the regions that are removed from moving object detection algorithm. The overview of the proposed framework is shown in Fig 2.

3.1. Moving Object Segmentation

For supervised segmentation, we apply a Fully Convolutional VGGNet (FC-VGGNet), due to its simplicity and the rich features that are used for moving object segmentation. We make slight modifications to VGGNet-16 by removing fully-connected layers and appending bilinear upsampling followed by convolution layer in the decoder. To create the segmentation mask for a specific image I_t at time t, the detection algorithm also uses the two images captured before and after it, *i.e*. the RGB images at time $t-2,\ldots,t+2$. Finally, from the LiDAR-based point cloud and the positions of each recording, the depth images for these time steps are created. Note that the RGB and the depth image are not captured at the same time. Hence, the moving objects are not at the same positions. Reprojecting the image $I_{t'}$ to the position of image I_t is achieved using its respective depth images $D_{t'}$ and D_t. Employing the depth images in conjunction with recorded GPS positions leads to real-world pixel positions $\vec{p}_{t'}$, $\vec{p}_t$, resulting in the defined image re-

Figure 3: Results from features extracted from FC-VGGNet (first) and VGGNet (second). Features from FC-VGGNet are well-localized and have strong activations.

projection $\hat{I}_{t'\to t}$. Evidently, some pixels in $\hat{I}_{t'\to t}$ cannot be filled due to occlusions. These pixels are replaced by the pixel values of I_t by comparing the distance between the real-world points to an heuristically defined threshold ϵ. This reprojection with a threshold is given by

$$\hat{I}'_{t'\to t} = \begin{cases} \hat{I}_{t'\to t} & \text{if } ||\vec{p}_t - \vec{p}'_{t'}|| < \epsilon, \\ I_t & \text{otherwise.} \end{cases} \tag{1}$$

Fig. 4 (Row 1) shows an example of reprojection for 4 neighbouring recordings of 5 consecutive images. A simple pixel-wise comparison between $\hat{I}'_{t'\to t}$ and I_t yields poor segmentation results, due to slight variations in the position of the car. We have empirically found that patch-based comparison utilizing pretrained network features produce better results than conventional pixel-wise and patch-based features. Feature extraction is often applied to generate descriptors or rich representations for applications such as image retrieval and person reidentification [37, 38, 39]. However, here we utilize the extracted features to obtain the moving objects. Instead of using VGG [40] or other pretrained network features, we extract features from FC-VGGNet that is trained to detect static objects, as it is easier to reuse the same network for moving object segmentation. Besides the simplicity of relying on a single network and higher performance, this also speeds up the pipeline. High-dimensional features $\mathcal{F}(I) \in \mathbb{R}^{64\times256\times512}$ are extracted from the output of the 4^{th} convolution block. The moving object segmentation score is the average of the L_1 norms between each of the projected images and I_t, which is specified by

$$s_t^{1/8} = \frac{1}{4} \sum_{i\in\{-2,-1,1,2\}} ||\mathcal{F}(I_t) - \mathcal{F}(\hat{I}'_{t+i\to t})||_1 \,, \tag{2}$$

where $s_t^{1/8}$ is upsampled by factor of 8 to obtain a scoring mask s_t of the original input size of 512×2048. Examples of the outputs s_t is shown in Fig. 3. To generate accurate segmentation masks of moving objects, FC-VGGNet is trained on the 4 classes that includes both moving and static objects. For each of the extracted objects from the final output segmentation masks m_t of FC-VGGNet, we compute the element-wise product with the scoring mask s_t. We classify an object as moving if the mean of the scores exceeds a threshold τ in the given object area $\mathcal{A}$, yielding

$$\frac{1}{n} \sum_{(x,y)\in\mathcal{A}} s_t(x,y) \cdot m_t(x,y) > \tau, \tag{3}$$

where n is the number of elements in $\mathcal{A}$. The value of the threshold τ is discussed in Section 4.2.

3.2. Inpainting

After obtaining the segmentation masks from the moving object segmentation, we remove the detected objects. With respect to previous approaches, our method requires inpainting from other views that serve as a prior. Our input images are also larger (512×512 pixels) compared to [28] (256×256) and hence we added an additional strided convolution layer in the generator and two discriminators. Our inputs consist of an RGB image with holes I_t^h, the binary mask with the holes B_t^h obtained from the moving object detection and RGB images $\hat{I}^r_{t'\to t}$ that are projected from the other views. The images $\hat{I}^r_{t'\to t}$ are obtained from reprojection after removal of moving object from other views (I^r denoting removed objects) in the regions where holes are present in the binary mask B_t^h. This is shown in the third row of the Fig. 4. The final input to the generator is a 16-channel input.

The 16-channel input is fed to the coarse network from [28] to produce the final output. We follow a similar approach, however, the refinement network is not used as no performance improvement is observed. This occurs as the input contains sufficient prior information which alleviates the need to produce a coarse output. We also observe that we need to train for longer period of time with a single-stage network to reach the performance of the two-stage network. We follow the same strategy in [32, 28] of training multiple discriminators to ensure both local and global consistency. Hence, the output from the network is fed to a global and local discriminator. For training, we use the improved WGAN objective [36] along with a spatially discounted reconstruction loss [28]. The final training objective $\mathcal{L}$ with a generator G and discriminator networks D^c (where c denotes the context, global or local discriminator) is expressed :

$$\mathcal{L} = \min_G \max_{D^c} \mathcal{L}^h_{\text{WGAN-GP}}(G, D^c) + \mathcal{L}^d_{L1}(G, I_t), \tag{4}$$

where $\mathcal{L}^h_{\text{WGAN-GP}}$ is the WGAN adversarial loss with gradient penalty applied to pixels within the holes and $\mathcal{L}^d_{L1}$ is the spatially discounted reconstruction loss. We follow the same WGAN adversarial loss with gradient penalty in [28] for our problem,

$$\begin{aligned} \mathcal{L}^h_{\text{WGAN-GP}}(G, D) &= \mathbb{E}_{\tilde{\mathbf{x}}\sim\mathbb{P}_f}[D(\tilde{\mathbf{x}})] - \mathbb{E}_{\mathbf{x}\sim\mathbb{P}_r}[D(\mathbf{x})] \\ &+ \lambda\, \mathbb{E}_{\hat{\mathbf{x}}\sim\mathbb{P}_{\hat{\mathbf{x}}}}\left[(\|\nabla_{\hat{\mathbf{x}}} D(\hat{\mathbf{x}}) \odot (\mathbf{1} - \mathbf{m})\|_2 - 1)^2\right], \end{aligned} \tag{5}$$

where $\nabla_{\hat{\mathbf{x}}} D(\hat{\mathbf{x}})$ denotes the gradient of $D(\hat{\mathbf{x}})$ with respect to $\hat{\mathbf{x}}$ and $\odot$ denotes the element-wise product. The samples $\mathbf{x}$

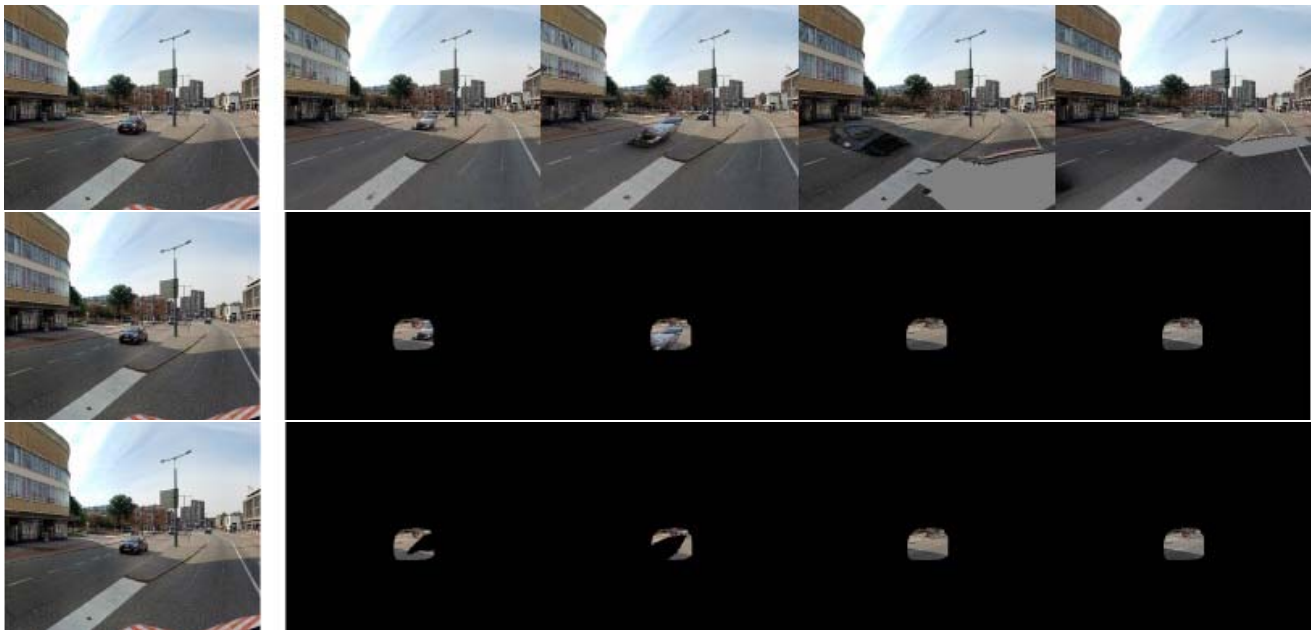

Figure 4: Images in the first column are the input images (I_t) at time t. The first row contains images $\hat{I}'_{t'\to t}$ for $t' \in \{-2,-1,+1,+2\}$ that is projected to the view point of I_t. Results in the second row are obtained after removal of regions around the area of interest. Finally, the third row ($\hat{I}^r_{t'\to t}$) is obtained after removal of moving objects from other views.

and $\tilde{\mathbf{x}}$ are sampled from real and generated distributions $\mathbb{P}_r$ and $\mathbb{P}_f$. $\mathbb{P}_f$ is implicitly defined by $\tilde{\mathbf{x}} = G([I_t, B^h_t, \hat{I}^r_{t'\to t}])$ where [,] denotes the concatenation operation and $t' \in \{-2,-1,1,2\}$. The sample $\hat{\mathbf{x}}$ is an interpolated point obtained from a pair of real and generated samples. The gradient penalty is computed only for pixels inside the holes, hence, a mask $\mathbf{1}-\mathbf{m}$ is multiplied with the input where the values are 0 for missing pixels and 1 otherwise. The spatial discounted reconstruction loss [28] $\mathcal{L}^d_{L1}$ is simply weighted $\mathbf{L1}$ distance using a mask $\mathbf{M}$ and is given as

$$\mathcal{L}^d_{L1}(G, I_t) = \|\mathbf{M} \odot G([I_t, B^h_t, \hat{I}^r_{t'\to t}]) - \mathbf{M} \odot I_t\|_1, \quad (6)$$

where each value in the mask $\mathbf{M}$ is computed as γ^l (l is the distance of the pixel to nearest known pixel). We set the gradient penalty coefficient λ and the value γ to 10 and 0.99 respectively as in [36, 28]. Intuitively, the $\mathcal{L}^h_{\text{WGAN-GP}}$ updates the generator weights to learn plausible outputs whereas $\mathcal{L}^d_{L1}$ tries to the reconstruct the ground truth.

4. Experiments

We evaluate our method on the datasets described in the next section. The final results are evaluated using peak signal-to-noise ratio (PSNR), L_1 loss and an image quality assessment survey.

4.1. Datasets

The datasets consists of several high-resolution panoramas and depth maps derived from LiDAR point clouds. Each of the high-resolution panoramas is obtained from a five-camera system that has its focal point on a single line parallel to the driving direction. The cameras are configured such that the camera centers are on the same location, in order to be able to construct a 360° panorama. The parallax-free 360° panoramic images are taken at every 5-meters and have a resolution of 100-megapixels. The images are well calibrated using multiple sensors (GPS, IMU) and have a relative positioning error less than 2 centimeters between consecutive images. The LiDAR scanner is a Velodyne HDL-32E with 32 planes, which is tilted backwards to maximize the density of the measurement. The RGB and LiDAR are recorded together and they are matched using pose graph optimization from several constraints such as IMU, GPS. The point cloud from the LiDAR is meshed and projected to a plane to obtain a depth map.

The **segmentation dataset** consists of 4,000 images of 512 × 512 pixels, 360° panoramas along with their depth maps. The dataset is divided into 70% for training and 30% for testing. Our internal dataset consists of 96 classes of objects out of which 22 are selected for training. The 22 classes are broadly segregated into 4 classes as recording vehicle, pedestrians, two-wheelers and motorized vehicles. The **inpainting dataset** contains of 8,000 images where 1,000 are used for testing. The holes for inpainting have varying sizes (128 × 128 to 384 × 384 pixels) that are placed randomly at different parts of the image. The inpainting dataset will be made publicly available.

4.2. Moving-Object Segmentation

We first train FC-VGGNet on our internal dataset across the pre-mentioned 4 sub-classes described in Section 4.1. Due to high class imbalances (recording vehicle (5.3%), pedestrians (0.05%), two-wheelers (0.09%) and motorized vehicle (3.2%), other objects (91.4%)), the losses are re-weighted inversely proportional to the percent of pixels of each class. We observe the best performance at 160 epochs to obtain a mean IoU of 0.583.

Since large public datasets of moving objects with ground truth are not available, we resort to manual evaluation of the segmentation results. To evaluate the performance of the moving object detection, we select every moving object in 30 random images. We measure the classification accuracy of these extracted moving objects across different layer blocks of FC-VGGNet to determine the best performing layer block. From Fig. 5, we can conclude that extracting features $\mathcal{F}(I_t)$ from convolution layers of the decoder (Layers 6-10) of FC-VGGNet leads to worse classification results than using the convolution layers of the encoder (Layers 1-5). The best results are obtained when the outputs are extracted from the fourth convolution layer. We also conducted experiments with a VGG-16, pretrained on ImageNet dataset. We have found that the best results are extracted from the eighth convolution layer, which produces an output of size $28 \times 28 \times 512$. However, on qualitative analysis, we have found that the features from the encoder layers of FC-VGGNet offer much better performance than VGG features. This is visualized in Fig. 3. The activations are stronger (higher intensities) for moving objects and have lower false positives. This is expected as FC-VGGNet is trained on the same data source as used for testing, whereas VGG is trained on ImageNet.

It is interesting to observe that features from the shallower layers (earlier layers) of FC-VGGNet perform much better than deeper layers (later layers) for moving object detection. This is due to features adapting to the final segmentation mask as the network grows deeper. As we compute the L_1 loss between $\mathcal{F}(I_t)$ and $\mathcal{F}(\hat{I}'_{t+i\to t})$ at deeper layers, the moving object segmentation is more close to the difference between the segmentation outputs of I_t and $\hat{I}'_{t+i\to t}$ (effectively removing overlapping regions), resulting in poor performance. The threshold τ to decide whether an object is moving (as in Eq. (3)), is empirically determined. Surprisingly, the threshold τ has minimal impact on the performance. The mean IoU varies slightly, between $0.76 - 0.8$ for $\tau \in [0.1, 0.9]$. For all the experiments, we set τ to 0.7.

4.3. Inpainting

Initially, we train the inpainting network proposed by Yu *et al.* [28]. However, we do not use the refinement network as we do not observe any performance improvements. As input we supply 16 channels, 5 RGB images and a binary mask. However, no real ground truth is present for an input image, *i.e.* we do not have an image taken at the exact same location without that moving object being there. Re-projected images could serve as ground truth, but have artifacts and a lower visual quality. Therefore, we randomly remove regions from the images (excluding regions of sky and recording vehicle) to generate ground truth. Instead of randomly selecting shapes for inpainting, the removed regions have the shapes of moving objects (obtained from moving object segmentation). The shapes of moving objects are randomly re-sized to different scales so as to learn to inpaint objects of different sizes.

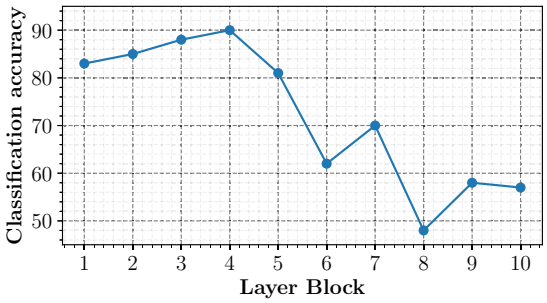

Figure 5: Classification accuracy of extracted objects from moving object segmentation results as moving/non-moving from different layer blocks of FC-VGGNet.

To provide an implicit attention for the inpainting GAN, instead of feeding in the complete reprojected images, we select only pixels from the non-empty regions of the binary mask from the other views. However, simply feeding in selected regions from other views have a drawback, since a moving object that is partially visible in other views, is also projected to the non-empty region. This is undesirable as it causes the inpainting network to learn unwanted moving objects from other views. Therefore, the moving objects from other views are removed prior to projecting pixels from other views. Therefore, the final input for the generator is $[I_t, B_t^h, \hat{I}^r_{t'\to t}]$, where $[a, b]$ denotes the concatenation operation of b after a. Optimization is performed using the Adam optimizer [41] with a batch size of 8 for both the discriminators and the generator. The learning rate is set to 10^{-5} and is trained for 200 epochs. The discriminator-to-generator training ratio is set to 2. The inpainted results after moving object segmentation are shown in Fig. 6.

Evaluation Evaluation metrics such as Inception score (IS) [42], MS-SSIM [43] and Birthday Paradox Test [44] for evaluating GAN models are not suitable for inpainting as inpainting focuses on filling in background rather than its capacity to generate diverse samples. In the case of Fréchet Inception Distance (FID) [45] and IS [42], a deep network is trained such that it is invariant to image transformations and artifacts making it unsuitable for image inpainting task as these metrics have low sensitivities to distortions.

Figure 6: Inpainted results after removing objects obtained from moving object segmentation (bottom row). Input images are shown in the top row. Inpainting in the second and third column have a slight ghosting effect. Participants have an average confidence of 20%, 62.9% and 80% (column 1-3) that it is an inpainted image.

For evaluation, we use both PSNR and L_1 loss comparing the ground-truth image against the inpainted image on a test set of 1000 images. In our case, these metrics are suitable as they measure the reconstruction quality from other views rather the plausibility or diversity of the inpainted content. However, applying reconstruction losses as a evaluation metric favors multi-view based inpainting. As we use multi-view information for inpainting, it is obvious that the results can be improved significantly making it difficult for a fair comparison. Nevertheless, we report the results in PSNR and L_1 losses on the validation set. We obtain a PSNR and L_1 loss of 27.2 dB and 2.5% respectively.

As a final experiment to assess overall quality, we have conducted a survey with 35 professionals within the domain. We have asked the participants to perform a strict quality check on 30 randomly sampled image tiles out of which 15 of them are inpainted after moving object detection. The number of tiles in which a moving object is removed is not revealed to the participants. Each participant is asked to observe an image tile for approximately 10 seconds and then determine if a moving object has been removed from the tile. Participants are also informed to pay close attention to misplaced shadows, blurry areas and other artifacts. The results of the survey is shown in Fig. 7. In total of 1050 responses were collected from 35 participants, 333 (31.7%) responses identified the true positives (inpainted images identified correctly as inpainted), 192 (18.3%) as false negatives (inpainted images not recognized as inpainted), 398 (37.9%) as true negatives (not inpainted and identified as not inpainted) and 127 (12.1%) as false positives (not inpainted but recognized as inpainted). Note that combination of true positives and false negatives are disjoint from the combination true negatives and false positives. The participants have an average confidence of 63.4% $\pm$ 23.8% that a moving object(s) was inpainted in the images where objects were removed (average of responses in blue line of Fig. 7). However, it is interesting to note that in cases when no object is removed, they have a confidence of 24.2% $\pm$ 13.5% stating that an object(s) is removed and inpainted (average of responses in orange line of Fig. 7).

Clearly, in most cases with meticulous observation, participants are able to discern if an object is removed. However, we also observe high deviation in the responses of images where objects are removed and hence we inspect images that have poor scores (high confidence from participants stating that it is inpainted). Few of the worst performing results (confidence higher than 90% on the blue line of Fig. 7) are shown in Fig. 8. We have found that the worst results (Image 7, Image 10 with average confidence of 94.3%, 97.1%) have the strongest artifacts. However, in the other

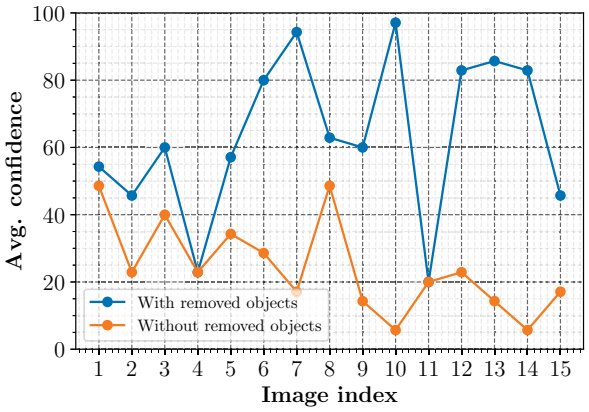

Figure 7: The average confidence per image of survey participants indicating if an image has an inpainted region. The blue line is for images that have objects removed and orange for unaltered images.

cases (average confidence of 82.9%, 85.7% and 82.9%), we note that minuscule errors such as slight variations in edges, lighting conditions, shadows, etc. are the reasons why participants are able to distinguish the inpainted examples. Although the poor cases are reported with high confidence, we believe that such artifacts would be hardly noticed in reality if not explicitly searched. Despite these cases, this framework ensures complete privacy, alleviates blurring artifacts and removes occluded regions which is beneficial for commercial purposes.

5. Discussion

Although the proposed framework is a good alternative to blurring, it is by no means perfect. The moving object segmentation algorithm invokes a few challenges. Since the moving object detection is class agnostic, there are false positives from certain objects such as traffic signs and light poles. The comparison between features from different viewpoints of a traffic sign (front and back) leads to false positives. Similarly, poles might be detected as moving since small camera position errors lead to a minor mismatch of depth pixels during reprojection. However, we are able to suppress these false positives by combining the outputs from FC-VGGNet. Even for a wide range of τ values [0.1 - 0.9], mIoU varies only by 4% ensuring the reliability and robustness of the method. The proposed method may fail when there is an overlap of moving and static objects. For example, a car driving in front of parked vehicles can results in all objects classified as moving or non-moving. However, this can be mitigated by applying instance segmentation.

Limitations Poor results occur in a few cases when a driving vehicle is in the same lane as the recording vehicle (Fig. 8, row 3). The moving object completely occludes all the views making it difficult for multi-view inpainting.

Figure 8: Worst performing results from the survey (high confidence that object is removed). Rows 1-3 with average confidence of 82.9%, 94.3% and 97.1% respectively.

In such a scenario inpainting based on context would be an alternative, however, this does not guarantee a genuine completion of the image. In non-commercial application, this is still a viable solution. Even though few inpainting artifacts such as shadows, slightly displaced edges are visible, we argue that they are still a better alternative to blurring as it ensures complete privacy and far less noticeable artifacts (Fig. 6, column 3 and Fig. 8, row 1). As the method does not explicitly target shadow, this too may reveal privacy-sensitive information in rare cases.

6. Conclusion

We presented a framework that is an alternative for blurring in the context of privacy protection in street-view images. The proposed framework comprises of novel convolutional feature based moving object detection algorithm that is coupled with a multi-view inpainting GAN to detect, remove and inpaint moving objects. We demonstrated through the multi-view inpainting GAN that legitimate information of the removed regions can be learned which is challenging for a standard context based inpainting GAN. We also evaluated overall quality by means of a user questionnaire. Despite the discussed challenges, the inpainting results of the proposed method are often hard to notice and ensures complete privacy. Moreover, the proposed approach mitigates blurring artifacts and removes occluded regions which are beneficial for commercial applications. Although most of the current solutions rely on blurring, we believe that the future of privacy protection lies in the direction of the proposed framework.

References

[1] M. McGee, "Google has stopped street view photography in Germany." https://searchengineland.com/google-has-stopped-street-view-photography-germany-72368. 1

[2] PTI, "Google street view denied permission in India: Heres the reason why." https://indianexpress.com/article/technology/tech-news-technology/googles-street-view-turned-down-by-india-2843618/. 1

[3] M. Chen and L. Denoyer, "Multi-view generative adversarial networks," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 175–188, Springer, 2017. 2

[4] L. Sun, W. Kang, Y. Han, and H. Ge, "Multi-view transformation via mutual-encoding infogenerative adversarial networks," *IEEE Access*, vol. 6, pp. 43315–43326, 2018. 2

[5] A. Frome, G. Cheung, A. Abdulkader, M. Zennaro, B. Wu, A. Bissacco, H. Adam, H. Neven, and L. Vincent, "Large-scale privacy protection in Google street view," in *Computer Vision, 2009 IEEE 12th International Conference on*, pp. 2373–2380, IEEE, 2009. 2

[6] A. Flores and S. Belongie, "Removing pedestrians from google street view images," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, pp. 53–58, IEEE, 2010. 2

[7] P. Agrawal and P. Narayanan, "Person de-identification in videos," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 3, pp. 299–310, 2011. 2

[8] A. Nodari, M. Vanetti, and I. Gallo, "Digital privacy: Replacing pedestrians from google street view images," in *Pattern Recognition (ICPR), 2012 21st International Conference on*, pp. 2889–2893, IEEE, 2012. 2

[9] J. R. Padilla-López, A. A. Chaaraoui, and F. Flórez-Revuelta, "Visual privacy protection methods: A survey," *Expert Systems with Applications*, vol. 42, no. 9, pp. 4177–4195, 2015. 2

[10] C. Sebastian, B. Boom, E. Bondarev, and P. H. N. de With, "LiDAR-assisted Large-scale privacy protection in street-view cycloramas," To appear in *Electronic Imaging*, vol. abs/1903.05598, 2019. 2

[11] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, pp. 91–99, 2015. 2

[12] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object detection via region-based fully convolutional networks," in *Advances in neural information processing systems*, pp. 379–387, 2016. 2

[13] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *European conference on computer vision*, pp. 21–37, Springer, 2016. 2

[14] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431–3440, 2015. 2

[15] V. Badrinarayanan, A. Handa, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling," *arXiv preprint arXiv:1505.07293*, 2015. 2

[16] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834–848, 2018. 2

[17] F. Ferri, M. Gianni, M. Menna, and F. Pirri, "Dynamic obstacles detection and 3D map updating," *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5694–5699, 2015. 2

[18] A. Azim and O. Aycard, "Detection, classification and tracking of moving objects in a 3D environment," *2012 IEEE Intelligent Vehicles Symposium*, pp. 802–807, 2012. 2

[19] H. Cho, Y.-W. Seo, B. V. K. V. Kumar, and R. Rajkumar, "A multi-sensor fusion system for moving object detection and tracking in urban driving environments," *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1836–1843, 2014. 2

[20] J. Yan, D. Chen, H. Myeong, T. Shiratori, and Y. Ma, "Automatic Extraction of Moving Objects from Image and LIDAR sequences," *2014 2nd International Conference on 3D Vision*, vol. 1, pp. 673–680, 2014. 2

[21] A. Takabe, H. Takehara, N. Kawai, T. Sato, T. Machida, S. Nakanishi, and N. Yokoya, "Moving object detection from a point cloud using photometric and depth consistencies," in *Pattern Recognition (ICPR), 2016 23rd International Conference on*, pp. 561–566, IEEE, 2016. 2

[22] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman, "Patchmatch: A randomized correspondence algorithm for structural image editing," *ACM Transactions on Graphics (ToG)*, vol. 28, no. 3, p. 24, 2009. 2

[23] A. Criminisi, P. Pérez, and K. Toyama, "Region filling and object removal by exemplar-based image inpainting," *IEEE Transactions on image processing*, vol. 13, no. 9, pp. 1200–1212, 2004. 2

[24] P. Buyssens, M. Daisy, D. Tschumperlé, and O. Lézoray, "Exemplar-based inpainting: Technical review and new heuristics for better geometric reconstructions," *IEEE transactions on image processing*, vol. 24, no. 6, pp. 1809–1824, 2015. 2

[25] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Nets," in *Advances in neural information processing systems*, pp. 2672–2680, 2014. 2

[26] D. Pathak, P. Krähenbühl, J. Donahue, T. Darrell, and A. A. Efros, "Context Encoders: Feature Learning by Inpainting," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2536–2544, 2016. 2

[27] S. Iizuka, E. Simo-Serra, and H. Ishikawa, "Globally and locally consistent image completion," *ACM Trans. Graph.*, vol. 36, pp. 107:1–107:14, 2017. 2

[28] J. Yu, Z. L. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, "Generative Image Inpainting with Contextual Attention," *CoRR*, vol. abs/1801.07892, 2018. 2, 4, 5, 6

[29] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2242–2251, 2017. 2

[30] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional GANs," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 2

[31] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5967–5976, 2017. 2

[32] R. Uittenbogaard, C. Sebastian, J. Vijverberg, B. Boom, and P. H. N. de With, "Conditional Transfer with Dense Residual Attention: Synthesizing traffic signs from street-view imagery," in *2018 24th International Conference on Pattern Recognition (ICPR)*, pp. 553–559, Aug 2018. 2, 4

[33] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *European Conference on Computer Vision*, pp. 694–711, Springer, 2016. 2

[34] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," *arXiv preprint arXiv:1609.04802*, 2016. 2

[35] G. Zhao, J. Liu, J. Jiang, and W. Wang, "A deep cascade of neural networks for image inpainting, deblurring and denoising," *Multimedia Tools and Applications*, vol. 77, pp. 29589–29604, 2017. 2

[36] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved Training of Wasserstein GANs," in *Advances in Neural Information Processing Systems 30* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), pp. 5767–5777, Curran Associates, Inc., 2017. 3, 4, 5

[37] R. Arandjelović, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "NetVLAD: CNN Architecture for Weakly Supervised Place Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, pp. 1437–1451, June 2018. 4

[38] H. Noh, A. Araujo, J. Sim, T. Weyand, and B. Han, "Largescale image retrieval with attentive deep local features," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3456–3465, 2017. 4

[39] C. Liu, T. Bao, and M. Zhu, "Part-based feature extraction for person re-identification," in *Proceedings of the 2018 10th International Conference on Machine Learning and Computing*, pp. 172–177, ACM, 2018. 4

[40] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations*, 2015. 4

[41] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014. 6

[42] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," in *Advances in Neural Information Processing Systems*, pp. 2234–2242, 2016. 6

[43] J. Snell, K. Ridgeway, R. Liao, B. D. Roads, M. C. Mozer, and R. S. Zemel, "Learning to generate images with perceptual similarity metrics," in *Image Processing (ICIP), 2017 IEEE International Conference on*, pp. 4277–4281, IEEE, 2017. 6

[44] S. Arora, A. Risteski, and Y. Zhang, "Do GANs learn the distribution? some theory and empirics," in *International Conference on Learning Representations*, 2018. 6

[45] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," in *Advances in Neural Information Processing Systems*, pp. 6626–6637, 2017. 6