Empty Cities: Image Inpainting for a Dynamic-Object-Invariant Space

Berta Bescos[1], José Neira[1], Roland Siegwart[2] and Cesar Cadena[2]

Abstract— In this paper we present an end-to-end deep learning framework to turn images that show dynamic content, such as vehicles or pedestrians, into realistic static frames. This objective encounters two main challenges: detecting all the dynamic objects, and inpainting the static occluded background with plausible imagery. The former challenge is addressed by the use of a convolutional network that learns a multi-class semantic segmentation of the image. The second problem is approached with a conditional generative adversarial model that, taking as input the original dynamic image and its dynamic/static binary mask, is capable of generating the final static image. These generated images can be used for applications such as augmented reality or vision-based robot localization purposes. To validate our approach, we show both qualitative and quantitative comparisons against other state-of-the-art inpainting methods by removing the dynamic objects and hallucinating the static structure behind them. Furthermore, to demonstrate the potential of our results, we carry out pilot experiments that show the benefits of our proposal for visual place recognition[1].

I. INTRODUCTION

Dynamic objects degrade the performance of vision-based robotic pose-estimation or localization tasks. The standard approach to deal with dynamic objects consists in detecting them in the images, and further classifying them as not valid information for such purposes. However, we propose to instead modify these images so that the dynamic content is converted realistically into static. We consider that the combination of experience and context allows us to hallucinate, *i.e.*, inpaint, a geometrically and semantically consistent appearance of the static structure behind dynamic objects.

Turning images that contain dynamic objects into realistic static frames reveals several challenges:

1) Detecting such dynamic content in the image. By this, we mean to detect not only those objects that are known to move such as vehicles, people and animals, but also the shadows and reflections that they might generate, since they also change the image appearance.

2) Inpainting the resulting space left by the detected dynamic content with plausible imagery. The resulting

[1]Berta Bescos and José Neira are with the Instituto de Investigación en Ingeniería de Aragón (I3A), Universidad de Zaragoza, Zaragoza 50018, Spain {bbescos, jneira}@unizar.es

[2]Roland Siegwart and Cesar Cadena are with the Autonomous Systems Lab (ASL), ETH Zürich, Zurich 8092, Switzerland {cesarc, rsiegwart}@ethz.ch

[1]All our code has been made available on https://github.com/bertabescos/EmptyCities.

Fig. 1: Images at the same location with different dynamic content such as cars or pedestrians (top row) can be converted into the same static image, *i.e.*, a dynamic-object-invariant space (bottom row).

image would succeed in being realistic if the inpainted areas are both semantically and geometrically consistent with the static content of the image.

The former challenge can be addressed with geometrical approaches if a sequence of images is available. This procedure usually consists in studying the optical flow consistency along the images [1], [2]. In the case in which only one frame is available, deep learning is the approach that excels at this task by the use of Convolutional Neural Networks (CNNs) [3], [4]. These frameworks have to be trained with the previous knowledge of what classes are dynamic and which ones are not. Recent works show that it is possible to acquire this knowledge in a self-supervised manner [5], [6].

Regarding the second challenge, recent image inpainting approaches that do not use deep learning use image statistics of the remaining image to fill in the holes. The work of Telea [7] estimates the pixel value with the normalized weighted sum of all the known pixels in the neighbourhood. While this approach generally produces smooth results, it is limited by the available image statistics and has no concept of visual semantics. However, neural networks learn semantic priors and meaningful hidden representations in an end-to-end fashion, which have been used for recent image inpainting efforts [8]–[11]. These networks employ convolutional filters on images, replacing the removed content with inpainted areas that usually have both geometrical and semantic consistency with the rest of the image.

Both challenges can also be seen as one single task: translating a dynamic image into a corresponding static image. In this direction, Isola *et al.* [12] propose a general-purpose solution for image-to-image translation.

In this paper we present an end-to-end deep learning framework to transform images that have dynamic content into realistic static frames. This can be used for augmented reality, cinematography, and vision-based localization tasks.

It could also be of interest for the creation of high-detail road maps, as well as for street-view imagery suppliers as a privacy measure to replace faces and license plates blurring.

Just like Isola *et al.* [12] succeed in translating images from day to night, aerial to map view, sketches to photos, *etc.*, our paper builds on their work to translate images from a dynamic space into a static one. The main difference between our objective and his is that, while they apply the same translation to the whole image, we keep the static areas of the image almost untouched, and translate the dynamic parts into static ones. We have adapted their framework to our specific task by introducing a new loss that, combined with the integration of a semantic segmentation network achieves the final objective of creating a dynamic-object-invariant space. An example of our pipeline results can be seen in Fig. 1.

II. RELATED WORK

Previous works have attempted to reconstruct the background occluded by dynamic objects in the images with information from previous frames [13]–[16]. When only one such frame is available, the occluded background can only be reconstructed by image inpainting techniques.

Image Inpainting. Among the non-learning approaches to image inpainting, propagating appearance information from neighboring pixels to the target region is the usual procedure [7]. These methods succeed in dealing with narrow holes, where color and texture vary smoothly, but fail when handling big holes, resulting in over-smoothing. Differently, patch-based methods [17] operate by iteratively searching for relevant patches from the image non-hole regions. These approaches are computationally expensive and therefore not fast enough for real-time applications. Moreover, they do not make semantically aware patch selections.

Deep learning based methods usually initialize the image holes with a constant value, and further pass it through a CNN. Context Encoders [11] were among the first ones to successfully use a standard pixel-wise reconstruction loss, as well as an adversarial loss for image inpainting tasks. Due to the resulting artifacts, Yang *et al.* [18] take the result from Context Encoders as input and then propagates the texture information from non-hole regions to fill the hole regions as post-processing. Song *et al.* [19] use a refinement network in which a blurry initial hole-filling result is used as the input, then iteratively replaced with patches from the closest non-hole regions in the feature space. Iizuka *et al.* [10] extend Content Encoders by defining both global and local discriminators, then apply a post-processing. Following this work, Yu *et al.* [9] replaced the post-processing with a refinement network powered by the contextual attention layers. The recent work of Liu *et al.* [8] obtains amazing inpainting results by using partial convolutions.

In contrast, the work by Ulyanov *et al.* [20] proves that there is no need for external dataset training. The generative network itself can rely on its structure to complete the corrupted image. However, this approach usually applies several iterations (∼50000) to get good and detailed results.
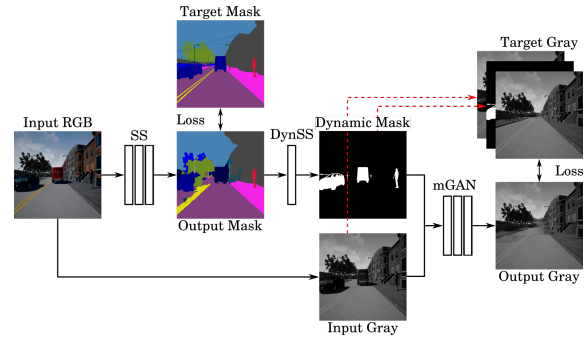

Fig. 2: Block diagram of our proposal. We first compute the segmentation of the RGB dynamic image, as well as its loss against the ground-truth. Then, the dynamic/static binary mask is obtained, and used together with the dynamic image to compute the static image. Its loss is back-propagated until the RGB dynamic image.

Our work does not perform pure inpainting but image-to-image translation with the help of a mask, coming from a semantic segmentation network. This means that we cannot initialize the "holes" with any placeholder values since we do not want to learn that pixels with this particular value have to be transformed. In our case, our input consists of the dynamic original image with the dynamic/static mask concatenated. Different from the other approaches, we perform this task in gray scale instead of in RGB. The motivation for this is that learning a mapping from 1D→1D instead of from 3D→3D is simpler and therefore leads to having less room for wrong reconstructions. In addition, many visual localization applications only need the images grayscale information. Still, as future work, we consider including a RGB version. Moreover, note that using the image-to-image translation approach allows us to slightly modify the image non-hole regions for better accommodation of the reconstructed areas.

III. SYSTEM DESCRIPTION

Fig. 2 shows an overview of our system during training time. First of all, we obtain the pixel-wise semantic segmentation of the RGB dynamic image (SS) and we compute its loss against the ground-truth. Then, the segmentation of only the dynamic objects is obtained with the convolutional layer DynSS. Once we have this dynamic/static binary mask, we convert the RGB dynamic image to gray scale and compute the static image, also in gray scale, with the use of a U-Net, which has been trained in an adversarial way (mGAN). The loss of this generated image, in addition to the appearance $L1$ loss, are back-propagated until the RGB dynamic image, together with the previous computed semantic segmentation loss. All the different stages, as well as the ground-truth generation, are described in subsections III-A to III-D.

A. Data Generation

We have explored our method using CARLA [21]. CARLA is an open-source simulator for autonomous driving research, that provides open digital assets –urban layouts, buildings, vehicles, pedestrians, *etc.*– and supports flexible specification of sensor suites and environmental conditions. We have generated over 12000 image pairs consisting of a target image captured with neither vehicles nor pedestrians,

and a corresponding input image captured at the same pose with the same illumination conditions, but with cars, tracks and people moving around. These images have been recorded using a front and a rear RGB camera mounted on a car. Their ground-truth semantic segmentation has also been captured. By manually selecting those dynamic classes (vehicles and pedestrians), we can easily obtain the ground-truth dynamic/static segmentation too. CARLA offers two different towns that we have used for training and testing, respectively. Our dataset, together with more information about our framework, is available on https://bertabescos.github.io/EmptyCities/.

At present, we are limited to training on synthetic datasets since, to our knowledge, no real-world dataset exists that provides RGB images captured under same illumination conditions at identical poses over long trajectories, with and without dynamic objects.

B. Dynamic-to-Static Translation

A Generative Adversarial Network (GAN) is a generative model that learns a mapping from a random noise vector z to an output image y, $G: z \rightarrow y$ [22]. In contrast, a conditional GAN (cGAN) learns a mapping from observed image x and optional random noise vector z, to y, $G : \{x, z\} \rightarrow y$ [23], or $G : x \rightarrow y$ [12]. The generator G is trained to produce outputs indistinguishable from "real" images by an adversarially trained discriminator D, which is trained to do as well as possible at detecting the generator's "fakes".

The objective of a cGAN can be expressed as

$$\mathcal{L}_{cGAN}(G, D) = \mathbb{E}_{x,y}[\log D(x, y)] + \\ \mathbb{E}_x[\log (1 - D(x, G(x)))], \quad (1)$$

where G tries to minimize this objective against an adversarial D that tries to maximize it. Previous approaches have found it beneficial to mix the GAN objective with a more traditional loss, such as $L1$ or $L2$ distance [11]. The discriminator's job remains unchanged, but the generator is tasked not only with fooling the discriminator, but also with being near the ground-truth in a $L1$ sense, as expressed in

$$G^* = \arg \min_G \max_D \mathcal{L}_{cGAN}(G, D) + \lambda_1 \cdot \mathcal{L}_{L1}(G), \quad (2)$$

where $\mathcal{L}_{L1}(G) = \mathbb{E}_{x,y}[||y - G(x)||_1]$. The recent work of Isola *et al.* [12] shows that cGANs are suitable for image-to-image translation tasks, where the output image is conditioned on its corresponding input image, *i.e.*, it translates an image from one space into another (RGB appearance to drawings, day to night, *etc.*). The main difference between our objective and his is that, while they apply the same mapping to the whole image, we want to keep almost untouched the static areas of the input image, and we want to translate the dynamic parts into plausible static ones. This problem could also be seen as inpainting. However, our method differs in that, in addition to changing the content of the image hole regions, it might also change the non-hole areas for a more realistic output (for example, dynamic objects shadows could also be removed even if unmasked).
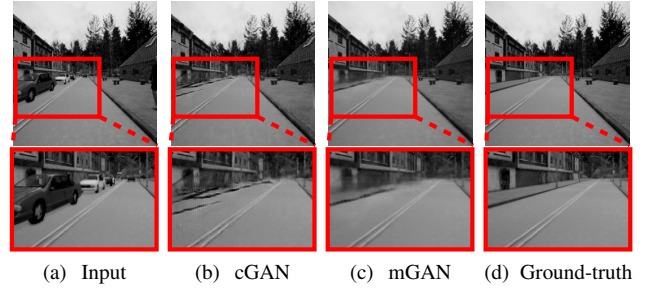

(a) Input   (b) cGAN   (c) mGAN   (d) Ground-truth

Fig. 3: Qualitative results of the improvements achieved by conditioning the discriminator on both input and dynamic/static mask (c), instead of only on the input (a), as cGANs do (b).

It is well known that $L2$ and $L1$ losses produce blurry results on image generation problems, *i.e.*, they can capture the low frequencies but fail to encourage high frequency crispness. This motivates restricting the GAN discriminator to only model high frequency structures. Following this idea, Isola *et al.* [12] adopt a discriminator architecture that classifies each $N \times N$ patch in an image –rather than classifying the image as a whole– as real or fake.

For our objective, object masks are specially considered to re-formulate the training objectives. We adopt a variant of the cGAN that we call mGAN. mGANs learn a mapping from observed image x and binary mask m, to y, $G : \{x, m\} \rightarrow y$. When applying this, we see that the dynamic objects in the image have been inpainted with high frequency texture but there are many artifacts (see Fig. 3b). One of the reasons is that, in most of the training images the relationship between the static and dynamic regions sizes is unbalanced, *i.e.*, static regions occupy usually a much bigger area. This leads us to believe that the influence of dynamic regions on the discriminator response is significantly reduced. As a solution to this problem, we propose to change the discriminator loss so that there is more emphasis on the main areas that have to be inpainted, according to

$$\mathcal{L}_{mGAN}(G, D) = \mathbb{E}_{x,y}[\log D_m(x, m, y)] + \\ \mathbb{E}_x[\log (1 - D_m(x, m, G(x, m)))], \quad (3)$$

where $D_m(x, m, y) = D(x, y) \otimes (1 + m \cdot (\gamma - 1))$. The operator $\otimes$ means the element-wise matrix product, and the parameter γ is a scalar that has been set to 2. A greater γ value leads to better inpainting results in the masked areas, but the quality of the unmasked ones is compromised. A smaller γ value has very little effect on the results with regard to the original discriminator setup. A good trade-off between the emphasis given to the masked compared to the unmasked regions is obtained with $\gamma = 2$. Fig. 3b shows our output if the discriminator is conditioned only on the input, in contrast with the dicriminator conditioned on both the input and the mask (Fig. 3c). The last one shows more realistic results. This training procedure is diagrammed in Fig. 4.

C. Semantic Segmentation

Semantic Segmentation (SS) is a challenging task that addresses many of the perception needs of intelligent vehicles in a unified way. Deep neural networks excel at this task, as
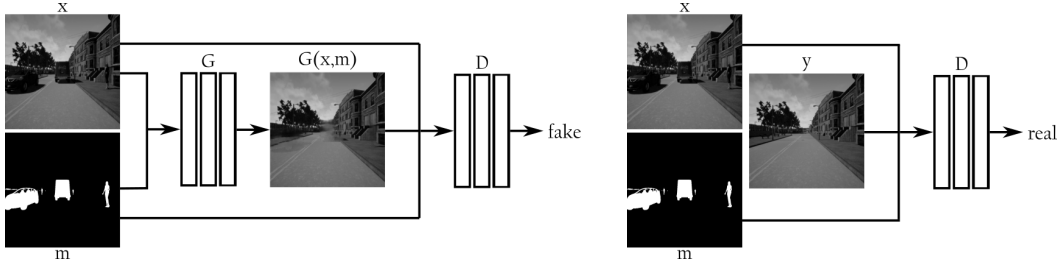
Fig. 4: The discriminator D has to learn to differ between the real images (y) and the images produced by the generator ($G(x,m)$). D is conditioned by both the dynamic/static binary mask m and the input image x to make a better decision about the origin of the image.

they can be trained end-to-end to accurately classify multiple object categories in an image at pixel level. However, few architectures have a good trade-off between high quality and computational resources. The recent work of Romera *et al.* [4] (ERFNet) uses residual connections to remain efficient while retaining remarkable accuracy.

Romera *et al.* [4] have made public some of their trained models [24]. We use for our approach the ERFNet model with encoder and decoder both trained from scratch on Cityscapes train set [25]. We have fine tuned their model to adjust it to our inpainting approach by back-propagating the loss of the semantic segmentation $\mathcal{L}_{CE}(SS)$, calculated using the class weights they suggest, w, and the adversarial loss of our final inpainted model $\mathcal{L}_{mGAN}(G, D)$. The SS network's job can be therefore expressed as:

$$SS^* = \arg \min_{SS} \max_D \mathcal{L}_{mGAN}(G, D) + \lambda_2 \cdot \mathcal{L}_{CE}(SS), \quad (4)$$

where $\mathcal{L}_{CE}(SS) = w[class] \cdot (log(\sum_j \exp(y_{SS}[j])) - y_{SS}[class])$. Its objective is to produce an accurate semantic segmentation y_{SS}, but also to fool the discriminator D.

D. Dynamic Objects Semantic Segmentation

Once the semantic segmentation of the RGB image is done, we can select those classes known to be dynamic (vehicles and pedestrians). This has been done by applying a *SoftMax* layer, followed by a convolutional layer with a kernel of $n \times 1 \times 1$, where n is the number of classes, and with the weights of those dynamic and static channels set to w_{dyn} and w_{stat} respectively. w_{dyn} and w_{stat} are calculated following $w_{dyn} = \frac{n - n_{dyn}}{n}$ and $w_{stat} = -\frac{n_{dyn}}{n}$, where n_{dyn} stands for the number of dynamic existing classes.

The consequent output passes through a $Tanh$ layer to obtain the wanted dynamic/static mask. Note that the defined weights w_{dyn} and w_{stat} are not changed during training time.

A possible extension of this work would contain a greater list of dynamic objects, such as construction sites, posters or temporary festival booths. This is not included at this time due to the lack of availability of training data.

IV. EXPERIMENTAL RESULTS

A. Main Contributions

Here we report the improvements achieved by using, for our particular case, gray-scale instead of RGB images. We also show how the error drops down when using a generator G that learns a mapping from observed image x and binary mask m to y, $G: \{x, m\} \rightarrow y$, instead of a mapping

Experiment	$G(x)_{RGB}$ $D(x,y)_{RGB}$	$G(x)$ $D(x,y)$	$G(x,m)$ $D(x,y)$	$G(x,m)$ $D(x,m,y)$
$L1(\%)$	2.27	1.87	1.21	**0.97**
$L1_{in}(\%)$	9.87	9.17	6.69	**6.00**
$L1_{out}(\%)$	2.02	1.61	1.00	**0.78**

TABLE I: Quantitative evaluations of the performance of our contributions in the inpainting task on the test synthetic images.

from image x to y, $G: x \rightarrow y$. Furthermore, we report how conditioning the discriminator on both the input and the binary mask $D(x, m, y)$, instead of on only the input $D(x, y)$, helps getting better results, see Table I.

The existence of many possible solutions renders difficult to define a metric to evaluate image inpainting [9]. Nevertheless, we follow previous works and report the $L1$ error. Using RGB images usually leads to obtaining inpainting efforts on colorful areas such as cars, but also road signals and traffic lights, that we certainly want to keep untouched. The main improvement carried out by working in gray scale is in the unmasked areas $L1_{out}$. By using the mask to train both the generator G and the discriminator D, we obtain a more accurate static-to-dynamic translation in both the masked ($L1_{in}$) and unmasked regions. Results reported from now on are obtained with $G(x, m)$ and $D(x, m, y)$, *i.e.*, mGANs.

B. Inpainting Comparisons

We compare qualitatively and quantitatively our "inpainting" method with three other approaches:

- **Geo:** a state-of-the-art non-learning based approach [7].
- **Lea1**, **Lea2**: two deep learning based methods [9], [10].

Since both **Lea1** and **Lea2** are methods conceived for general inpainting purposes, we directly use their released models [9], [10] trained on the Places2 dataset [26]. We provide them with the same mask than to our method to generate the holes in the images. We evaluate qualitatively on the 3000 images from our synthetic test dataset, and on the 500 validation images from the Cityscapes dataset [25]. We can see in Figs. 5 and 6 the qualitative comparisons on both datasets respectively[2]. Visually, we observe that our method obtains a more realistic output. Also, it is the only one capable of removing the shadows generated by the dynamic objects even though they are not included in the dynamic/static mask (Fig. 5 row 2). The utilized masks

---

[2]Results generated with both inpainting methods **Lea1** and **Lea2** have been generated with the color images at a 256×256 resolution and then converted to gray scale for visual comparison with our network's output.

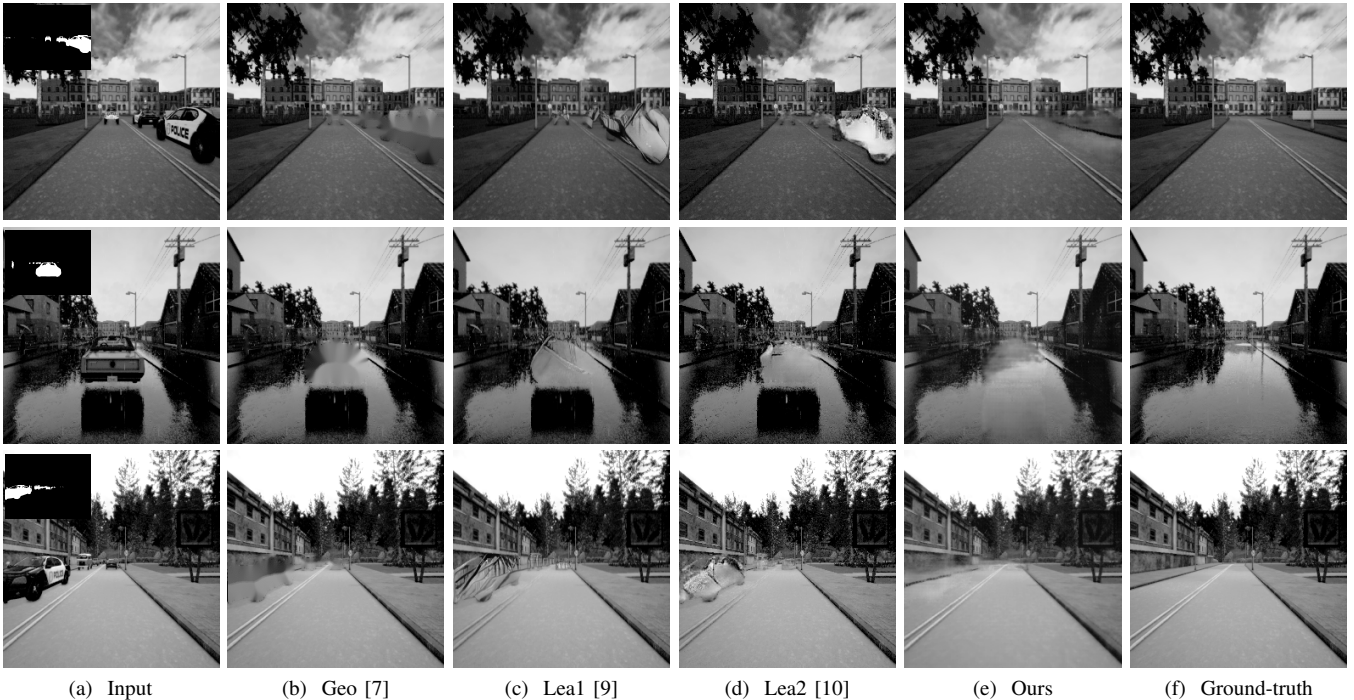| (a) Input | (b) Geo [7] | (c) Lea1 [9] | (d) Lea2 [10] | (e) Ours | (f) Ground-truth |

Fig. 5: Qualitative comparison of our method (e) against other inpainting techniques (b), (c), (d) on our synthetic dataset.

Experiment	Geo [7]	Lea1 [9]	Lea2 [10]	Ours
$L1_{in}(\%)$	6.66	10.45	10.49	**6.00**

TABLE II: Quantitative results of our method against other inpainting approaches in our CARLA dataset.

are included in the images in Fig. 5a. Table II shows the quantitative comparison of our method against **Geo**, **Lea1** and **Lea2** on our CARLA dataset. For a fair comparison we only report the $L1$ error within the mask $L1_{in}$. Reporting the non-hole regions error would be unfair since the other methods are not conceived to notoriously change them.

Regarding the Cityscapes dataset evaluation, quantitatively measuring the performance of the different methods is not possible since ground-truth does not exist. In view of these results, we claim that our approach outperforms both qualitatively and quantitatively the other methods in such task.

C. Transfer to Real Data

Models trained on synthetic data can be useful for real world vision tasks [27]–[30]. Accordingly, we provide a preliminary study of synthetic-to-real transfer learning using data from the Cityscapes dataset [25], which offers a variety of real-world environments similar to the synthetic ones.

When testing our method on real data, we see qualitatively that results are not as good as with the CARLA images (Fig. 6e). This happens because such data has different statistics than the real one, and therefore cannot be easily used. The combination of real and synthetic data is possible during training despite the lack of ground-truth static real images. In the case of the real images, the network only learns the texture and the style of the static real world by encoding its information and decoding back the original image non-hole regions. The synthetic data is substantially

more plentiful and has information about the inpainting process. The rendering, however, is far from realistic. Thus, the chosen representation attempts to bridge the reality gap encountered when using simulated data, and to remove the need for domain adaptation. Fig. 6f shows how adding real images in the training process leads the testing in real data to give slightly better results[3]. Still, the results are not as accurate/realistic as the ones obtained with CARLA images.

Differently, using a CycleGAN type approach [31] would allow us to work with real-world imagery and hence delete this domain adaptation requirement. This approach would learn our desired mapping in the absence of paired images.

D. Visual Localization Experiments

We believe that the images generated by our framework have a potential use for visual localization tasks. Even though utilizing only the static parts of images would also bring benefits to localization systems such as ORB-SLAM [32], DSO [33], SVO [34], *etc.*, they would require modifications, as for example in DynaSLAM [13] among others [1], [35]. Using inpainted images rather than just masked images allows us to use whichever localization system with no modification. This is a remarkable strength of our framework. As a proof of concept, we conduct three additional experiments.

First, we generated a CARLA dataset consisting of 20 different locations with 6 images taken per location. These 6 images show a different dynamic objects setup (Fig. 1). Then the global descriptors (from an off-the-shelf CNN [36]) computed from the different versions of the same location were compared. The euclidean distance between the descriptors of the scenes with dynamic objects was always greater than

---

[3]We perform extensive data augmentation –Gaussian blur, Gaussian noise, brightness, contrast and saturation– to avoid overfitting to the synthetic data.

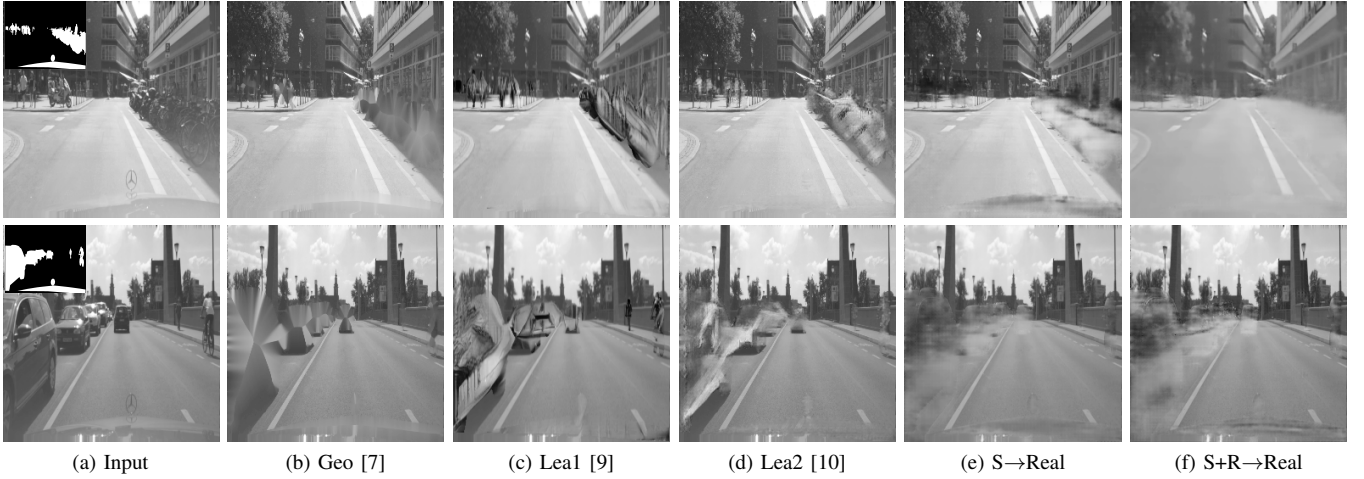| (a) Input | (b) Geo [7] | (c) Lea1 [9] | (d) Lea2 [10] | (e) S→Real | (f) S+R→Real |

Fig. 6: Qualitative comparison of our method (e), (f) against other image inpainting approaches (b), (c), (d) on the Cityscapes validation dataset [25]. (e) shows our results when the training images are all synthetic (S). Albeit, (f) shows our results when real images from the Cityscapes dataset have been incorporated into our training set together with the synthetic CARLA images with a ratio of 1/10 (S+R).

| (a) Ref | (b) Query | (c) Empty Ref | (d) Empty Query |

Fig. 7: (a) and (b) show the same location at different times with different viewpoints, weather conditions and dynamic objects setups [37]. The place recognition work by Olid *et al.* [36] fails to match them as the same place. However, it succeeds in matching them when our framework is previously employed –(c) and (d)–.

that of the images after dynamic removal and inpainting. A 32% average reduction in the distance was observed.

In the second experiment, we generated 6 CARLA images at 6 different locations with a very similar vehicle setup. With the same global descriptor used in the previous experiment, we compared the distances between all possible image pair combinations. Then, we obtained the inpainted images with our framework, and computed the same distances. We repeated this experiment 4 times varying the vehicle setup used and saw that the mean distance of the inpainted sets was higher than that of the original images by 65%.

The third experiment was conducted with real world images from the SVS dataset [37]. We performed place recognition [36] with both the original images, and the ones processed by our framework. In the first case, this task was successful in 58% of the cases, whereas with our images the success rate was of 67%. Fig. 7 shows a case in which our framework makes place recognition successful. Even though the inpainting algorithm is not perfect and might introduce false appearance, the two images global descriptors are closer with the fake static images than with the dynamic ones.

In view of these results, our framework brings closer images from the same place with different dynamic objects while pulling apart images from different places but with similar dynamic objects. We are confident that localization and mapping systems could benefit from these advances.

Also, we expect similar methods to show comparable improvements by incorporating our proposal. An extended version of this work would include such inquiries. Furthermore, a strong benefit of our approach is that such methods would require no modification to work with our processed images.

E. Timing Analysis

Reporting our framework efficiency is crucial to judge its suitability for robotic tasks. The end-to-end pipeline runs at 50 fps on a nVidia GeForce GTX 1070 8GB with images of a 256×256 resolution. Out of the 20 ms it takes to process one frame, 18 ms are invested into obtaining its SS, and 2 ms are used for the inpainting task. Other than to deal with dynamic objects, the SS may be needed for many other tasks involved in automatic navigation. In such cases, our framework would only add 2 extra ms per frame. Based on our analysis, we consider that the inpainting task is not the bottleneck, even though higher resolution images may be needed.

V. CONCLUSION

We have presented an end-to-end deep learning framework that takes as input an RGB image from a city environment containing dynamic objects such as cars, and converts it into a gray realistic image with only static content. For this objective, we develop mGANs, an adaptation of generative adversarial networks for inpainting problems. The provided comparison against other state-of-the-art inpainting methods shows that our approach performs better. Also, our approach has a feature that makes it different from other inpainting methods: areas of the non-hole image can be changed for the objective of a more realistic output. Further experiments show that visual localization and mapping systems can benefit from our advances without any further modification.

Future work might include, among others, converting the resulting static images from gray scale to color [38], [39]. Also, exploiting higher-resolution models would be convenient for robotic labours, as well as carrying out more research efforts on its transferability to the real world domain.

REFERENCES

[1] Y. Wang and S. Huang, "Motion segmentation based robust rgb-d slam," in *Intelligent Control and Automation (WCICA), 2014 11th World Congress on*, pp. 3122–3127, IEEE, 2014.

[2] P. F. Alcantarilla, J. J. Yebes, J. Almazán, and L. M. Bergasa, "On combining visual SLAM and dense scene flow to increase the robustness of localization and mapping in dynamic environments," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pp. 1290–1297, IEEE, 2012.

[3] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Computer Vision (ICCV), 2017 IEEE International Conference on*, pp. 2980–2988, IEEE, 2017.

[4] E. Romera, J. M. Alvarez, L. M. Bergasa, and R. Arroyo, "ERFNet: Efficient Residual Factorized ConvNet for Real-Time Semantic Segmentation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 1, pp. 263–272, 2018.

[5] D. Barnes, W. Maddern, G. Pascoe, and I. Posner, "Driven to distraction: Self-supervised distractor learning for robust monocular visual odometry in urban environments," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1894–1900, IEEE, 2018.

[6] G. Zhou, B. Bescos, M. Dymczyk, M. Pfeiffer, J. Neira, and R. Siegwart, "Dynamic objects segmentation for visual localization in urban environments," *arXiv preprint arXiv:1807.02996*, 2018.

[7] A. Telea, "An image inpainting technique based on the fast marching method," *Journal of graphics tools*, vol. 9, no. 1, pp. 23–34, 2004.

[8] G. Liu, F. A. Reda, K. J. Shih, T.-C. Wang, A. Tao, and B. Catanzaro, "Image inpainting for irregular holes using partial convolutions," in *The European Conference on Computer Vision (ECCV)*, 2018.

[9] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, "Generative Image Inpainting with Contextual Attention," *arXiv preprint arXiv:1801.07892*, 2018.

[10] S. Iizuka, E. Simo-Serra, and H. Ishikawa, "Globally and locally consistent image completion," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 4, p. 107, 2017.

[11] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2536–2544, 2016.

[12] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," *CVPR*, 2017.

[13] B. Bescos, J. M. Fácil, J. Civera, and J. Neira, "DynaSLAM: Tracking, Mapping and Inpainting in Dynamic Scenes," *IEEE Robotics and Automation Letters*, 2018.

[14] R. Scona, M. Jaimez, Y. R. Petillot, M. Fallon, and D. Cremers, "StaticFusion: Background Reconstruction for Dense RGB-D SLAM in Dynamic Environments," in *Robotics and Automation (ICRA), 2018 IEEE International Conference on*, Institute of Electrical and Electronics Engineers, 2018.

[15] M. Granados, K. I. Kim, J. Tompkin, J. Kautz, and C. Theobalt, "Background inpainting for videos with dynamic objects and a free-moving camera," in *European Conference on Computer Vision*, pp. 682–695, Springer, 2012.

[16] R. Uittenbogaard, "Moving object detection and image inpainting in street-view imagery," 2018.

[17] A. A. Efros and W. T. Freeman, "Image quilting for texture synthesis and transfer," in *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pp. 341–346, ACM, 2001.

[18] C. Yang, X. Lu, Z. Lin, E. Shechtman, O. Wang, and H. Li, "High-resolution image inpainting using multi-scale neural patch synthesis," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, p. 3, 2017.

[19] Y. Song, C. Yang, Z. L. Lin, H. Li, Q. Huang, and C.-C. J. Kuo, "Image inpainting using multi-scale feature image translation," *CoRR*, vol. abs/1711.08590, 2017.

[20] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Deep image prior," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9446–9454, 2018.

[21] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proceedings of the 1st Annual Conference on Robot Learning*, pp. 1–16, 2017.

[22] 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.

[23] J. Gauthier, "Conditional generative adversarial nets for convolutional face generation," *Class Project for Stanford CS231N: Convolutional Neural Networks for Visual Recognition, Winter semester*, vol. 2014, no. 5, p. 2, 2014.

[24] E. Romera, J. M. Alvarez, L. M. Bergasa, and R. Arroyo, "ERFNet." https://github.com/Eromera/erfnet, 2017.

[25] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3213–3223, 2016.

[26] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: A 10 million image database for scene recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[27] A. Gaidon, Q. Wang, Y. Cabon, and E. Vig, "Virtual worlds as proxy for multi-object tracking analysis," *arXiv preprint arXiv:1605.06457*, 2016.

[28] M. Peris, S. Martull, A. Maki, Y. Ohkawa, and K. Fukui, "Towards a simulation driven stereo vision system," in *Pattern Recognition (ICPR), 2012 21st International Conference on*, pp. 1038–1042, IEEE, 2012.

[29] J. Skinner, S. Garg, N. Sünderhauf, P. Corke, B. Upcroft, and M. Milford, "High-fidelity simulation for evaluating robotic vision performance," in *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, pp. 2737–2744, IEEE, 2016.

[30] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*, pp. 23–30, IEEE, 2017.

[31] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networkss," in *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017.

[32] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "Orb-slam: a versatile and accurate monocular slam system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[33] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 3, pp. 611–625, 2018.

[34] C. Forster, M. Pizzoli, and D. Scaramuzza, "Svo: Fast semi-direct monocular visual odometry," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pp. 15–22, IEEE, 2014.

[35] Y. Sun, M. Liu, and M. Q.-H. Meng, "Motion removal for reliable rgb-d slam in dynamic environments," *Robotics and Autonomous Systems*, vol. 108, pp. 115–128, 2018.

[36] D. Olid, J. M. Fácil, and J. Civera, "Single-view place recognition under seasonal changes," *arXiv preprint arXiv:1808.06516*, 2018.

[37] F. Werburg and J. Civera, "Street View Sequences: A dataset for lifelong place recognition with viewpoint and dynamic changes," *arXiv*, 2018.

[38] Y. Li, M.-Y. Liu, X. Li, M.-H. Yang, and J. Kautz, "A closed-form solution to photorealistic image stylization," *The European Conference on Computer Vision*, 2018.

[39] S. Iizuka, E. Simo-Serra, and H. Ishikawa, "Let there be Color!: Joint End-to-end Learning of Global and Local Image Priors for Automatic Image Colorization with Simultaneous Classification," *ACM Transactions on Graphics (Proc. of SIGGRAPH 2016)*, vol. 35, no. 4, 2016.