Dynamic-GAN: Learning Spatial-Temporal Attention for Dynamic Object Removal in Feature Dense Environments

Christopher M Trombley, Student Member, IEEE Sumit Kumar Das, Member IEEE,
and Dan O Popa, Senior Member, IEEE

***Abstract*— This paper presents an attention-based, deep learning framework that converts robot camera frames with dynamic content into static frames to more easily apply simultaneous localization and mapping (SLAM) algorithms. The vast majority of SLAM methods have difficulty in the presence of dynamic objects appearing in the environment and occluding the area being captured by the camera. Despite past attempts to deal with dynamic objects, challenges remain to reconstruct large, occluded areas with complex backgrounds. Our proposed Dynamic-GAN framework employs a generative adversarial network to remove dynamic objects from a scene and inpaint a static image free of dynamic objects. The novelty of our approach is in utilizing spatial-temporal attention to encourage the generative model to focus on areas of the image occluded by dynamic content as opposed to equally weighting the whole image. The evaluation of Dynamic-GAN was conducted both quantitatively and qualitatively by testing it on benchmark datasets, and on a mobile robot in indoor navigation environments. As people appeared dynamically in close proximity to the robot, results showed that large, feature-rich occluded areas can be accurately reconstructed in real-time with our attention-based deep learning framework for dynamic object removal. Through experiments we demonstrate that our proposed algorithm has about 25% better performance on average, under various circumstances, as compared to the standard benchmark algorithms.**

I. Introduction

Simultaneous Localization and Mapping (SLAM) is an essential task for robots operating in real-world scenarios. In many applications with collaborative robots, dynamic objects and people will frequently, but randomly appear in close proximity to the robot. If the dynamic object covers a small portion of the field of view (FOV), then traditional robust estimation methods can efficiently eliminate such artifacts by treating them as outliers. For example RANSAC has been used as an outlier rejection algorithm for ORB-SLAM [1] and robust cost function [2] for PTAM [3]. ORB or Oriented FAST and Rotated BRIEF is a well-established algorithm for determining such key features in an image [4].

However, if a large portion of the FOV is covered by the dynamic object then many SLAM estimators tend to fail [5] [6]. Recently, there have been attempts to address the dynamic object problem more directly. Zhang *et al.* [7] segmented dynamic objects using RGB optical flow residuals. Due to recent advances in deep learning, semantic segmentation has been increasingly utilized. Yu *et al.* [8] proposed the DS-SLAM methodology that employs SegNet [9] to segment the scene followed by a moving consistency check method to filter out dynamic features. These types of approaches filter out the dynamic content or only utilize the static regions. Although they are beneficial for visual SLAM systems [1], [10], they often require modifications to traditional SLAM algorithms [11], [12], and the information that is being occluded is not used during pose estimation.

Inpainting approaches [13] [14] have been shown to reduce localization and mapping errors. The benefit of inpainting is that it requires no modification to already existing SLAM systems. However, a noted limitation of current inpainting methods is that they are still unable to handle large, complex occluded regions [13] [15]. These are sufficient for some settings; however, indoor mobile robots require the ability to remove large dynamic objects that are occluding an often complex background [15].

Recently, numerous learning and non-learning methodologies for image inpainting have been proposed. Telea *et al.* [16] used contextual information from neighboring pixels to fill in the missing area, while Efros and Freeman [17] proposed patch-based methods. However, patch-based methods are prone to errors and computationally expensive. Deep learning has also recently been employed to remove dynamic objects and inpaint the static background within a SLAM pipeline. Empty cities is a deep learning approach that consists of a segmentation network followed by a generative adversarial network (GAN)[13]. Additionally, another recent method [14] attempts to handle large occlusion by utilizing both high-level cues extracted from semantic segmentation along with fine grained details from edge extraction. SECI-GAN [18] uses multiple views and inpaints the area occluded from the dynamic object based on contextual information. SECI-GAN was motivated by using information from multiple views to eliminate the refinement network.

In summary, translating frames with large, complex occluded areas to static frames includes three notable challenges:

1) **Large Occluded Area**: Large occluded areas are notoriously difficult [19]. This is in-part because there is less spatial information to utilize when inpainting. Furthermore, methods that discard dynamic regions of the image are not effective when there is a large occluded area.

2) **Occluded areas with complex features**: The higher the occluded feature density and complexity, the more difficult it is to accurately reconstruct the occluded area [15].

Christopher Trombley, Sumit Kumar Das, and Dan Popa are with the Louisville Automation and Robotics Institute (LARRI), University of Louisville in Kentucky, USA (e-mail: christopher.trombley@louisville.edu; sumitkumar.das@louisville.edu;dan.popa@louisville.edu). This work was supported by NSF Grants FOW DRL# 2026584 and EPSCOR OIA#1849213.

For example, removing a dynamic object that is covering a wall is an easier task than if the object is occluding a desk full of objects.

3) **Unable to utilize sequential information:** There are many approaches that rely heavily on the sequential image components to remove dynamic objects. If portions of the sequential information is not available, these approaches fail and deep learning approaches must be employed [13].

In this paper we propose a generative model with a spatial-temporal attention mechanism to address these challenges. Traditional convolutional neural networks (CNN) treat each area of an image, or series of images, with equal weights. Transformers [20], on the other hand, have garnered increased popularity due to their performance in natural language processing tasks [21], [22] as well as computer vision [23], [24]. Transformers have a similar encoder-decoder approach similar to Recurrent Neural Networks (RNN) [25]. Recently, there has been increased research attention using transformers instead of traditional convolution-based GANs [26], [27], [28]. As opposed to convolutional neural networks, transformers weight certain spatial and temporal areas of input. This is advantageous for dynamic object removal because the algorithm can weigh areas of the image with dynamic content higher than areas without. The underlying hypothesis of our work is that using an attention mechanism will improve large dynamic object removal with complex backgrounds. This will in turn improve the performance of SLAM algorithms that are affected by dynamic entities.

The contribution of the paper is to present Dynamic-GAN, an end-to-end deep learning system for the removal of dynamic content in indoor static scenes using transformers in conjunction with Generative Adversarial Networks (GAN). We demonstrate that the proposed approach is robust to large, complex occluded areas where other methods fall short. Our approach builds on recent work on spatial and temporal transformers [27] [26] to develop a transformer-based GAN in PyTorch. The GAN was implemented on a mobile robot using ROS, and we compared our approach to both state-of-the-art geometric [12] and learning [38] inpainting methods. The localization and inpainting performance was measured using the TUM dataset [29] and with a mobile robot in our lab. The paper is organized as follows: in section II we introduce the Dynamic-GAN methodology, in section III we discuss the experiments and their results. Finally, in section IV we present conclusions and discuss future work.

II. Algorithm Formulation

An overview of the generative model is shown in Fig. 1. First, pixel level segmentation on the dynamic RGB image is performed in order to segment dynamic objects in the input image. Then, the input image and the segmented dynamic mask is passed to the frame level encoder which transforms the segmented image and original image into feature representations. Spatial and temporal transformers are then applied in the encoding space. These transformers jointly learn spatio-temporal transformations in a lower dimensional space. Then images are then given to the decoder which takes the features and decodes them to output frames. The output produced are static images that can be used for robotics tasks such as visual odometry. In the encoding space, the transformers operate as multi-headed spatial and temporal transformers, which means they operated in parallel attention layers also known as heads [20].

A. Generative Model

A Generative Adversarial Network (GAN) [30] is a generative model that maps a random noise vector z to an output vector $\hat{y}$, $G : z \rightarrow \hat{y}$. This mapping is learned through a training procedure framed as a supervised learning problem. The training procedure consists of two sub-models: a generator, G, and a discriminator, D.

The loss function of the GAN can be expressed as [31]:

$$\mathbf{L}_{GAN} = \mathbb{E}_x[log(D(x))] + \mathbb{E}_z[log(1 - D(G(z)))], \quad (1)$$

where G and D represent the two functions for the generator and discriminator respectively. Function G minimizes the given objective, where as function D acts as adversarially and tries to maximize it. Here, x is the input image vector and z is the random noise vector as defined above. Therefore, $G(z)$ would be equal to $\hat{y}$.

B. Spatial-Temporal Attention

Instead of traditional convolutions, we propose utilizing transformers during training. One of the key components in transformers is their self-attention mechanism. The attention mechanism operates under the assumption that not all pixels should be weighted equally. The attention mechanism in this work captures dependencies both spatially and temporally. A multi-head patch-based attention module was utilized. Different heads of the transformers computes attentions on different patches at different scales. Transformers differ from traditional RNNs by replacing recurrence with multi-head self-attention mechanism. Multi-head attention enables jointly attending information from different representations at different positions. We utilize the attention mechanism of transformer along both the spatial and temporal dimensions. The temporal attention mechanism uses a set of dynamic object embeddings, $h_1^N = \{h_1.h_2, ..., h_N\}$ as input and outputs a set of updated embeddings, ${h'_1}^N = \{h'_1.h'_2, ..., h'_N\}$ considering each dynamic object independently. Here h represents the encoder level feature vector with i representing the i-th input frame out of N number of frames. Also, h' are updated embeddings which are the output of transformers. The self-attention block learns query matrices $\{Q^i\}_{i=1}^N$, key matrices $\{K^i\}_{i=1}^N$, and value matrices $\{V^i\}_{i=1}^N$. Therefore for the i-th frame we have:

$$Q^i = f_Q(\{h_1\}, \{h_2\}, ..., \{h_N\}), \quad (2)$$

$$K^i = f_K(\{h_1\}, \{h_2\}, ..., \{h_N\}), \quad (3)$$

$$V^i = f_V(\{h_1\}, \{h_2\}, ..., \{h_N\}), \quad (4)$$

where f_Q, f_K, f_V are the corresponding query, key, and value functions respectively [20]. Multihead attention for k

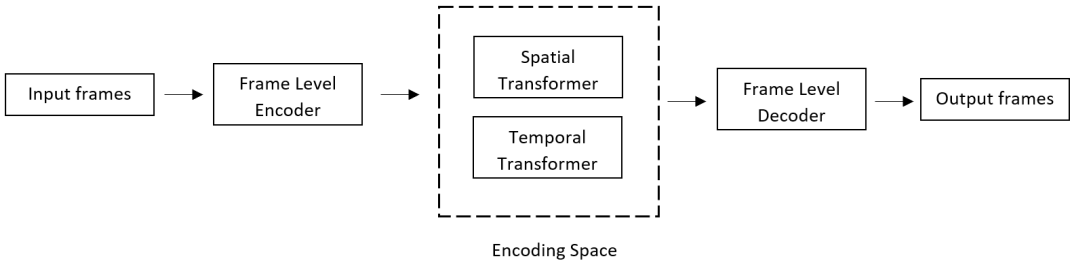

Fig. 1. An overview of Dynamic-GAN. The input frames are given to a convolution based encoder. In lower dimensional space, the encoded representations are given to a multi-head spatial and temporal transformer. An decoder then reconstructs the occluded background and removes dynamic content.

heads is then calculated as:

$$MultiHead(Q^i, K^i, V^i) = f_O([head_j]_{j=1}^k), \quad (5)$$

$$head_j = Att_j(Q^i, K^i, V^i), \quad (6)$$

where f_O is a fully connected layer that merges the k heads and Att_j indexes at head j using a scaling factor d_j as:

$$Att_j(Q^i, K^i, V^i) = \frac{Softmax(Q^i K^{iT})}{\sqrt{d_j}} V^i. \quad (7)$$

In eq. (7) K^{iT} is the transpose of matrix K^i [20], while the $Softmax$ function normalizes a vector to its probability distribution as defined in [32].

C. Loss Function

The loss function used to train the generative model is a weighted combination loss function consisting of an adverserial spatial-temporal loss and L1 loss. The spatial-temporal loss is adopted from [27]. The spatial-temporal loss uses a Temporal Patch GAN as the discriminator. This loss has shown to improve reconstruction quality and spatial-temporal coherence. The Temporal Patch GAN learns to distinguish each spatial-temporal feature as real or fake so the spatial-temporal features can be captured by the generative model. The optimization function of the Temporal Patch Generator is defined by eq.(1).

Previous studies have found it advantageous to combine the traditional adversarial loss with a L1 or L2 distance [13] [33]. Therefore, the generative model not only is encouraged to learn spatial temporal features but also to reconstruct backgrounds as close as possible to the actual background measured by L1 distance. We define the L1 term as:

$$\begin{aligned} \mathbf{L}_{L1} &= \mathbb{E}_x[||y - G(z)||], \\ \implies \mathbf{L}_{L1} &= \mathbb{E}_x[||y - \hat{y}||], \end{aligned} \quad (8)$$

where y is the ground truth image that we wish to generate for the occluded space and $\hat{y}$ is the output of the generator which is an estimation of y. These loss terms are weighted and combined which gives us the resulting loss function:

$$\mathbf{L} = \lambda_1 \times \mathbf{L}_{GAN} + \lambda_2 \times \mathbf{L}_{L1}, \quad (9)$$

where λ_1 and λ_2 are the weights of the respective loss terms.

D. Dynamic-to-Static Translation

Given a series X, with frames containing dynamic objects O_i and corresponding masks P_i with frame index i ranging from $1 \rightarrow N$, we wish to produce a corresponding series Y of frames with the dynamic objects removed. Therefore, we want to learn a mapping $f : X \mapsto Y$ where the conditional distribution of the real data can be estimated by the generated data. The inpainting task is formulated as [27]:

$$p(\hat{Y}|X) = \Pi_{i=1}^N p(\hat{Y}_{i-r}^{i+r}|X_{i-r}^{i+r}, X_u), \quad (10)$$

where r is the temporal radius defining the number of frames and u is the number of uniformly sampled frames.

The transformer-based GAN was trained on the YouTube VOS dataset. YouTube VOS [34] is a large-scale video dataset that contains multiple dynamic objects per frame with over 4,000 videos. The transformer requires an input image along with a mask. To create the mask, we segment the dynamic objects in the training set and compute a mask for the dynamic object. This dataset was used because it is representative of many indoor scenes with large, complex regions occluded.

E. Dynamic Object Segmentation

Finally, the transformer network requires a mask of the segmented dynamic objects along with the RGB image as input. This is a challenging perception task for indoor mobile robots. Deep learning based methods have excelled at semantic segmentation. They can be trained in an end-to-end manner to classify pixels belonging to multiple object categories. Recently, there has been significant work creating deep learning segmentation methods that run at high frame rates which is important for real-time robotic applications [35], [36]. We utilized a deep learning approach to segment dynamic objects at the pixel level. To acquire the dynamic mask, the SegNet [9] segmentation network is utilized due its accuracy and efficiency in terms of both speed and memory. In our case, the network segments humans in the FOV and returns the segmented mask.

III. Experiment Setup and Results

A. Inpainting Evaluation

In this section we present experimental results to evaluate the proposed methodology. First, we performed experiments on benchmark datasets available in the public domain to test the performance of our proposed algorithm. Afterwards, we

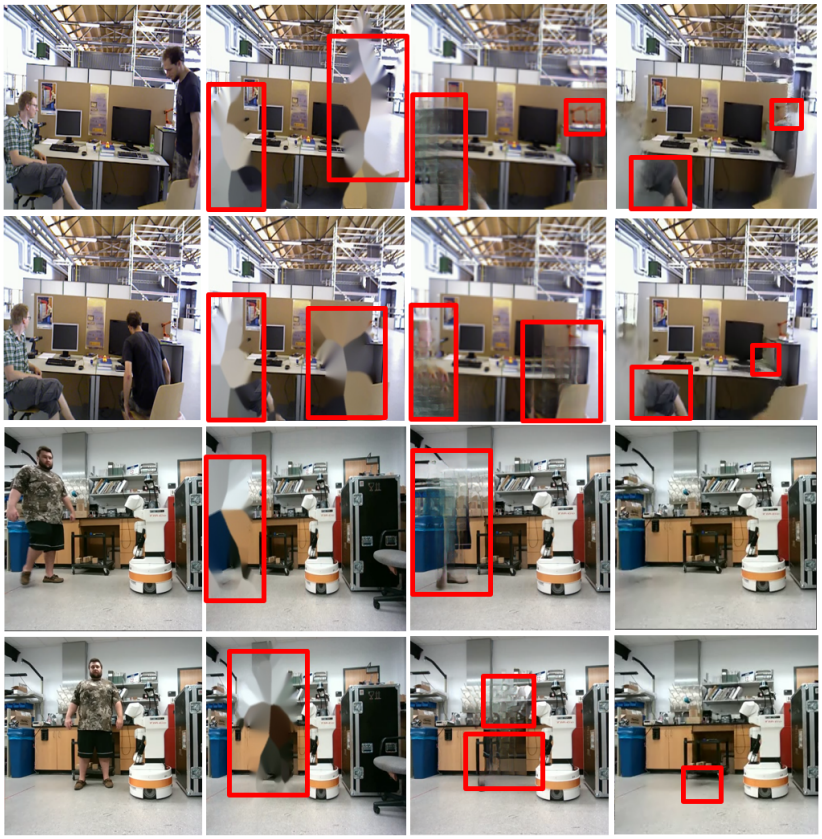

Fig. 2. Qualitative comparison of our method to other inpainting methods for the Frieburg dataset and our mobile robot experiments.

also implemented and evaluated Dynamic-GAN on a mobile robot in our lab. Our approach is compared to two state-of-the-art approaches namely, Telea [16] and AOT [37]. Telea is a geometric based approach that uses information from neighboring pixels to inpaint the occluded area. OpenCV's [38] implementation of Telea was used for the purposes of this study. AOT is a generative learning-based approach that uses a form of spatial attention. AOT differs from our method, which combines spatio-temporal attention, and is one of the best publicly available model for inpainting large regions to the best of our knowledge. For the purposes of testing, the publicly released AOT model was used. Telea was chosen to compare our approach to a non-learning based approach and AOT was chosen to compare it to a learning-based approach for dynamic object removal. All methods implemented were provided with the same mask obtained by SegNet.

Due to many alternatives, choosing a metric to measure inpainting quality is challenging [15]. In this work we utilized L1 error which is a metric commonly used for a fair comparison [15] [13], along with four dynamic sequences from the TUM dataset to evaluate our approach [29]. The L1 error is measured for each frame and then averaged over all the frames. The four sequences consist of two high-dynamic walking and two low-dynamic sitting sequences. The low-dynamic sequences are fr3/sit_static and fr3/sit_halfsphere[29]. The high-dynamic sequences are fr3/walk_static and fr3/walk_halfsphere[29]. We choose these sequences to evaluate the approach's performance on different levels of dynamic content. For a fair comparison, we computed the L1 error only in the masked region of the image since the outer regions remain unchanged.

In our experiments, we set the value of N and u as 10 in eq. (10). Furthermore, λ_1 and λ_2 constants in eq. (9) were assigned the values 1.0 and 0.01, respectively. The size of the input image x in eq. (1) was 256×256 pixels, while the spatial feature size of h in eq. (2) was 64×64.

Table I shows the results for inpainting averaged over 200 frames of the TUM benchmark dataset [29] from each dynamic sequence. We also recorded the number of ORB keypoints in the occluded region. The purpose of these records was to compare the inpainting error with the number of ORB keypoints in Fig. 3. Our qualitative results are shown in Fig. 2. We observe that Telea is unable to inpaint the fine details of large, occluded areas. AOT can better inpaint the occluded area, however, some regions are blurry. Furthermore, Dynamic-GAN is the clearest out of the three methods. Although the background is accurately inpainted by our method, there are a few finer details that are missed such as the drink bottle on the desk.

B. Benchmark Dataset Experiments

The next two sub-sections discuss the experiments conducted to measure localization error. We use ORB-SLAM as the SLAM system for these two experiments. Our method removes the dynamic content and then ORB-SLAM is used to localize the robot's pose. The ROS ORB-SLAM repository

was used for ORB-SLAM implementation [39] [1], while we evaluated our approach on the TUM RGB-D dynamic object dataset [29]. This dataset contains the ground truth camera trajectory which allows us to measure localization error due to dynamic objects. TUM contains multiple sequences with dynamic content (e.g. humans walking around and sitting). The dynamic objects occlude simple backgrounds such as walls and more complex backgrounds such as a cluttered desk. The dynamic objects in the dataset can exceed 50% of the FOV at times. This makes it useful to measure how well our algorithm does with large, occluded areas. We measured localization error using both absolute trajectory error (ATE), and relative pose error (RPE). ATE measures the accuracy of the trajectory globally and RPE measures the drift [29].

TABLE I

QUANTITATIVE RESULTS FOR THE INPAINTING TASK.

RGB-D Sequence	Method		
	Telea [16]	AOT [37]	Dynamic-GAN
fr3/sit_static	42.3	44.7	35.3
fr3/walk_static	46.7	45.3	39.2
fr3/sit_halfsphere	32.2	31.1	24.7
fr3/walk_halfsphere	39.5	32.9	29.8

L1% was reported for four dynamic sequences in the TUM dataset.

The ORB keypoints in the occluded areas were also recorded for later comparison as shown in Fig. 3. The top two rows in Fig. 2 show the qualitative results for the benchmark dataset. We observe that our method obtains a better reconstruction of the occluded area compared to the other approaches. The finer details are better reconstructed, and it indeed appears to better handle large, occluded areas. Table II summarizes the results for six dynamic object sequences. The top row in Fig. 3 depicts the L1 error, ATE, and RPE as a function of ORB feature percentage. ORB feature percentage was measured as the number of ORB keypoints in the occluded area divided by the total number of occluded pixels.

TABLE II

QUANTITATIVE RESULTS FROM THE BENCHMARK DATASET EXPERIMENTS.

RGB-D Sequence	Method		
	Telea [16]	AOT [37]	Dynamic-GAN
fr3/sit_static	2.9	3.3	2.3
fr3/walk_static	17.9	18.2	14.6
fr3/sit_halfsphere	19.8	19.1	12.2
fr3/walk_halfsphere	71.3	78.4	47.9

(a) Absolute Trajectory Error RMSE (cm)

RGB-D Sequence	Method		
	Telea [16]	AOT [37]	Dynamic-GAN
fr3/sit_static	2.3	2.7	1.5
fr3/walk_static	12.9	13.2	11.1
fr3/sit_halfsphere	15.6	14.8	8.7
fr3/walk_halfsphere	37.7	45.1	29.4

(b) Relative Pose Error RMSE (cm/s)

C. Mobile Robot Experiments

We also evaluated our proposed approach on a mobile robot developed in our lab, ARNA [40]. The Adaptive Robotic Nursing Assistant (ARNA) is an indoor mobile manipulator designed for hospital assistance, such as patient walking and patient sitting. The robot consists of an omnidirectional base with an instrumented handlebar, and a 7-DOF robotic arm. It includes an Intel Realsense D455i RGB-D camera attached to the base of the robot. The Dynamic-GAN algorithm was deployed on the robot's Nvidia GeForce GTX 1080 8 GB GPU. In three additional robotic experiments, we used the Realsense® camera on the base of ARNA, and the wheel odometry as the ground truth trajectory to compare the performance of dynamic object removal methods. The robot used a prebuilt map to localize itself within a room. A human sporadically walked through and gesticulated in the FOV of the camera while the robot was moving around the room. The dynamic object covered as much as 60% of the FOV at times. The dynamic content in the image was removed by an inpainting method and ORB-SLAM used the static frame to localize the robot. We measured the localization error when the dynamic object is present in the frame, while the ground truth odometry was calculated using wheel ticks.

TABLE III

QUANTITATIVE RESULTS FROM THE MOBILE ROBOT EXPERIMENTS.

RGB-D Sequence	Method		
	Telea [16]	AOT [37]	Dynamic-GAN
seq1	13.1	14.5	9.7
seq2	19.0	22.6	13.2
seq3	13.2	16.7	9.3

(a) Absolute Trajectory Error RMSE (cm)

RGB-D Sequence	Method		
	Telea [16]	AOT [37]	Dynamic-GAN
seq1	9.8	12.2	7.9
seq2	14.9	11.3	10.6
seq3	10.4	14.5	6.2

(b) Relative Pose Error RMSE (cm/s)

RGB-D Sequence	Method		
	Telea [16]	AOT [37]	Dynamic-GAN
seq1	39.8	41.9	34.1
seq2	44.3	46.1	41.2
seq3	34.1	36.9	30.4

(c) L1% error

Absolute trajectory error and relative pose error were calculated to measure the localization performance. We also calculated L1 error for the occluded region using 200 frames. The number of ORB keypoints in the occluded area were recorded to measure how each method performs with high feature occlusion. The bottom row in Fig. 3 shows the L1 error, APE, and RPE as a function of ORB feature percentage. ORB feature percentage was calculated by taking the number of ORB keypoints in the occluded region divided by the total number of occluded pixels. Fig. 3 shows that our method degrades less than other methods when there are many occluded ORB keypoints. This could be attributed to the combination of spatial and temporal attention. Table III shows the absolute trajectory error and relative pose error. Fig. 4 shows the attention map from the transformer. This shows that the transformer is weighing the area of the dynamic object more than other areas. Fig. 2 also shows the qualitative results in the bottom two rows. We observe that our qualitative results for the mobile robot experiment are even better than the benchmark experiment. This could be

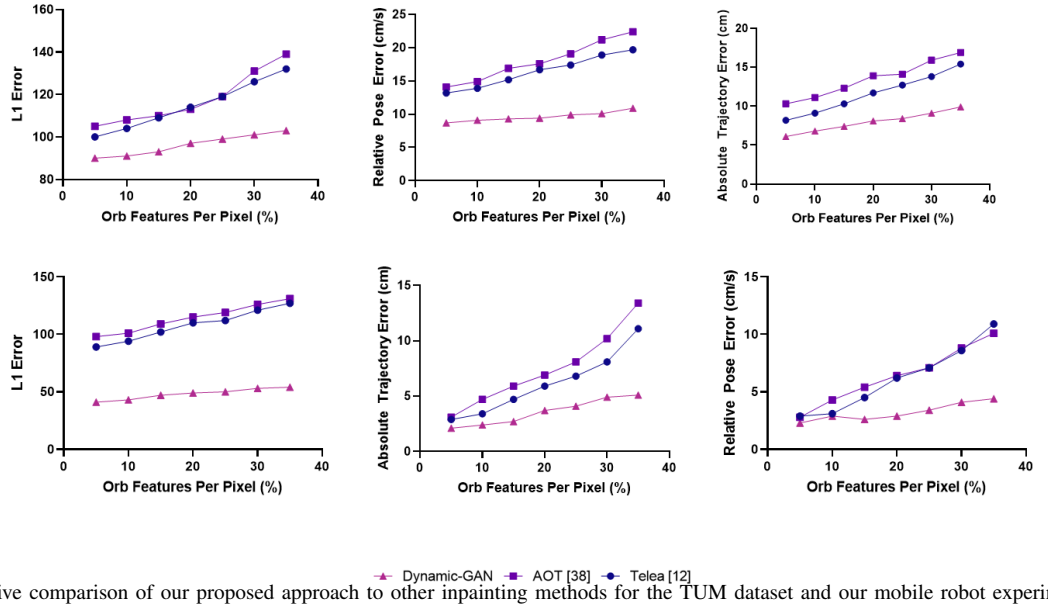

Fig. 3. Quantitative comparison of our proposed approach to other inpainting methods for the TUM dataset and our mobile robot experiment. **Top:** The top shows the results from the experiments of the TUM dataset. The graphs show L1 error, RPE, and ATE as a function of feature density. **Bottom:**The bottom shows the experiment with the mobile robot using the same metrics as a function of feature density. The shows that as the background becomes more complex, attention based inpainting degrades less compared to non-attention based counterparts.

in part due to the fact the person is walking across the frame. Therefore, information from the previous frames can be used to inpaint the occluded area of the current frame, while for the benchmark data, there is a person sitting down moving their leg. In this case, the information from prior frames is not as useful. Results demonstrate that Dynamic-GAN achieves better results when information from the previous frames is used. However, even when this information is not available, the results are still outperform the methods we compared it to. Our method is also able to perform well under different parameters such as lighting conditions, size of person, and camera since the image parameters are different in the benchmark data and the mobile robot experiments, yet the performance is consistent.

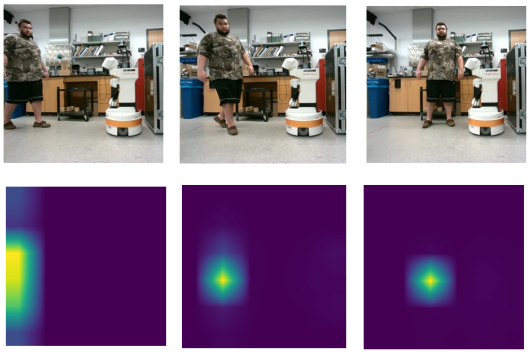

Fig. 4. The top row shows images of a human walking across the frame and the bottom row shows the corresponding attention maps from the spatial-temporal transformers.

D. Timing Analysis

Being able to run at a high frame rate is essential for visual SLAM applications on mobile robots. We tested the time of inference of our proposed approach. To optimize the generative model, we converted it to the ONNX format and further to TensorRT. ONNX stands for Open Neural Network Exchange and is used as a standardized saved model representation [41]. ONNX models can be used with TensorRT to optimize inference on Nvidia hardware. This is opposed to the PyTorch or TensorFlow representations that are not standardized.

The time taken from when a frame is received by the transformer network to when the frame is fully processed with the dynamic object removed was measured. In our study, an image of size 256 x 256 ran at 32 fps.

IV. Conclusion and Future Work

In this paper we presented Dynamic-GAN, an end-to-end deep learning framework that takes a RGB image as input and removes dynamic content to produce a purely static frame free of dynamic objects. Our algorithm uses attention-based deep learning to inpaint areas of an image occluded by dynamic content after the dynamic content is segmented. Comparison against other state of the art geometric and learning approaches shows that our approach performs better, especially with large, occluded areas with complex occluded areas. We observe that our proposed algorithm performed better by about 25% on average as compared to the benchmark Telea and AOT algorithms. The attention-based mechanism allows the approach to focus on certain parts of the image that are more difficult to reconstruct making it robust to complex backgrounds commonly found in indoor environments. Experiments demonstrate that our method improves visual based localization systems such as ORB-SLAM. Importantly, our method is able to improve these systems by removing the dynamic content completely rather than filtering or excluding this content.

Future directions include removing the segmentation step, improving efficiency of Dynamic-GAN in terms of both speed and memory, and testing the algorithm's performance with more dynamic content such as other mobile robots, and several humans in the environment.

REFERENCES

[1] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE Transactions on Robotics*, vol. 33, pp. 1255–1262, 2017.

[2] G. Klein and D. Murray, "Parallel tracking and mapping for small ar workspaces," in *2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, 2007, pp. 225–234.

[3] G. Klein and D. Murrat, "Parallel tracking and mapping for small ar workspaces," in *2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, 2007, pp. 225–234.

[4] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *2011 International Conference on Computer Vision*, 2011, pp. 2564–2571.

[5] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. Leonard, "Simultaneous localization and mapping: Present, future, and the robust-perception age," *ArXiv*, vol. abs/1606.05830, 2016.

[6] W. Dai, Y. Zhang, P. Li, and Z. Fang, "Rgb-d slam in dynamic environments using points correlations," *IEEE transactions on pattern analysis and machine intelligence*, vol. PP, 2020.

[7] T. Zhang, H. Zhang, Y. Li, Y. Nakamura, and L. Zhang, "Flowfusion: Dynamic dense rgb-d slam based on optical flow," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 7322–7328.

[8] C. Yu, Z. Liu, X.-J. Liu, F. Xie, Y. Yang, Q. Wei, and F. Qiao, "Ds-slam: A semantic visual slam towards dynamic environments," *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1168–1174, 2018.

[9] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, pp. 2481–2495, 2017.

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

[11] Y. Wang and S. Huang, "Motion segmentation based robust rgb-d slam," *Proceeding of the 11th World Congress on Intelligent Control and Automation*, pp. 3122–3127, 2014.

[12] F. J. C. J. Bescos, Berta and J. Neira, "DynaSLAM: Tracking, mapping and inpainting in dynamic environments," *IEEE RA-L*, 2018.

[13] B. Bescos, J. Neira, R. Siegwart, and C. Cadena, "Empty cities: Image inpainting for a dynamic-object-invariant space," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 5460–5466.

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

[15] J. Yu, Z. L. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, "Generative image inpainting with contextual attention," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5505–5514, 2018.

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

[17] A. A. Efros and W. T. Freeman, "Image quilting for texture synthesis and transfer," *Proceedings of SIGGRAPH 2001*, pp. 341–346, August 2001.

[18] F. Pinto, A. Romanoni, M. Matteucci, and P. H. Torr, "Seci-gan: Semantic and edge completion for dynamic objects removal," in *2020 25th International Conference on Pattern Recognition (ICPR)*, 2021, pp. 10 441–10 448.

[19] R. Long, C. Rauch, T. Zhang, V. Ivan, and S. Vijayakumar, "Rigidfusion: Robot localisation and mapping in environments with large dynamic rigid objects," *IEEE Robotics and Automation Letters*, vol. 6, pp. 3703–3710, 2021.

[20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017.

[21] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *NAACL*, 2019.

[22] M. Zaheer, G. Guruganesh, K. A. Dubey, J. Ainslie, C. Alberti, S. Ontanon, P. Pham, A. Ravula, Q. Wang, L. Yang *et al.*, "Big bird: Transformers for longer sequences," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[23] A. Srinivas, T.-Y. Lin, N. Parmar, J. Shlens, P. Abbeel, and A. Vaswani, "Bottleneck transformers for visual recognition," *ArXiv*, vol. abs/2101.11605, 2021.

[24] B. Yan, H. Peng, J. Fu, D. Wang, and H. Lu, "Learning spatio-temporal transformer for visual tracking," *ICCV*, vol. abs/2103.17154, 2021.

[25] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *CoRR*, vol. abs/1409.0473, 2015.

[26] C.-H. Lin, E. Yumer, O. Wang, E. Shechtman, and S. Lucey, "St-gan: Spatial transformer generative adversarial networks for image compositing," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[27] Y. Zeng, J. Fu, and H. Chao, "Learning joint spatial-temporal transformations for video inpainting," in *The Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[28] Y. Jiang, S. Chang, and Z. Wang, "Transgan: Two transformers can make one strong gan," 2021.

[29] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of rgb-d slam systems," in *Proc. of the International Conference on Intelligent Robot Systems (IROS)*, Oct. 2012.

[30] 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*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, Eds., 2014.

[31] Y.-L. Chang, Z. Y. Liu, K.-Y. Lee, and W. Hsu, "Free-form video inpainting with 3d gated convolution and temporal patchgan," *In Proceedings of the International Conference on Computer Vision (ICCV)*, 2019.

[32] B. Gao and L. Pavel, "On the properties of the softmax function with application in game theory and reinforcement learning," *arXiv preprint arXiv:1704.00805*, 2017.

[33] D. Pathak, P. Krähenbühl, J. Donahue, T. Darrell, and A. Efros, "Context encoders: Feature learning by inpainting," in *Computer Vision and Pattern Recognition (CVPR)*, 2016.

[34] N. Xu, L. Yang, Y. Fan, D. Yue, Y. Liang, J. Yang, and T. S. Huang, "Youtube-vos: A large-scale video object segmentation benchmark," *ArXiv*, vol. abs/1809.03327, 2018.

[35] I. Alonso, L. Riazuelo, and A. C. Murillo, "Mininet: An efficient semantic segmentation convnet for real-time robotic applications," *IEEE Transactions on Robotics*, vol. 36, no. 4, pp. 1340–1347, 2020.

[36] A. Milioto, L. Mandtler, and C. Stachniss, "Fast instance and semantic segmentation exploiting local connectivity, metric learning, and one-shot detection for robotics," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 5481–5487.

[37] Y. Zeng, J. Fu, H. Chao, and B. Guo, "Aggregated contextual transformations for high-resolution image inpainting," in *Arxiv*, 2020.

[38] Itseez, "Open source computer vision library," https://github.com/itseez/opencv, 2015.

[39] R. Mur-Artal, J. Montiel, and J. D. Tardós, "Orb-slam: A versatile and accurate monocular slam system," *IEEE Transactions on Robotics*, vol. 31, pp. 1147–1163, 2015.

[40] S. Abubakar, S. Das, C. Robinson, M. Saadatzi, M. Logsdon, H. Mitchell, D. Chlebowy, and D. Popa, "Arna, a service robot for nursing assistance: System overview and user acceptability," *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*, pp. 1408–1414, 2020.

[41] J. Bai, F. Lu, K. Zhang *et al.*, "Onnx: Open neural network exchange," https://github.com/onnx/onnx, 2019.