GANLoc: Camera relocalization based on Conditional Adversarial Networks*

Nicolò Valigi[1]

***Abstract*— Camera localization is a challenging and important task for Robotics, one that has historically been tackled with a mixture of appearance based methods ([11]), keypoint correspondence ([12]), and machine learning ([3], [6]).**

In this paper, we incorporate recent advancements in Deep Generative Adversarial Networks (GANs) and geometrical Computer Vision to achieve state-of-the-art accuracy and robustness in the TUM RGBD dataset (at the cost of training time).

We use GANs to perform Image-to-Image translation from the query image to the world scene coordinates of each of its pixels (an approach pioneered by [6]). We then adopt a RANSAC-based robust estimation scheme that recovers the camera pose with centimeter accuracy.

We compare our method to an open source implementation of [6] on the TUM RGBD dataset, and show comparable accuracy with improved robustness, especially to (simulated) image occlusions.

I. INTRODUCTION

Camera localization from single images is an important component of a robust visual navigation pipeline, with important applications in robotics and virtual/augmented reality. Navigation systems based on SLAM or Visual Odometry estimate the camera trajectory by tracking one image frame to the next, and are thus sensitive to abrupt motion or textureless scenes. When tracking is lost, the system needs the ability to recover its pose estimate without external references.

We propose a novel combination of deep generative models and robust geometry estimation techniques that is able to recover the camera pose within a known scene with cm-level accuracy.

The key contributions of this paper are:

- An accurate camera localization system that is highly robust to image occlusions.
- An interpretation of the scene coordinate regression task of [6] as image-to-image translation problem, solved using a Conditional Generative Adversarial Network (GAN) based on [9].
- An exploration of the trade-offs between training time and accuracy on the TUM RGBD dataset.

II. RELATED WORK

Traditionally, the camera relocalization problem has been tackled with either *keyframe* or *keypoint* methods. Keyframe-based approaches compute global image descriptors and use them to look up the query image in a database of known frames with their corresponding ground truth poses. For example, [11] uses a *bag of words* approach to collect multiple SIFT descriptors into per-image binary vectors that can be efficiently stored in a tree-like structure. In general, the keyframe approach suffers from a relative inability to generalize to unseen viewpoints.

On the other hand, *keypoint* methods rely on 2D-to-3D correspondences between interest points in the image and the 3D world map, and are thus more robust to viewpoint changes as long as enough matches are found. Often, these methods use RANSAC schemes to filter out outliers. The weakness of keypoint-based approaches lies in their limited scalability to large environments in terms of computing costs and ability to resolve perceptual ambiguities in the case of repetitive structures. [12] and [13] are examples of successful implementations of this approach.

More recently, [6] proposed a novel method where a regression forest (SCoRe forest) learns a mapping from RGBD patches in the image to their respective Cartesian coordinates in the world scene. These 3D points are used to compute the camera pose using Kabsch's algorithm and RANSAC. This approach features state of the art accuracy, while doing away with the complex feature extraction and description steps. Compared to keypoint or keyframe methods, the main disadvantage is the lengthy training procedure required to localize in new environments ([8] proposed a method for on-the-fly learning). Further development of [6] aimed at improving the expressiveness of the model by using mixtures of Gaussians, substantially improving robustness ([7]).

After the major successes of Deep Neural Networks in image classification, another line of work explored their use for the camera relocalization task. [3] implemented an end-to-end trainable system that learns to directly regress from raw image pixels to the 3D camera pose. However, even in light of its subsequent refinements ([4], [5]), the accuracy of this model is roughly an order of magnitude worse than keypoint-based approaches (e.g. *0.5m* within a *3m* workspace in the 7-Scenes dataset).

Given the amount of labeled data that is generally available for this type of problems, it is intuitively evident how end-to-end methods lack the ability to learn the geometric invariants that the computer vision community has discovered over the course of several decades. [2] partially addresses this issue and demonstrates how CNNs can perform better than random forests in the coordinate prediction task of [6].

Our approach aims to bridge the gap between the modeling power of deep neural networks and the geometric understanding that seems to be the key to good performance.

*This work was not supported by any organization

[1] nico@nicolovaligi.com

Fig. 1. The generative network learns to map from RGB images to 3D coordinates that can be processed through geometric algorithms to recover the camera pose even after extreme occlusion. From left to right, the RGB frame, the Cartesian scene coordinates ground truth, and GAN outputs after 10, 30, and 70 epochs of training.

To achieve this goal, we adopt a state-of-the-art generative model to estimate world coordinates for each pixel in the image, and then use the same geometric pipeline of [6] to robustly recover the camera pose.

III. GENERATIVE ADVERSARIAL NETWORKS

Core to our method is the Conditional Generative Adversarial Network (CGAN) that, following the scheme of [6], learns to predict Cartesian scene coordinates given the RGB values of pixels in the captured frame. This section outlines the motivation between this choice, and compares our GAN to other commonly-used alternatives.

As explored in [2], we would expect Convolutional Neural Networks (CNN) to perform well on the coordinate regression task. Indeed, work on this paper started with a CNN based on the Fully Convolutional Network of [19]. As commonly done for regression tasks, we tried training the network using a pixel-wise L2 loss, and obtained disappointing, blurry results that were not useful for localization. This is consistent with the literature ([14], [10]), and motivated the search for a more capable model.

Since their introduction in [20], GANs have been successfully employed in a variety of tasks, like generating photo-realistic human faces ([21]), super-resolution ([22]), Image-to-Image translation ([23]), and unsupervised representation learning ([24]).

In their most basic incarnation, GANs learn a generative model to sample from an arbitrary data distribution $p(x)$. Their architecture ([10]) is based on pairing a *generator* model $G(z)$, which maps a random noise vector z to the distribution p, and a *discriminator* model $D(x)$ which decides if the sample x belongs to the data distribution. By training the generator and discriminator in lockstep, the GAN learns to tell real and generated images apart, while developing a generative model that can be used to sample from the data distribution. GANs have been shown to produce sharper samples than autoencoder architectures trained on pixel-wise losses, thanks to their ability to develop an implicit loss function that is adequate for the data.

In this work, we employ a *conditional* generative network (CGAN, [25]), where we learn to sample dense Cartesian coordinate maps conditioned on RGB images. By framing the scene coordinate regression task of [6] as an Image-to-Image translation problem, we can borrow the model of [9]. In particular, our conditional GAN is trained on the following objective function:

$$\begin{aligned}\mathbf{L}(G,D) = \quad & \mathbb{E}_{x,y\sim p(x,y)}\left[\log D(x,y)\right] \\ & +\mathbb{E}_{x\sim p(x),z\sim p_z}\left[\log(1-D(x,G(x,z))\right]\end{aligned}$$

Skip connections are one important feature of the chosen generator model $G(x,z)$, as they enable sharing small-scale image structure between the input and output images, thus improving the sharpness of the prediction.

IV. CAMERA POSE ESTIMATION

The camera pose estimation step largely follows [6], and sets up an energy minimization problem:

$$H^{\star} = \text{argmin}_H E(H)$$

over the camera pose matrix H. For each camera pose hypothesis H, the energy function E counts the number of outlier pixels, found by comparing the predicted scene coordinates with the backprojected location obtained from the depth image.

The energy function is optimized using preemptive RANSAC ([26]) and proceeds by repeatedly discarding the half highest-energy (i.e. most unlikely) hypotheses. The initial pose hypothesis are generated by feeding randomly-sampled sets of three coordinate correspondences to the Kabsch algorithm ([28]), which uses Singular Value Decomposition (SVD) to solve for the camera pose. Once RANSAC converges on the optimal hypothesis, we run the Kabsch algorithm on the final set of inliers to improve the accuracy of the final estimate.

Compared to [6], our method pre-computes scene coordinates for all pixels, instead of limiting the regression to the subset of pixels needed by RANSAC at each step. We believe that this approach is a natural fit for the parallel nature of GPUs and convolutional networks, and also enables the GAN to consider large-scale consistency throughout the whole output image. In particular, we want to point out the difference with respect to the CNN employed in [2], which is a patch-based model with inherently limited field of view.

V. EXPERIMENTS

A. Datasets

We run our experiments on the TUM RGBD dataset described in [18], with paired 640x480 RGB and depth images captured by a Kinect sensor. The dataset also includes the ground truth camera trajectory measured using an external motion capture system. The ground truth trajectory was only used for training and validation, and ignored at test-time.

B. Hardware and Software

The GAN is based on the open source implementation available at [16] and uses the TensorFlow deep learning library ([17]). We used a Nvidia GTX 1080 card with 8GB VRAM for both training and inference.

The regression forest and pose estimation RANSAC are based on [15], the same open-source C++ implementation that [1] used for their experiments. We used the same pose estimation code for both the GAN and the random forests.

C. Training

To train the GAN, we match each RGB image with its corresponding label, i.e. a 3-channel image with the X, Y, Z scene coordinates of each pixel. Throughout the rest of the paper, we refer to these labels as *XYZ* images, and visualize them with normalized RGB colors (Red: X, Green: Y, Blue: Z), like in figure 1.

We use the known pinhole camera geometry parameters to transform the coordinates from the camera frame to the world frame according to the external ground truth pose measurement:

$$\mathbf{m} = H\mathbf{x}$$

where $\mathbf{m}$ is the scene coordinate in the world frame (regression objective), H is the ground truth camera pose, and $\mathbf{x}$ is the 3D coordinate of the scene point in the camera frame of reference.

Figure 1 shows examples of the GAN output during the training process, at respectively 10, 30, and 70 epochs on the `fr1-desk` scene. It is clear how the accuracy and clarity of the XYZ images plateaus quickly with a limited training set (around 500 RGBD images).

While visually unpleasant, the artifacts do not have a significant impact on the localization accuracy thanks to the RANSAC step. We also observe that the GAN has partially learned to predict scene coordinates in images regions lacking depth information (for example, highly light-absorbent surfaces like the LCD monitor in the `desk` scene).

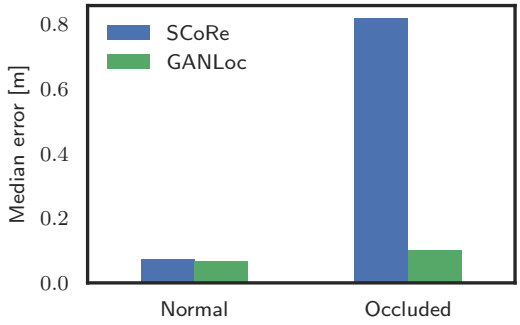

Fig. 2. Median localization error for the desk dataset

VI. RESULTS

A. Localization accuracy

We train GANLoc on 110 epochs of the `fr1-desk` scene of the TUM RGBD dataset, and compare its results to the SCoRe forest (trained using the default settings of 5 trees, 500 frames per tree, and 5000 pixels per frame).

Even with identical RANSAC pipelines, our method is more robust than SCoRe, and is able to correctly localize more frames. To quantify this improvement, we compute the same performance metrics as [6] and [7]: the camera pose is considered correct if it is within 10 cm translation and 10° rotation from the ground truth. Table I presents the quantitave results.

TABLE I
ACCURACY AND ROBUSTNESS METRICS

Scene		SCoRe	GANLoc
desk	Transl. error	7.2 cm	**6.6 cm**
	% correct	51.6%	**83.9%**
desk occluded	Transl. error	81cm	**10cm**
	% correct	40.2%	**48.9%**

While the median localization error is roughly equal to SCoRe forests (6.6cm vs 7.2cm), GANLoc is able to correctly localize many more frames (roughly 80% vs 50%). This suggests that SCoRe forests have higher best-case accuracy, but are less robust.

B. Robustness against camera occlusions

Since GANs have been successfully used for image inpainting ([14]), we expect our model to perform well even in the presence of major occlusions in the captured image.

To test this hypothesis, we artificially black out four regions of the captured image. Figure 3 shows an example of the results of this procedure. Notably, the occlusions have little effect on the predicted XYZ image, as the GAN has partially learned how to enforce its large-scale consistency to an extent that is largely impossible for regression networks with an autoencoder-like architecture.

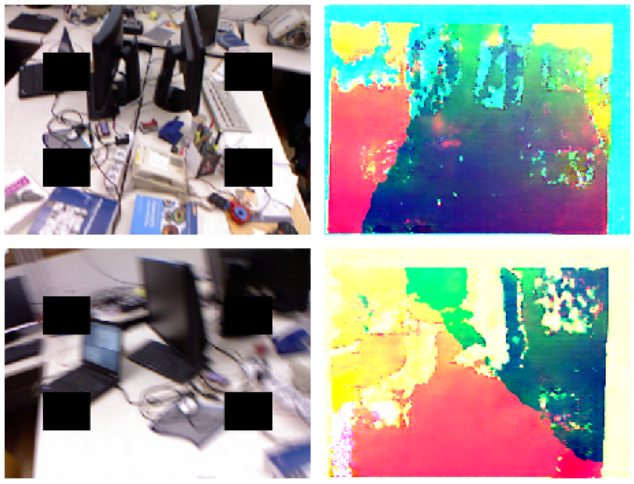

Fig. 3. The scene coordinate predictor is accurate even in the presence of major occlusions in the image, artificially introduced with black rectangles.

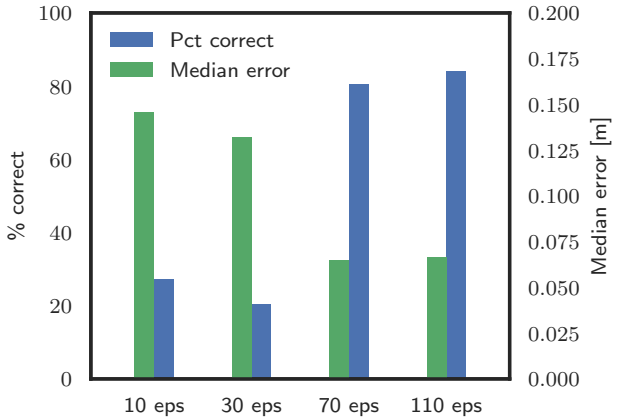

Fig. 4. Median localization error and robustness for different training regimes (10, 30, 70, and 110 epochs).

As expected, this result translates to better localization performance. The drop in localization robustness and accuracy is much less severe for GANLoc than for SCoRe forests, as can be observed from table I and figure 2.

C. *Effect of training regime*

We also conducted a preliminary investigation on the effect of the training regime on the accuracy and robustness of the localizer. In all cases, we used the Adam optimizer with an initial learning rate of 3×10^{-4}.

Figure 4 shows the evolution of the above-mentioned accuracy metrics throughout the training progression. Consistently with the qualitative results in figure 1, the key metrics plateau after roughly 100 training epochs on the fr1-desk dataset alone.

VII. CONCLUSIONS AND FUTURE WORK

Our experiments show that Conditional GANs can effectively learn to predict pixel-wise scene coordinates from a single RGB image. In combination with a robust pose estimation scheme, our system has state of the art accuracy and performs especially well in the case of major image occlusions.

Our plans for future development include testing with a larger-scale dataset and different GAN architectures, to fully understand the relationship between overfitting, transfer learning, and the representational capability of the network.

We are aware of the ongoing work in modeling uncertainty in Neural Networks (for example, the Bayesian viewpoint presented in[4]), and would like to explore consolidating the RANSAC pose estimation step within a single end-to-end trained network (possibly using the differentiable RANSAC layers introduced in [27]).

REFERENCES

[1] Clark, R., Wang, S., Markham, A., Trigoni, N., & Wen, H. (2017). VidLoc: 6-DoF Video-Clip Relocalization. Retrieved from http://arxiv.org/abs/1702.06521

[2] Massiceti, D., Krull, A., Brachmann, E., Rother, C., & Torr, P. H. S. (2016). Random Forests versus Neural Networks - Whats Best for Camera Relocalization? arXiv Preprint. Retrieved from http://arxiv.org/abs/1609.05797

[3] Kendall, A., Grimes, M., & Cipolla, R. (2015). PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization, 29382946. https://doi.org/10.1109/ICCV.2015.336

[4] Kendall, A., & Cipolla, R. (2015). Modelling Uncertainty in Deep Learning for Camera Relocalization. Retrieved from http://arxiv.org/abs/1509.05909

[5] Kendall, A., & Cipolla, R. (2017). Geometric Loss Functions for Camera Pose Regression with Deep Learning. IEEE Conference on Computer Vision and Pattern Recognition. https://doi.org/10.1109/CVPR.2017.694

[6] Shotton, J., Glocker, B., Zach, C., Izadi, S., Criminisi, A., & Fitzgibbon, A. (2013). Scene coordinate regression forests for camera relocalization in RGB-D images. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 29302937. https://doi.org/10.1109/CVPR.2013.377

[7] Valentin, J., Niener, M., Shotton, J., Fitzgibbon, A., Izadi, S., & Torr, P. (2015). Exploiting uncertainty in regression forests for accurate camera relocalization. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 7-12-NaN-2015, 44004408. https://doi.org/10.1109/CVPR.2015.7299069

[8] Cavallari, T., Golodetz, S., Lord, N. A., Valentin, J., Di Stefano, L., & Torr, P. H. S. (2017). On-the-Fly Adaptation of Regression Forests for Online Camera Relocalisation. arXiv Preprint. https://doi.org/10.1109/CVPR.2017.31

[9] Isola, P., Zhu, J.-Y., Zhou, T., & Efros, A. A. (2016). Image-to-Image Translation with Conditional Adversarial Networks. arXiv, 16. https://doi.org/arXiv:1611.07004

[10] Goodfellow, I. (2016). NIPS 2016 Tutorial: Generative Adversarial Networks. Nips, 57. https://doi.org/10.1001/jamainternmed.2016.8245

[11] Dorian, G. (n.d.). Bags of Binary Words for Fast Place Recognition in Image Sequences, 19.

[12] Irschara, A., Zach, C., Frahm, J.-M., & Bischof, H. (n.d.). From Structure-from-Motion Point Clouds to Fast Location Recognition.

[13] Sattler, T., Leibe, B., & Kobbelt, L. (2011). Fast image-based localization using direct 2D-to-3D matching. Proceedings of the IEEE International Conference on Computer Vision, 667674. https://doi.org/10.1109/ICCV.2011.6126302

[14] Pathak, D., Donahue, J., & Efros, A. A. (2016). Context Encoders : Feature Learning by Inpainting. Cvpr 2016, 25362544. https://doi.org/10.1109/CVPR.2016.278

[15] `github.com/ISUE/relocforests`

[16] `github.com/affinelayer/pix2pix-tensorflow`

[17] Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Brain, G. (2016). TensorFlow: A System for Large-Scale Machine Learning TensorFlow: A system for large-scale machine learning. 12th USENIX Symposium on Operating

Systems Design and Implementation (OSDI 16), 265284. Retrieved from https://www.usenix.org/conference/osdi16/technical-sessions/presentation/abadi

[18] Sturm, J., Engelhard, N., Endres, F., Burgard, W., & Cremers, D. (2012). A benchmark for the evaluation of RGB-D SLAM systems. IEEE International Conference on Intelligent Robots and Systems, 573580. https://doi.org/10.1109/IROS.2012.6385773

[19] Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 0712June, 34313440. https://doi.org/10.1109/CVPR.2015.7298965

[20] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Bengio, Y. (2014). Generative Adversarial Nets. Advances in Neural Information Processing Systems 27, 26722680. https://doi.org/10.1017/CBO9781139058452

[21] Berthelot, D., Schumm, T., & Metz, L. (2017). BEGAN: Boundary Equilibrium Generative Adversarial Networks. arXiv, 19. https://doi.org/1703.10717

[22] Ledig, C., Theis, L., Huszar, F., Caballero, J., Aitken, A., Tejani, A., Shi, W. (2016). Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. arXiv, 114. https://doi.org/10.1109/CVPR.2017.19

[23] Zhu, J.-Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. arXiv. Retrieved from http://arxiv.org/abs/1703.10593

[24] Radford, A., Metz, L., & Chintala, S. (2015). Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks, 115. https://doi.org/10.1051/0004-6361/201527329

[25] Mirza, M., & Osindero, S. (2014). Conditional Generative Adversarial Nets. CoRR, 17. Retrieved from http://arxiv.org/abs/1411.1784

[26] Nist, D. (2003). Preemptive RANSAC for Live Structure and Motion Estimation Corresponding Author : Preemptive RANSAC for Live Structure and Motion Estimation. Machine Vision and Applications, 16(Iccv), 129.

[27] Brachmann, E., Krull, A., Nowozin, S., Shotton, J., Michel, F., Gumhold, S., & Rother, C. (2016). DSAC - Differentiable RANSAC for Camera Localization. arXiv. https://doi.org/10.1109/CVPR.2017.267

[28] W. Kabsch. A solution for the best rotation to relate two sets of vectors. Acta Crystallographica, 1976. 4.