ViewFormer

NeRF-Free Neural Rendering from Few Images Using Transformers

Kulhánek, Jonáš; Derner, Erik; Sattler, Torsten; Babuška, Robert

DOI
10.1007/978-3-031-19784-0_12

Publication date
2022

Document Version
Final published version

Published in
Proceedings Computer Vision – ECCV 2022

Citation (APA)
Kulhánek, J., Derner, E., Sattler, T., & Babuška, R. (2022). ViewFormer: NeRF-Free Neural Rendering from Few Images Using Transformers. In S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, & T. Hassner (Eds.), *Proceedings Computer Vision – ECCV 2022 : 17th European Conference, 2022* (pp. 198-216). (Lecture Notes in Computer Science; Vol. 13675 LNCS). Springer. https://doi.org/10.1007/978-3-031-19784-0_12

ViewFormer: NeRF-Free Neural Rendering from Few Images Using Transformers

Jonáš Kulhánek[1,2(✉)], Erik Derner[1], Torsten Sattler[1], and Robert Babuška[1,3]

[1] Czech Institute of Informatics, Robotics and Cybernetics, Czech Technical University in Prague, Prague, Czech Republic
jonas.kulhanek@cvut.cz

[2] Faculty of Electrical Engineering, Czech Technical University in Prague, Prague, Czech Republic

[3] Cognitive Robotics, Faculty of 3mE, Delft University of Technology, Delft, The Netherlands

Abstract. Novel view synthesis is a long-standing problem. In this work, we consider a variant of the problem where we are given only a few context views sparsely covering a scene or an object. The goal is to predict novel viewpoints in the scene, which requires learning priors. The current state of the art is based on Neural Radiance Field (NeRF), and while achieving impressive results, the methods suffer from long training times as they require evaluating millions of 3D point samples via a neural network for each image. We propose a 2D-only method that maps multiple context views and a query pose to a new image in a single pass of a neural network. Our model uses a two-stage architecture consisting of a codebook and a transformer model. The codebook is used to embed individual images into a smaller latent space, and the transformer solves the view synthesis task in this more compact space. To train our model efficiently, we introduce a novel *branching attention* mechanism that allows us to use the same model not only for neural rendering but also for camera pose estimation. Experimental results on real-world scenes show that our approach is competitive compared to NeRF-based methods while not reasoning explicitly in 3D, and it is faster to train.

Keywords: Novel view synthesis · Neural rendering · Localization

1 Introduction

Image-based novel view synthesis, *i.e.*, rendering a 3D scene from a novel viewpoint given a set of context views (images and camera poses), is a long-standing

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-19784-0_12.

S. Avidan et al. (Eds.): ECCV 2022, LNCS 13675, pp. 198–216, 2022.
https://doi.org/10.1007/978-3-031-19784-0_12

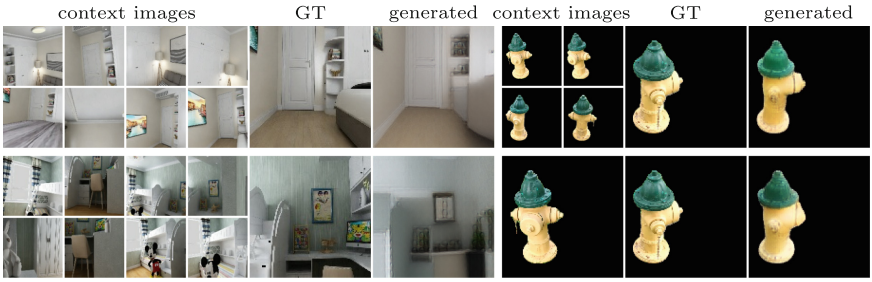


Fig. 1. Our novel view synthesis method renders images of previously unseen objects based on a few context images. It operates in 2D space without any explicit 3D reasoning (as opposed to NeRF-based approaches [51,72]). The results are shown on the CO3D [51] (**right**) and InteriorNet [32] (**left**) datasets rendered for unseen scenes

problem in computer graphics with applications ranging from robotics (*e.g.* planning to grasp objects) to augmented and virtual reality (*e.g.* interactive virtual meetings). Recently, the field has gained a lot of popularity thanks to Neural Radiance Field (NeRF) methods [2,40] that were successfully applied to the problem and outperformed prior approaches. We distinguish between two variants of the view synthesis problem. The first variant renders a novel view from multiple context images taken from similar viewpoints [40,69]. Only a (very) sparse set of context images is provided in the second variant [51,72], *i.e.*, larger viewpoint variations and missing observations need to be handled. The latter task is much more difficult as it is necessary to learn suitable priors that can be used to predict unseen scene parts. This paper focuses on the second variant.

Recently, generalizable NeRF-based approaches have been proposed to tackle this problem by learning priors for a class of objects and scenes [51,72]. Instead of learning a radiance field for each scene, they use context views captured from the target scene to construct the radiance field on the fly by projecting the image features from all context views into 3D. Highly optimized NeRF approaches [22,43,50,71] can be sped up by tuning or caching the radiance field representation [43], although often requiring lots of images per scene. To the best of our knowledge, these techniques do not apply to generalizable NeRF-based methods that do not learn a scene-specific radiance field, and take thousands of GPU-hours to train [51]. In contrast, 2D-only feed-forward networks can be highly efficient. However, explicitly encoding 3D geometric principles in them can be challenging. In our work, we thus pose the question: *Is reasoning in 3D necessary for high-quality novel view synthesis, or can a purely image-based method achieve a competitive performance?*

Recently, Rombach *et al.* [54] successfully tackled single-view novel view synthesis, where the model was able to predict novel views without explicit 3D reasoning. Inspired by these findings, we tackle the more complex problem of multi-view novel view synthesis. To answer the question, we propose a method with no explicit 3D reasoning able to predict novel views using multiple context images in a forward pass of a neural network. We train our model on a large collection of diverse scenes to enable the model to learn 3D priors implicitly. Our

approach is able to render a view in a novel scene, unseen at training time, three orders of magnitude faster than state-of-the-art (SoTA) NeRF-based approaches [51], while also being ten times faster to train. Furthermore, we are able to train a single model to render multiple classes of scenes (see Fig. 1), whereas the SoTA NeRF-based approaches typically train per-class models [51].

Our model uses a two-stage architecture consisting of a Vector Quantized-Variational Autoencoder (VQ-VAE) codebook [45] and a transformer model. The codebook model is used to embed individual images into a smaller latent space. The transformer solves the novel view synthesis task in this latent space before the image is recovered via a decoder. This enables the codebook to focus on finer details in images while the transformer operates on shorter input sequences, reducing the quadratic memory complexity of its attention layer.

For training, we pass a sequence of views into the transformer and optimize it for all context sizes at the same time, effectively utilizing all images in the training batch, which is different from other methods [20,21,46,48] that train only one query view. Unlike autoregressive models [21,46,48], we do not decode images token-by-token but all tokens are decoded at once which is both faster and mathematically exact (while autoregressive models rely on greedy strategies). Our approach can be considered a combination of autoregressive [47,68] and masked [18] transformer models. With the standard attention mechanism, the complexity would be quadratic in the number of views, because we would have to stack different query views corresponding to different context sizes along the batch dimension. Therefore, we propose a novel attention mechanism called *branching attention* with constant overhead regardless of how many query views we optimize. Our attention mechanism also allows us to optimize the same model for the camera pose estimation task – predicting the query image's camera pose given a set of context views. Since this task can be considered an "inverse" of the novel view synthesis task [70], we consider the ability to perform both tasks via the same model to be an intriguing property. Even though the localization results are not yet competitive with state-of-the-art localization pipelines, we achieve a similar level of pose accuracy as comparable methods such as [1,60].

In summary, this paper makes the following contributions: **1**) We propose an efficient novel view synthesis approach that does not use explicit 3D reasoning. Our two-stage method consisting of a codebook model and a transformer is competitive with state-of-the-art NeRF-based approaches while being more efficient to train. Compared to similar methods that do not use explicit 3D reasoning [15,20,66], our approach is not only evaluated on synthetic data but performs well on real-world scenes. **2**) Our transformer model is a combination of an autoregressive and a masked transformer. We propose a novel attention mechanism called *branching attention* that allows us to optimize for multiple context sizes at once with a constant memory overhead. **3**) Thanks to the branching attention, our model can both render a novel view from a given pose and predict the pose for a given image. **4**) Our source code and pre-trained models are publicly available at https://github.com/jkulhanek/viewformer.

2 Related Work

Novel view synthesis has a long history [12,63]. Recently, deep learning techniques have been applied with great success, enabling higher realism [16,24,38,52,53]. Some approaches use explicit reconstructed geometry to warp context images into the target view [16,24,52,53,65]. In our approach, we do not require any proxy geometry and only operate on 2D images.

Neural Radiance Field methods [2,27,36,38,40,50,71] use neural networks to represent the continuous volumetric scene function. To render a view, for each pixel in the image plane, they project a ray into 3D space and query the radiance field in 3D points along each ray. The radiance field is trained for each scene separately. Some methods generalize to new scenes by conditioning the continuous volumetric function on the context images [55,64], which allows them to utilize trained priors and render views from scenes on which the model was not trained, much like our approach. Other approaches remove the trainable continuous volumetric scene function altogether. Instead, they reproject the context image's features into the 3D space and apply the NeRF-based rendering pipeline on top of this representation [25,51,67,69,72]. Similarly to these methods, our approach also utilizes few context views (less than 20), and it also generalizes to unseen objects. However, we do not use the continuous volumetric function nor the reprojection into the 3D space. A different approach, IBRNet [69], learns to copy existing colours from context views, effectively interpolating the context views. Unlike ours, it thus cannot be applied to the settings where the object is not covered enough by the context views [25,51,67,72].

A different line of work directly maps 2D context images to the 2D query image using an end-to-end neural network [15,20,66]. GQN-based methods [15,20,66] apply a CNN to context images and camera poses and combine the resulting features. While some GQN methods [15,20] do not use any explicit 3D reasoning (same as our approach), Tobin *et al.* [66] uses an epipolar attention to aggregate the features from the context views. We optimize our model on all context images and fully utilize the training sequences, whereas GQN methods optimize only a single query view.

A recent work by Rombach *et al.* [54] proposed an approach for novel view synthesis without explicit 3D modeling. They used a codebook and a transformer model to map a single context view to a novel view from a different pose. Their approach is limited in its scope to mostly forward-facing scenes where it is easier to render the novel view given a single context view and the poses have to be close to one another. It cannot be extended to more views due to the limit on the sequence size of the transformer model. In contrast, in our approach, we focus on using multiple context views, which we tackle through the proposed branching attention. Furthermore, we can jointly train the same model for both the novel view synthesis and camera pose estimation and our decoding is faster because we decode the output at once instead of autoregressive decoding.

Visual Localization. There is an enormous body of work tackling the problem of localization, where the goal is to output the camera pose given the camera

image. *Structure-based* approaches use correspondences between 2D pixel positions and 3D scene coordinates for camera pose estimation [6,11,34,37,56,58,62]. Our method does not explicitly reason in 3D space, and the camera pose is instead predicted by the network. Simple *image retrieval* (IR) approaches store a database of all images with camera poses and for each query image they try to find the most similar images [9,10,17,26,59,74] and use them to estimate the pose of the query. IR methods can also be used to select relevant images for accurate pose estimation [4,26,56,74,75].

Pose regression methods train a convolutional neural network (CNN) to regress the camera pose of an input image. There are two categories: *absolute pose regression* (APR) methods [5,8,14,28,30,33,41,60] and *relative pose regression* (RPR) methods [1,19,31,33,39]. It was shown [59] that APR is often not (much) more accurate than IR. RPR methods do not train a CNN per scene or a set of scenes, but instead, condition the CNN on a set of context views. While our approach performs relative pose regression, the main focus of our method is on the novel view synthesis. Some pose regression methods use novel view synthesis methods [14,41,42,44], however, they assume there is a method that generates images, whereas our method performs both the novel view synthesis and camera pose regression in a single model. *Iterative refinement* pose regression methods [57,70] start with an initial camera pose estimate and refine it by an iterative process, however, our approach generates novel views and the camera pose estimates in a single forward pass.

3 Method

In this work, we tackle the problem of image-based novel view synthesis – given a set of *context* views, the algorithm has to generate the image it would most likely observe from a *query* camera pose. We focus on the case where the number of context views is small, and the views sparsely cover the 3D scene. Thus, the algorithm must hallucinate parts of the scene in a manner consistent with the context views. Therefore, it is necessary to learn a prior over a class of scenes (*e.g.*, all indoor environments) and use this prior for novel scenes. Besides rendering novel views, our model can also perform camera pose estimation, *i.e.*, the "inverse" of the view synthesis task: given a set of context views and a query image, the model outputs the camera pose from which the image was taken.

Our framework consists of two components: a codebook model and a transformer model. The codebook is used to map images to a smaller discrete latent space (*code space*), and back to the image space. In the *code space*, each image is represented by a sequence of *tokens*. For the novel view synthesis task, the transformer is given a set of context views in the code space and the query camera pose, and it generates an image in the *code space*. The codebook then maps the image tokens back to the image space. See Fig. 2 for an overview. For the camera pose estimation task, the transformer is given the set of context views and the query image in the code space, and it generates the camera pose using a regression head attached to the output of the transformer corresponding to the query image tokens.

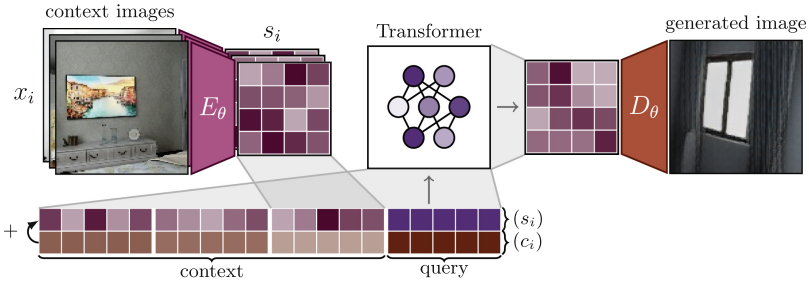


Fig. 2. Inference pipeline. The context images x_i are encoded by the codebook's encoder E_θ to the code representation s_i. We embed all tokens in s_i, and add the transformed camera pose c_i. The transformer generates the image tokens which are decoded by the codebook's decoder D_θ

Having the codebook and the transformer as separate components was inspired by the recent work on image generation [21,48,54]. The main motivation is to decrease its sequence size, because the required memory grows quadratically with it. It also allows us to separate image generation and view synthesis, enabling us to train the transformer more efficiently in a simpler space.

Codebook model is a VQ-VAE [45,49], which is a variational autoencoder with a categorical distribution over the latent space. The model consists of two parts: the encoder E_θ and decoder D_θ. The encoder first reduces the dimension of the input image from 128×128 pixels to 8×8 tokens by several strided convolution layers. The convolutional part is followed by a quantization layer, which maps the resulting feature map to a discrete space. The quantization layer stores n_{lat} embedding vectors of the same dimension as the feature vectors returned by the convolutional part of the encoder. It encodes each point of the feature map by returning the index of the closest embedding vector. The output of the encoder at position (i, j) for image x is:

$$\arg\min_k \|(f_\theta^{(enc)}(x))_{i,j} - W_k^{(emb)}\|_2, \tag{1}$$

where $W^{(emb)} \in \mathbb{R}^{n_{lat} \times d_{lat}}$ is the embedding matrix with rows $W_k^{(emb)}$ of length d_{lat} and $f_\theta^{(enc)}$ is the convolutional part of the encoder. The decoder then performs an inverse operation by first encoding the indices back to the embedding vectors by using $W^{(emb)}$ followed by several convolutional layers combined with upscaling to increase the spatial dimension back to the original image size.

Since the operation in Eq. (1) is not differentiable, we approximate the gradient with a straight-through estimator [3] and copy the gradients from the decoder input to the encoder output. The final loss for the codebook is a weighted sum of three parts: the pixel-wise mean absolute error (MAE) between the input image and the reconstructed image, the perceptual loss between the input and reconstructed image [21], and the commitment loss [45,49] $\mathcal{L}_c$, which encourages the output of the encoder to stay close to the chosen embedding vector to prevent it from fluctuating too frequently from one vector to another:

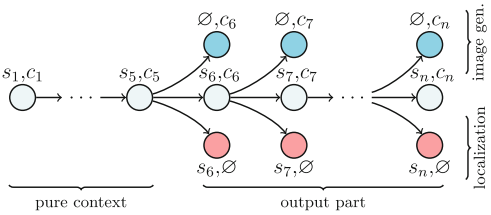


Fig. 3. Branching attention mechanism: the nodes represent parts of the processed sequence. Starting in any node and tracing the arrows backwards gives the sequence over which the attention is computed, *e.g.*, node $s_7, \varnothing$ attends to s_1, c_1, s_2, c_2, ..., $s_7, \varnothing$. Blue and red nodes in the last transformer block are used in the loss computation (Colour figure online)

$$\mathcal{L}_c = \min_k ||f_\theta^{(enc)}(x)_{i,j} - \text{sg}(W_k^{(emb)})||_2^2, \tag{2}$$

where sg is the stop-gradient operation [45]. We use the exponential moving average updates for the codebook [45]. See [45,49] for more details on the codebook, and the *supp. mat.* for the architecture details.

Transformer. We first describe the case of image generation and extend the approach to camera pose estimation later. We optimize the transformer for multiple context sizes and multiple query views in the batch at the same time. This has two benefits: it will allow the trained model to handle different context sizes, and the model will fully utilize the training batch (multiple images will be targets in the loss function). Each training batch consists of a set of n views. Let $(x_i)_{i=1}^n$ be the sequence of images under a random ordering and $(c_i)_{i=1}^n$ be the sequence of the associated camera poses. Let us also define the sequence of images transformed by the encoder E_θ parametrized by θ as $s_i = E_\theta(x_i)$, $i = 1, \ldots, n$. Note that each s_i is itself a sequence of tokens. With this formulation, we generate the next image in the sequence given all the previous views, effectively optimizing all different context sizes at once. Therefore, we model the probability $p(s_i|s_{<i}, c_{\leq i})$. Note that we do not optimize the first $n_{\min}$ views (called the *pure context*), because they usually do not provide enough information for the task.

In practice, we need to replace the tokens corresponding to each query view with mask tokens to allow the transformer to decode them in a single forward pass. For the image generation task, the tokens of the last image in the sequence are replaced with special mask tokens λ, and, for the localization task, the tokens of the last image do not include the camera pose (denoted as $\varnothing$). However, if we replaced the tokens in the training batch, the next query image would not be able to perceive the original tokens. Therefore, we have to process both the original and the masked tokens. For the i-th query image, we need the sequence of $i-1$ context views ending with masked tokens at the i-th position. We can represent the sequences as a tree (see Fig. 3) where different endings branch off the shared trunk. By following a leaf node back to the root of the tree, we recover the original sequence corresponding to the particular query view.

For localization, we train the model to output the camera pose c_i given $s_{\leq i}$ and $c_{<i}$. For image generation, this leads to $n - n_{\text{min}}$ sequences. We attach a regression head to the hidden representation of all tokens of the last image in the sequence. The query image tokens form the input, and we mask the camera poses by replacing the camera pose representation with a single trainable vector.

Branching Attention. In this section, we introduce the *branching attention* which computes attention over the tree shown in Fig. 3, and allows us to optimize the transformer model for all context sizes and tasks very efficiently. Note that we have to forward all tree nodes through all layers of the transformer. Therefore, the memory and time complexity is proportional to the number of nodes in the tree and thus to the number of views and tasks.

The input to the branching attention is a sequence of triplets of keys, values, and queries: $\left((K^{(i)}, Q^{(i)}, V^{(i)})\right)_{i=0}^{p}$ for $p = 2$, because we train the model on two tasks. Each element in the sequence corresponds to a single row in Fig. 3 and $i = 0$ is the middle row. All $K^{(i)}$, $Q^{(i)}$, $V^{(i)}$ have the size $(nk^2) \times d_m$ where d_m is the dimensionality of the model and k is the size of the image in the latent space. The output of the branching attention is a sequence $\left(R^{(i)}\right)_{i=0}^{p}$. The case of $R^{(0)}$ is handled differently because it corresponds to the trunk shared for all tasks and context sizes. Let us define a lower triangular matrix $M \in \mathbb{R}^{n \times n}$, where $m_{i,j} = 1$ if $i \leq j$. We compute the causal block attention as:

$$R^{(0)} = (\text{softmax}(Q^{(0)}(K^{(0)})^T) \odot M \otimes \mathbf{1}^{k^2 \times k^2}) V^{(0)}, \tag{3}$$

where $\otimes$ and $\odot$ are the Kronecker and element-wise product, respectively, and $\mathbf{1}^{m \times n}$ is a matrix of ones. Equation (3) is similar to normal masked attention [68] with the only difference in the causal mask. In this case, we allow the model to attend to all previous images and all other vectors from the same image. For $i > 0$ we can compute $R^{(i)}$ as follows:

$$D = Q^{(i)}(K^{(0)})^T, \tag{4}$$

$$C = \begin{bmatrix} Q^{(i)}_{1:k^2}(K^{(i)}_{1:k^2})^T \\ \vdots \\ Q^{(i)}_{(n-1)\cdot k^2+1:n\cdot k^2}(K^{(i)}_{(n-1)\cdot k^2+1:n\cdot k^2})^T \end{bmatrix}, \tag{5}$$

$$S = \text{softmax}([D, C]) \odot [(M - I) \otimes \mathbf{1}^{k^2 \times k^2}), \mathbf{1}^{nk^2 \times k^2}], \tag{6}$$

$$S' = S_{\cdot,1:n\cdot k^2}, S'' = S_{\cdot,n\cdot k^2+1:(n+1)\cdot k^2}, \tag{7}$$

$$R^{(i)} = S'V^{(0)} + \begin{bmatrix} S''_{1:k^2}V^{(i)}_{1:k^2} \\ \vdots \\ S''_{n\cdot k^2+1:(n+1)\cdot k^2}V^{(i)}_{n\cdot k^2+1:(n+1)\cdot k^2} \end{bmatrix}. \tag{8}$$

Matrix D represents the unmasked raw attention scores between i-th queries and keys from all previous images. Matrix C contains the raw pairwise attention scores between i-th queries and i-th keys (the ending of each sequence). Then, the softmax is computed to normalize the attention scores and the causal mask is

applied to the result, yielding the attention matrix S, and the respective values are weighted by the computed scores. In particular, the scores contained in the last k^2 columns of the attention matrix are redistributed back to the associated i-th values. The result $R^{(0)}$ corresponds to the nodes in the middle row in Fig. 3, whereas $R^{(i)}, i > 0$ are the other nodes.

Transformer Input and Training. To build the input for the transformer, we first embed all image tokens into trainable vector embeddings of length d_m. Before passing camera poses to the network, we express all camera poses relative to the first context camera pose in the sequence. We represent camera poses by concatenating the 3D position with the normalized orientation quaternion (a unit quaternion with a positive real part). Finally, we transform the camera poses with a trainable feed-forward neural network in order to increase the dimension to the same size as image token embeddings d_m in order to be able to sum them.

Similarly to [47], we also add the positional embeddings by summing the input sequence with a sequence of trainable vectors. However, our positional embeddings are shared for all images in the sequence, *i.e.*, the i-th token of every image will share the same positional embedding.

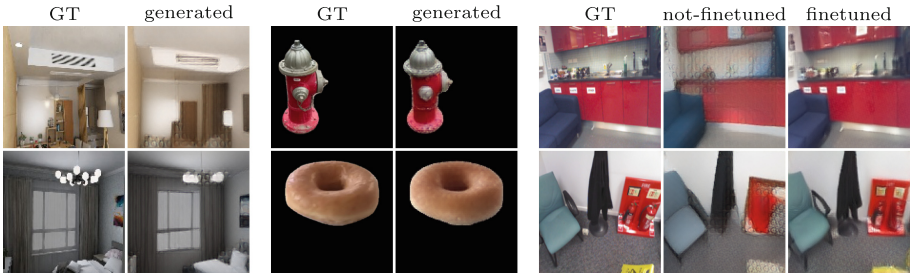
The output of the last transformer block is passed to an affine layer followed by a softmax layer, and it is trained using the cross-entropy loss to recover the last k^2 tokens $(s_{j,1}, \ldots, s_{j,k^2})$. For the localization task, the output is passed through a two-layer feed-forward neural network, and it is trained using the mean square error to match the ground-truth camera pose of the last k^2 tokens. Note that we compute the losses over position and orientation separately and add them together without weighing.[1] Since we attach the pose prediction head to the hidden representation of all tokens of the query image, we obtain multiple pose estimates. During inference, we simply average them.

4 Experiments

To answer the question of whether explicit 3D reasoning is really needed for novel view synthesis, we designed a series of experiments evaluating the proposed approach. First, we evaluate the codebook, whose performance is the upper bound on what we can achieve with the full pipeline. We next compare our method to GQN-based methods [14,20,66] that also do not use continuous volumetric scene representations. We continue by evaluating our approach on other synthetic data. Then, we compare our approach to state-of-the-art NeRF-based approaches on a real-world dataset. Finally, we show our model's localization performance.

We evaluate our approach on both real and synthetic datasets: a) **Shepard-Metzler-7-Parts (SM7)** [20,61] is a synthetic dataset, where objects composed of 7 cubes of different colours are rotated in space. b) **ShapeNet** [13] is a synthetic dataset of simple objects. We use 128×128 pixel images rendered by [64] containing two categories: cars and chairs. c) **InteriorNet** [32] is a collection of interior environments designed by 1,100 professional designers. We used the publicly available part of the dataset (20k scenes with 20 images each). While

[1] We tried dynamic weighting as described in [29], but it performed worse.



(a) InteriorNet [32] (b) CO3D [51] (c) 7-Scenes [23]

Fig. 4. Codebook evaluation on multiple datasets comparing the ground truth (**GT**) with the reconstructed image. For the 7-Scenes dataset, we compare the model fine-tuned and not-finetuned on the 7-Scenes dataset

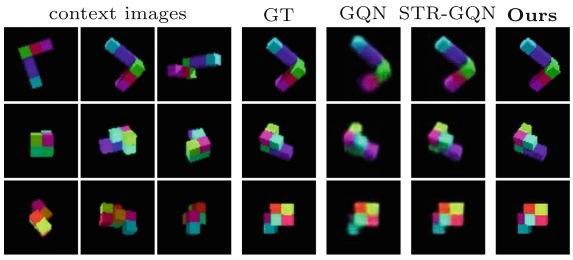


Fig. 5. Results on the SM7 dataset. We compare against GQN [20] and STR-GQN [15]

the dataset is synthetic, the renderings are similar to real-world environments. The first 600 environments serve as our test set. d) **Common Objects in 3D (CO3D)** [51] is a real-world dataset containing 1.5 million images showing almost 19k objects from 51 MS-COCO [35] categories (*e.g.*, apple, donut, vase, etc.). The capture of the dataset was crowd-sourced. e) **7-Scenes** [23] is a real-world dataset depicting 7 indoor scenes as captured by a Kinect RGB-D camera. The dataset consists of 44 sequences of 500–1,000 frames each and it is a standard benchmark for visual localization [1,8,30,31,39].

Codebook Evaluation. First, we evaluate the quality of our codebooks by measuring the quality of the images generated by the encoder-decoder architecture without the transformer. We trained codebooks of size 1,024 using the same hyperparameters for all experiments using an architecture very similar to [21]. The training took roughly 480 GPU-hours. A detailed description of the model and the hyperparameters is given in *supp. mat.* as well as in the published code.

Examples of reconstructed images are shown in Fig. 4. As can be seen, although losing some details and image sharpness, the codebooks can recover the overall shape well. The results show that using the codebook leads to good results, even though we use only 8×8 codes to represent an image. In some images, there are noticeable artifacts. In our analysis, we pinpointed the perceptual loss to be the cause, but removing the perceptual loss led to more blurry images. Further analysis of the codebooks is included in the *supp. mat.*

Fig. 6. Evaluation of our method on the InteriorNet dataset with the context size 19

Full Method Evaluation. The transformer is trained using only the tokens generated by the codebook. Having verified that our codebooks work as intended, we evaluate our complete approach in the context of image synthesis. The architecture of our transformer model is based on GPT2 [47]. We give more details on the architecture, the motivation, and the hyperparameters in the *supp. mat.*

The **SM7** dataset was used to compare our approach to other methods that only operate in 2D image space [15,20,66]. Our method achieved the best mean absolute error (MAE) of **1.61**, followed by E-GQN [66] with 2.14, STR-GQN [14] with 3.11 and the original GQN [20] method with MAE 3.13. The results were averaged over 1,000 scenes (context size was 3) and computed on images with size 64×64 pixels. A qualitative comparison is shown in Fig. 5.

We use the **InteriorNet** dataset because of its large size and realistic appearance. The models pre-trained on it are also used in other experiments. Since each scene provides 20 images, we use 19 context views. Figure 6 shows images generated by the model trained for both the localization and novel view synthesis tasks.

ShapeNet Evaluation. We used the InteriorNet pre-trained model and we fine-tuned it on the ShapeNet dataset. We trained a single model for both categories (cars and chairs) using 3 context views. The training details and additional results are given in *supp. mat.* We show the qualitative comparison with PixelNeRF [72] in Fig. 7. PixelNeRF trained a different model for each category.

The results show that our method achieves good visual quality overall, especially on the cars dataset. However, the geometry is slightly distorted on the chairs. Compared to PixelNeRF, it prefers to hallucinate a part of the scene instead of rendering a blurry image. This can cause some neighboring views to have a different colour or shape in places where the scene is less covered by context views. However, this problem can be reduced by simply adding the previously generated view to the set of context views. See the video in the *supp. mat.*

Common Objects in 3D. In order to show that we can transfer a model pre-trained on synthetic data to real-world scenes, we evaluate our method on the CO3D dataset [51]. We compare our approach with NeRF-based methods using the results reported in [51]. Unfortunately, we tried to train the PixelNeRF

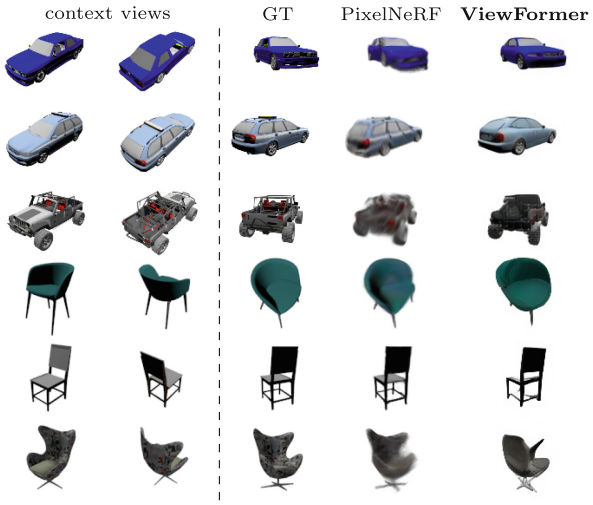


Fig. 7. ShapeNet qualitative comparison with PixelNeRF [72] using 2 context views

Table 1. Novel view synthesis results on the CO3D dataset [51] on all categories and 10 categories from [51]. We compare ViewFormer with and without localization ('no-loc') trained on all categories ('@ all cat.') and 10 selected categories ('@ 10 cat.'). We show the PSNR and LPIPS for seen and unseen scenes ('train' and 'test') and test PSNR with varying context size. The best value is **bold**; the second is underlined

			avg. test		avg. train		PSNR↑ @ # ctx. size				
EC	Method	3D	PSNR↑	LPIPS↓	PSNR↑	LPIPS↓	9	7	5	3	1
all categories	**ViewFormer @ all cat.**	✗	15.3	**0.23**	15.6	**0.22**	16.1	15.9	15.5	15.1	13.7
	ViewFormer no-loc @ all cat.	✗	15.4	**0.23**	15.8	**0.22**	16.2	16.0	15.6	15.2	13.8
	NerFormer [51]	✗	**15.7**	0.24	16.5	0.24	**16.7**	**16.4**	**16.1**	**15.5**	**13.9**
	SRN+WCE	✗	14.2	0.27	16.3	0.25	14.4	14.3	14.3	14.2	13.5
	SRN+WCE+γ	✗	13.7	0.28	**17.1**	0.25	14.0	13.8	13.9	13.7	13.2
	NeRF+WCE [25]	✗	11.6	0.27	12.6	0.27	11.9	11.8	11.8	11.6	10.8
10 categories	**ViewFormer @ 10 cat.**	✗	15.6	**0.25**	16.6	0.23	16.5	16.3	15.8	15.3	14.0
	ViewFormer no-loc @ 10 cat.	✗	15.6	**0.25**	17.1	**0.22**	16.5	16.2	15.8	15.3	14.0
	ViewFormer @ all cat.	✗	16.0	**0.25**	16.4	0.24	17.0	16.7	16.3	15.7	14.3
	ViewFormer no-loc @ all cat.	✗	16.1	**0.25**	16.6	0.23	17.0	16.8	16.3	15.8	14.3
	NerFormer [51]	✓	**17.6**	0.27	**17.9**	0.26	**18.9**	**18.6**	**18.1**	**17.1**	**15.1**
	SRN+WCE+γ	✓	14.4	0.27	17.6	0.24	14.6	14.5	14.6	14.5	13.9
	SRN+WCE	✓	14.6	0.27	16.6	0.26	14.9	14.8	14.8	14.6	13.9
	NeRF+WCE [25]	✓	13.8	0.27	14.3	0.27	12.6	14.5	14.4	14.2	13.8
	IPC+WCE	✓	13.5	0.37	14.1	0.36	13.8	13.8	13.7	13.6	12.6
	P3DMesh	✓	12.4	0.26	17.2	0.23	12.6	12.5	12.5	12.5	12.1
	NV+WCE	✓	11.6	0.35	12.3	0.34	11.7	11.6	11.6	11.6	11.3

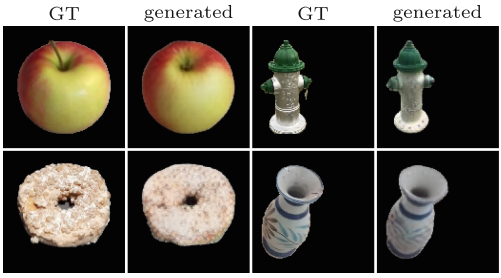


Fig. 8. Evaluation of our method on the CO3D dataset [51] with the context size 9

[72] on the CO3D dataset, but were not able to obtain good results. Therefore we omit it from the comparison. While the baselines are trained separately per category, we train two transformer models: one on the 10 categories used for evaluation in [51] and one for all dataset categories. We fine-tune the model trained on the InteriorNet dataset. The context size is 9. Additional details and hyperparameters are given in *supp. mat.*

The testing set of each category in the CO3D dataset is split into two subsets: 'train' and 'test' containing unseen images of objects seen and unseen during training respectively. We use the evaluation procedure provided by Reizenstein *et al.* [51]. It evaluates the model on 1,000 sequences from each category with context sizes 1, 3, 5, 7, 9. The PSNR) and the LPIPS distance [73] are reported. Note that the PSNR is calculated only on foreground pixels. For more details on the evaluation procedure and the details of compared methods, please see [51].

Table 1 shows results of the evaluation on all CO3D categories and on the 10 categories used for evaluation in [51]. Our method is competitive even though it does not explicitly reason in 3D as other baselines, does not utilize object masks, and even though we trained a single model for all categories while other baselines are trained per category. Note that on the whole dataset, the top-performing method, NerFormer [51], was trained for about 8400 GPU-hours while training our codebook took 480 GPU-hours, training the transformer on InteriorNet took 280 GPU-hours, and fine-tuning the transformer took 90 GPU-hours, giving a total of 850 GPU-hours. Also, note that rendering a single view takes 178 s for the NerFormer and only 93 ms for our approach.

The results show that our model has a large capacity (it is able to learn all categories while the baselines are only trained on a single category), and it benefits from more training data as can be seen when comparing models trained on 10 and all categories. We also observe that models achieve a higher performance on 10 categories than on all categories, suggesting that the categories selected by the authors of the dataset are easier to learn or of higher quality. All our models outperform all baselines in terms of LPIPS, which indicates that the images can look more realistic while possibly not matching the real images very precisely.

Figure 1 and 8 show qualitative results. Our method is able to generalize well to unseen object instances, although it tends to lose some details. To answer the original question if explicit 3D reasoning is needed for novel view synthesis, based on our results, we claim that even without explicit 3D reasoning, we can achieve similar results, especially when the data are noisy, *e.g.* a real-world dataset.

Evaluating Localization Accuracy on 7-Scenes. We compare the localization part of our approach to methods from the literature on the 7-Scenes dataset [23]. Due to space constraints, here we only summarize the results of the comparisons. Detailed results can be found in the *supp. mat.*

Our approach performs similar to existing APR and RPR techniques that also use only a single forward pass in a network [1,8,30,60], but worse than iterative approaches such as [19] or methods that use more densely spaced synthetic views as additional input [41]. Note that these approaches that do not use 3D scene geometry are less accurate than state-of-the-art methods based on 2D-3D correspondences [7,56,58]. Overall, the results show that our approach achieves a similar level of pose accuracy as comparable methods. Furthermore, our approach is able to perform both localization and novel view synthesis in a simple forward pass, while other methods can only be used for localization.

5 Conclusions and Future Work

This paper presents a two-stage approach to novel view synthesis from a few sparsely distributed context images. We train our model on classes of similar 3D scenes to be able to generalize to a novel scene with only a handful of images as opposed to NeRF and similar methods that are trained per scene. The model consists of a VQ-VAE codebook [45] and a transformer model. To efficiently train the transformer, we propose a novel branching attention module. Our approach, ViewFormer, can render a view from a previously unseen scene in 93 ms without any explicit 3D reasoning and we train a single model to render multiple categories of objects, whereas NeRF-based approaches train per-category models [51]. We show that our method is competitive with SoTA NeRF-based approaches especially on real-world data, even without any explicit 3D reasoning. This is an intriguing result because it implies that either current NeRF-based methods are not utilizing the 3D priors effectively or that a 2D-only model is able to learn it on its own without explicit 3D modeling. The experiments also show that ViewFormer outperforms other 2D-only multi-view methods.

One limitation of our approach is the large amount of data needed, which we tackle through pre-training on a large synthetic dataset. Also, we need to fine-tune both the codebook and the transformer to achieve high-quality results on new datasets, which could be resolved by utilizing a larger codebook trained on more data. Using more tokens to represent images should increase the rendering quality and pose accuracy. We also want to experiment with a simpler architecture with no codebook and larger scenes, possibly of outdoor environments.

Acknowledgement. This work was supported by the European Regional Development Fund under projects IMPACT (reg. no. CZ.02.1.01/0.0/0.0/15_003/0000468) and Robotics for Industry 4.0 (reg. no. CZ.02.1.01/0.0/0.0/15_003/0000470), the EU Horizon 2020 project RICAIP (grant agreement No 857306), the Grant Agency of the Czech Technical University in Prague (grant no. SGS22/112/OHK3/2T/13), and the Ministry of Education, Youth and Sports of the Czech Republic through the e-INFRA CZ (ID:90140).

References

1. Balntas, V., Li, S., Prisacariu, V.: RelocNet: continuous metric learning relocalisation using neural nets. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 751–767 (2018)
2. Barron, J.T., Mildenhall, B., Tancik, M., Hedman, P., Martin-Brualla, R., Srinivasan, P.P.: Mip-NeRF: a multiscale representation for anti-aliasing neural radiance fields. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 5855–5864 (2021)
3. Bengio, Y., Léonard, N., Courville, A.: Estimating or propagating gradients through stochastic neurons for conditional computation. arXiv preprint arXiv:1308.3432 (2013)
4. Bhayani, S., Sattler, T., Barath, D., Beliansky, P., Heikkilä, J., Kukelova, Z.: Calibrated and partially calibrated semi-generalized homographies. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2021)
5. Blanton, H., Greenwell, C., Workman, S., Jacobs, N.: Extending absolute pose regression to multiple scenes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pp. 38–39 (2020)
6. Brachmann, E., Rother, C.: Visual camera re-localization from RGB and RGB-D images using DSAC. IEEE Trans. Pattern Anal. Mach. Intell. **44**(9), 5847–5865 (2021)
7. Brachmann, E., Rother, C.: Visual camera re-localization from RGB and RGB-D images using DSAC. IEEE Trans. Pattern Anal. Mach. Intell. **44**, 5847–5865 (2021)
8. Brahmbhatt, S., Gu, J., Kim, K., Hays, J., Kautz, J.: Geometry-aware learning of maps for camera localization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2616–2625 (2018)
9. Camposeco, F., Cohen, A., Pollefeys, M., Sattler, T.: Hybrid camera pose estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 136–144 (2018)
10. Cao, S., Snavely, N.: Graph-based discriminative learning for location recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 700–707 (2013)
11. Cavallari, T., et al.: Real-time RGB-D camera pose estimation in novel scenes using a relocalisation cascade. IEEE Trans. Pattern Anal. Mach. Intell. **42**(10), 2465–2477 (2019)
12. Chan, S., Shum, H.Y., Ng, K.T.: Image-based rendering and synthesis. IEEE Signal Process. Mag. **24**(6), 22–33 (2007)
13. Chang, A.X., et al.: ShapeNet: an information-rich 3D model repository. arXiv preprint arXiv:1512.03012 (2015)
14. Chen, S., Wang, Z., Prisacariu, V.: Direct-PoseNet: absolute pose regression with photometric consistency. arXiv preprint arXiv:2104.04073 (2021)
15. Chen, W.C., Hu, M.C., Chen, C.S.: STR-GQN: Scene representation and rendering for unknown cameras based on spatial transformation routing. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 5966–5975 (2021)
16. Choi, I., Gallo, O., Troccoli, A., Kim, M.H., Kautz, J.: Extreme view synthesis. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 7781–7790 (2019)

17. Derner, E., Gomez, C., Hernandez, A.C., Barber, R., Babuška, R.: Change detection using weighted features for image-based localization. Robot. Auton. Syst. **135**, 103676 (2021)
18. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp. 4171–4186. Association for Computational Linguistics, Minneapolis (2019)
19. Ding, M., Wang, Z., Sun, J., Shi, J., Luo, P.: CamNet: coarse-to-fine retrieval for camera re-localization. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 2871–2880 (2019)
20. Eslami, S.A., et al.: Neural scene representation and rendering. Science **360**(6394), 1204–1210 (2018)
21. Esser, P., Rombach, R., Ommer, B.: Taming transformers for high-resolution image synthesis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12873–12883 (2021)
22. Garbin, S.J., Kowalski, M., Johnson, M., Shotton, J., Valentin, J.: FastNeRF: high-fidelity neural rendering at 200FPS. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 14346–14355 (2021)
23. Glocker, B., Izadi, S., Shotton, J., Criminisi, A.: Real-time RGB-D camera relocalization. In: 2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR), pp. 173–179. IEEE (2013)
24. Hedman, P., Philip, J., Price, T., Frahm, J.M., Drettakis, G., Brostow, G.: Deep blending for free-viewpoint image-based rendering. ACM Trans. Graph. (TOG) **37**(6), 1–15 (2018)
25. Henzler, P., et al.: Unsupervised learning of 3D object categories from videos in the wild. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4700–4709 (2021)
26. Irschara, A., Zach, C., Frahm, J.M., Bischof, H.: From structure-from-motion point clouds to fast location recognition. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 2599–2606. IEEE (2009)
27. Jain, A., Tancik, M., Abbeel, P.: Putting NeRF on a diet: Semantically consistent few-shot view synthesis. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 5885–5894 (2021)
28. Kendall, A., Cipolla, R.: Modelling uncertainty in deep learning for camera relocalization. In: 2016 IEEE International Conference on Robotics and Automation (ICRA), pp. 4762–4769. IEEE (2016)
29. Kendall, A., Cipolla, R.: Geometric loss functions for camera pose regression with deep learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5974–5983 (2017)
30. Kendall, A., Grimes, M., Cipolla, R.: PoseNet: a convolutional network for real-time 6-DOF camera relocalization. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2938–2946 (2015)
31. Laskar, Z., Melekhov, I., Kalia, S., Kannala, J.: Camera relocalization by computing pairwise relative poses using convolutional neural network. In: Proceedings of the IEEE International Conference on Computer Vision Workshops, pp. 929–938 (2017)
32. Li, W., et al.: InteriorNet: mega-scale multi-sensor photo-realistic indoor scenes dataset. In: British Machine Vision Conference (BMVC) (2018)
33. Li, X., Ling, H.: TransCamP: graph transformer for 6-DoF camera pose estimation. ArXiv abs/2105.14065 (2021)

34. Li, Y., Snavely, N., Huttenlocher, D., Fua, P.: Worldwide pose estimation using 3D point clouds. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012. LNCS, vol. 7572, pp. 15–29. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33718-5_2
35. Lin, T.-Y., et al.: Microsoft COCO: common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 740–755. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1_48
36. Liu, L., Gu, J., Zaw Lin, K., Chua, T.S., Theobalt, C.: Neural sparse voxel fields. Adv. Neural Inf. Process. Syst. **33**, 15651–15663 (2020)
37. Lynen, S., et al.: Large-scale, real-time visual-inertial localization revisited. Int. J. Robot. Res. **39**(9), 1061–1084 (2020)
38. Martin-Brualla, R., Radwan, N., Sajjadi, M.S., Barron, J.T., Dosovitskiy, A., Duckworth, D.: NeRF in the wild: neural radiance fields for unconstrained photo collections. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7210–7219 (2021)
39. Melekhov, I., Ylioinas, J., Kannala, J., Rahtu, E.: Relative camera pose estimation using convolutional neural networks. In: Blanc-Talon, J., Penne, R., Philips, W., Popescu, D., Scheunders, P. (eds.) ACIVS 2017. LNCS, vol. 10617, pp. 675–687. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-70353-4_57
40. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: NeRF: representing scenes as neural radiance fields for view synthesis. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12346, pp. 405–421. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58452-8_24
41. Moreau, A., Piasco, N., Tsishkou, D., Stanciulescu, B., de La Fortelle, A.: LENS: localization enhanced by NeRF synthesis. In: 5th Annual Conference on Robot Learning (2021)
42. Mueller, M.S., Sattler, T., Pollefeys, M., Jutzi, B.: Image-to-image translation for enhanced feature matching, image retrieval and visual localization. ISPRS Ann. Photogram. Remote Sens. Spat. Inf. Sci. **4**, 111–119 (2019)
43. Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. arXiv preprint arXiv:2201.05989 (2022)
44. Ng, T., Lopez-Rodriguez, A., Balntas, V., Mikolajczyk, K.: Reassessing the limitations of CNN methods for camera pose regression. arXiv:2108.07260 (2021)
45. van den Oord, A., Vinyals, O., kavukcuoglu, k.: Neural discrete representation learning. In: Guyon, I., et al. (eds.) Advances in Neural Information Processing Systems, vol. 30. Curran Associates, Inc. (2017)
46. Parmar, N., et al.: Image transformer. In: International Conference on Machine Learning, pp. 4055–4064. PMLR (2018)
47. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I.: Language models are unsupervised multitask learners (2019)
48. Ramesh, A., et al.: Zero-shot text-to-image generation. arXiv preprint arXiv:2102.12092 (2021)
49. Razavi, A., van den Oord, A., Vinyals, O.: Generating diverse high-fidelity images with VQ-VAE-2. In: Wallach, H., Larochelle, H., Beygelzimer, A., d' Alché-Buc, F., Fox, E., Garnett, R. (eds.) Advances in Neural Information Processing Systems, vol. 32. Curran Associates, Inc. (2019)
50. Reiser, C., Peng, S., Liao, Y., Geiger, A.: KiloNeRF: speeding up neural radiance fields with thousands of tiny MLPs. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 14335–14345 (2021)

51. Reizenstein, J., Shapovalov, R., Henzler, P., Sbordone, L., Labatut, P., Novotny, D.: Common objects in 3D: large-scale learning and evaluation of real-life 3D category reconstruction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 10901–10911 (2021)
52. Riegler, G., Koltun, V.: Free view synthesis. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12364, pp. 623–640. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58529-7_37
53. Riegler, G., Koltun, V.: Stable view synthesis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12216–12225 (2021)
54. Rombach, R., Esser, P., Ommer, B.: Geometry-free view synthesis: transformers and no 3D priors. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 14356–14366 (2021)
55. Saito, S., Huang, Z., Natsume, R., Morishima, S., Kanazawa, A., Li, H.: PIFu: pixel-aligned implicit function for high-resolution clothed human digitization. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 2304–2314 (2019)
56. Sarlin, P.E., Cadena, C., Siegwart, R., Dymczyk, M.: From coarse to fine: robust hierarchical localization at large scale. In: CVPR (2019)
57. Sarlin, P.E., et al.: Back to the feature: learning robust camera localization from pixels to pose. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3247–3257 (2021)
58. Sattler, T., Leibe, B., Kobbelt, L.: Efficient & effective prioritized matching for large-scale image-based localization. IEEE Trans. Pattern Anal. Mach. Intell. **39**(9), 1744–1756 (2016)
59. Sattler, T., Zhou, Q., Pollefeys, M., Leal-Taixe, L.: Understanding the limitations of CNN-based absolute camera pose regression. In: Proceedings of the IEEE/CVF Conference On computer Vision and Pattern Recognition, pp. 3302–3312 (2019)
60. Shavit, Y., Ferens, R., Keller, Y.: Learning multi-scene absolute pose regression with transformers. arXiv preprint arXiv:2103.11468 (2021)
61. Shepard, R.N., Metzler, J.: Mental rotation of three-dimensional objects. Science **171**(3972), 701–703 (1971)
62. Shotton, J., Glocker, B., Zach, C., Izadi, S., Criminisi, A., Fitzgibbon, A.: Scene coordinate regression forests for camera relocalization in RGB-D images. In: CVPR (2013)
63. Shum, H., Kang, S.B.: Review of image-based rendering techniques. In: Visual Communications and Image Processing 2000, vol. 4067, pp. 2–13. International Society for Optics and Photonics (2000)
64. Sitzmann, V., Zollhöfer, M., Wetzstein, G.: Scene representation networks: continuous 3D-structure-aware neural scene representations. Adv. Neural Inf. Process. Syst. **32**, 1121–1132 (2019)
65. Thies, J., Zollhöfer, M., Theobalt, C., Stamminger, M., Nießner, M.: Image-guided neural object rendering. In: 8th International Conference on Learning Representations. OpenReview. net (2020)
66. Tobin, J., Zaremba, W., Abbeel, P.: Geometry-aware neural rendering. Adv. Neural. Inf. Process. Syst. **32**, 11559–11569 (2019)
67. Trevithick, A., Yang, B.: GRF: learning a general radiance field for 3D representation and rendering. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 15182–15192 (2021)
68. Vaswani, A., et al.: Attention is all you need. In: Advances in neural information processing systems, pp. 5998–6008 (2017)

69. Wang, Q., et al.: IBRNET: learning multi-view image-based rendering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4690–4699 (2021)
70. Yen-Chen, L., Florence, P., Barron, J.T., Rodriguez, A., Isola, P., Lin, T.Y.: iNeRF: inverting neural radiance fields for pose estimation. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (2021)
71. Yu, A., Li, R., Tancik, M., Li, H., Ng, R., Kanazawa, A.: PlenOctrees for real-time rendering of neural radiance fields. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 5752–5761 (2021)
72. Yu, A., Ye, V., Tancik, M., Kanazawa, A.: pixelNeRF: neural radiance fields from one or few images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4578–4587 (2021)
73. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 586–595 (2018)
74. Zhang, W., Kosecka, J.: Image based localization in urban environments. In: Third International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT 2006), pp. 33–40. IEEE (2006)
75. Zhou, Q., Sattler, T., Pollefeys, M., Leal-Taixé, L.: To learn or not to learn: visual localization from essential matrices. In: 2020 IEEE International Conference on Robotics and Automation (ICRA), pp. 3319–3326 (2020)