Geometry-aware Deep Network for Single-Image Novel View Synthesis

Miaomiao Liu
CECS, ANU
Canberra, Australia
miaomiao.liu@anu.edu.au

Xuming He
ShanghaiTech University
Shanghai, China
hexm@shanghaitech.edu.cn

Mathieu Salzmann
CVLAB, EPFL
Lausanne, Switzerland
mathieu.salzmann@epfl.ch

Abstract

This paper tackles the problem of novel view synthesis from a single image. In particular, we target real-world scenes with rich geometric structure, a challenging task due to the large appearance variations of such scenes and the lack of simple 3D models to represent them. Modern, learning-based approaches mostly focus on appearance to synthesize novel views and thus tend to generate predictions that are inconsistent with the underlying scene structure.

By contrast, in this paper, we propose to exploit the 3D geometry of the scene to synthesize a novel view. Specifically, we approximate a real-world scene by a fixed number of planes, and learn to predict a set of homographies and their corresponding region masks to transform the input image into a novel view. To this end, we develop a new region-aware geometric transform network that performs these multiple tasks in a common framework. Our results on the outdoor KITTI and the indoor ScanNet datasets demonstrate the effectiveness of our network in generating high-quality synthetic views that respect the scene geometry, thus outperforming the state-of-the-art methods.

1. Introduction

Human beings can easily hallucinate what a scene would look like from a different viewpoint, or, for a dynamic scene, in the near future. Automatically performing such a *novel view synthesis*, however, remains a challenging task for computer vision systems.

Over the past two decades, the most popular approach to synthesizing new views has been to reconstruct an exact or approximate 3D scene model from multiple views [30, 17, 18, 25, 2]. By contrast, view synthesis from a single image, which can be applied to a broader range of problems, has received much less attention. To overcome the lack of depth information, early methods have proposed to leverage semantic-based priors [12] and geometric cues, such as vanishing points [13], which, while effective, tend to be less robust than their multi-view counterparts.

Inspired by the recent deep learning revolution in computer vision, several works have proposed to exploit Deep Convolutional Neural Networks (CNNs) to tackle the novel view synthesis problem [6, 24, 29]. Whether predicting image pixels directly [24], plane-sweep volumes [6], appearance flow [29], or appearance flow, visibility and the intensity of pixels that were not in the input view [20], these methods, in essence, all aim to solely leverage appearance to predict the flow of each pixel from the input view to the novel view without exploiting the flow of the other pixels. As such, as shown in Fig. 1, they tend to generate artefacts, such as distorted local structures in the synthesized images.

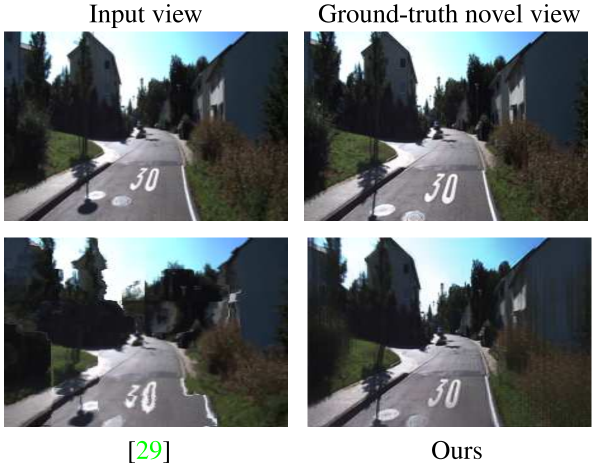

Figure 1. **Novel view synthesis from a single image.** Given an input image of the scene and a relative pose, we seek to predict a new image of the scene observed from this new viewpoint. To this end, and in contrast with state-of-the-art methods, we propose to explicitly rely on 3D geometry within a deep learning paradigm. As a consequence, and as evidenced by our results, our predictions better respect the scene structure and are thus more realistic.

In this paper, we propose to explicitly account for 3D geometry, and thus respect 3D scene structure, in the single-image novel view synthesis process. To this end, we approximate the scene by a fixed number of planes, and learn to predict corresponding homographies that, once applied to the input image, generate a set of candidate images for the novel view. We then learn to predict a selection map corresponding to each homography, which, after warping, is used to combine the candidate images to generate the novel view. In essence, our homography-based approach enforces geometric constraints on the flow field, thus modeling scene structure. Our approach can be thought of as a divide-and-conquer strategy that allows us to encode a 3D geometric prior while learning the image transformation.

To achieve this, we develop a novel deep architecture consisting of two subnetworks. The first one estimates pixel-wise depth and normals in the input image, which, in conjunction with the relative pose between the input and novel views, are then used to estimate one homography for each planar region in the scene. These homographies then let us produce a set of warped input images. The second subnetwork aims to predict a pixel-wise probability, or selection map encoding to which homography each input pixel should be associated. These maps are then warped with the corresponding predicted homographies, and the novel view is generated by combining the warped input images according to the warped selection maps. To account for pixels not in the input view and potential blur arising from the combination of multiple warped images, inspired by [20], we further propose to refine the synthesized image with an encoder-decoder network with skip connections. As evidenced by Fig. 1, our complete framework yields realistic-looking novel views.

We demonstrate the effectiveness of our approach on the challenging KITTI odometry dataset [9] and ScanNet [4], depicting complex urban outdoor scenes and indoor scenes, respectively. Thanks to our geometry-based reasoning, our method not only outperforms the state-of-the-art *appearance flow* technique of [29] quantitatively, but also yields visually more realistic predictions.

2. Related Work

Over the years, two main classes of methods have been proposed to address the novel view synthesis problem: those that rely on geometry, and the more recent ones that exploit deep learning. Below, we review the methods belonging to these two classes.

Geometry-based view synthesis. Originally, the most popular approach to view synthesis consisted of explicitly modeling 3D information, via either a detailed 3D model, or an approximate representation of the 3D scene structure. This idea was introduced in [18] more than two decades ago, by relying on multi-view stereo and a warping strategy. With the impressive progress of multi-view 3D reconstruction techniques [7], highly detailed models can be obtained, and novel views generated by making use of the target pose given as input. In complex scenes, however, this process remains challenging due to, e.g., occlusions leading to holes in the 3D models. In this context, [2] first reconstructs a partial scene from multiple images, and then synthesizes depth to fill in the missing pixels and correct the unreliable regions. Instead of relying on dense reconstruction, [30] leverages sparse points obtained from structure-from-motion in conjunction with segmented image regions, each of which is assumed to be planar and associated to a homography to warp the input image. While effective in their context, these methods are inapplicable to the scenario where a single image is available to synthesize a novel view.

Only little work has been done to leverage geometry for single-image novel view synthesis. In particular, [13] models the scene as an axis-aligned box, and requires a user to annotate the box coordinates, vanishing points and foreground to be able to render the model from a different viewpoint. In [12], the image is labeled into three geometric classes, which defines an approximate scene structure that can be rendered from a new viewpoint. These methods, however, only model a very coarse structure of the scene, and therefore cannot yield realistic novel views. By contrast, the recent work of [22] leverages a large collection of 3D models to infer the one closest to an input image. While effective for individual objects, this approach does not translate well to complex, real-world scenes with rich structures and dynamic motion, such as urban ones.

View synthesis from CNNs. With the advent of deep learning in computer vision, CNNs have recently been investigated to generate novel views. In particular, [6] proposes to synthesize the novel image from neighboring views. To this end, a plane-sweep volume, encoding a set of possible image appearances, was used as input to a network whose goal was to select the correct pixel appearance in the volume. This framework, however, requires a large memory and was only evaluated for view interpolation. Similarly, [15] tackles the view interpolation task from a pair of images, but aims to learn to rectify the two images and predict pixels correspondences. The novel view is generated by fusing the pixels of the image pair using the estimated correspondence. In contrast to these methods, we focus on single-image view synthesis.

In this context, [16] trains a variational auto-encoder to decouple the image into hidden factors, constrained to correspond to viewpoint and lighting conditions. While this network can generate an image from a new viewpoint by manipulating the hidden factors, it is mostly restricted to small rotations. In [24], an encoder-decoder network is trained to directly synthesize the pixels of the new view from the input image and the relative pose. While this network was shown to handle large rotations, the predicted images are typically blurry. Instead of directly synthesizing the image, [29] proposes to predict the displacements of the pixels from the input view to the new one, named the *appearance flow*. While this method yields sharper results, by predicting the displacements in a pixel-wise manner, it doesn't account for the scene structure, and thus, as illustrated in Fig. 1, introduces unrealistic artefacts. The recent work of [20] builds upon appearance flow by additionally predicting a visibility map, whose goal is to reflect the visibility constraints arising from a 3D object shape. During

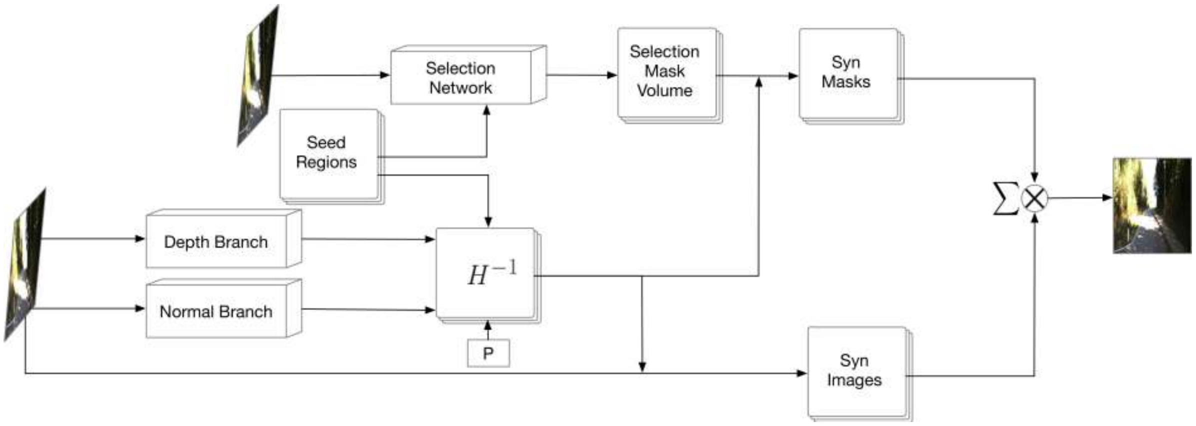

Figure 2. **Our region-aware geometric-transform network.** To tackle the single-image novel view synthesis problem, we develop a geometry-aware deep architecture consisting of two subnetworks. Given an input image, the first one predicts pixel-wise depth and normal maps. These predictions are then used in conjunction with segmentation masks obtained from the image and the desired relative pose to generate a fixed number of homographies, which are in turn employed to produce warped images. The second subnetwork predicts pixel-wise selection maps that associate each input pixel with one homography. These maps are warped by their respective homographies, and the novel view is obtained by combining the warped images according to the warped selection maps.

training, the ground-truth visibility maps are obtained by making use of 3D CAD models of the objects of interest. While this indeed exploits 3D geometry, at test time, the synthesis process neither explicitly encodes notions of geometry nor preserves local geometric structures in the new image. Furthermore, its use of 3D CAD models makes this approach better-suited to single-object view synthesis than to tackling complex real-world scenes.

By contrast, here, we explicitly leverage 3D geometry during the synthesis of the novel view, by developing a deep learning framework that exploits the notion of local homographies. As illustrated by Fig. 1, our geometry-aware deep learning strategy yields realistic predictions that better reflect the scene structure.

Note that some work has focused on the specific case of stereo view synthesis, that is, generating an image of one view from that of the other in a stereo setup [26]. While effective, this does not generalize to arbitrary novel views, since not all 3D information can be explained by disparity. Furthermore, view synthesis has been employed as supervision for depth estimation [8, 28]. However, novel views generated from predicted depth maps are typically highly incomplete, and, while suitable for depth estimation, not realistic-looking. Here, we focus on synthesizing realistic novel views with general pose variations.

3. Our Approach

Our goal is to explicitly leverage information about the 3D scene structure to perform single-image novel view synthesis. To this end, we assume that the scene can be represented with multiple planes and learn to predict their respective homographies, which let us generate a set of candidate images in the new view. We additionally learn to estimate selection maps corresponding to the homographies, which encode to which homography each input pixel should be associated. Warping these maps and using them in conjunction with the candidate new view images lets us synthesize the novel view. We then complete the regions that were unseen in the input view, and thus cannot be synthesized with this strategy, using an encoder-decoder network similar to the generator of [20]. Below, we first introduce our region-aware geometric-transform network, and then discuss this encoder-decoder refinement.

3.1. Region-aware Geometric-transform Network

To learn to predict a novel view from a single image while exploiting the 3D geometry of the scene, we develop the network shown in Fig. 2. This architecture consists of two subnetworks. The bottom one first predicts pixel-wise depth and normals from a single image in two independent streams. These predictions are then used, together with region masks extracted from the input image and the relative pose between the input view and the novel one, to compute multiple homographies, which we employ to warp the input image, thus generating candidate synthesized views. The second subnetwork, at the top of Fig. 2, predicts selection masks indicating, for each pixel, to which homography it should be associated. We then compute the novel view by assembling the candidate synthesized images according to the warped selection masks. Below, we describe these different stages in more detail.

Depth and Normal Prediction. We use standard fully-convolutional architectures to predict pixel-wise depth and normal maps separately. The details of these architectures are provided in the experiments section.

Generating Homographies. Since we represent the scene as a set of m planar surfaces, a novel view can be obtained by applying one homography to each surface. For one plane, a homography can be computed from its depth and normal, given the desired relative pose, i.e., 3D rotation and translation, and camera intrinsic parameters. To model m different planes, we make use of a segmentation of the input image into m regions, referred to as *seed regions* and described in Section 3.1.2, to pool the above-mentioned pixel-wise depth and normal estimates.

More specifically, let M be an $h \times w \times m$ binary tensor encoding m segmentation masks obtained from the $h \times w$ input image I^s. Furthermore, let us denote by M_j the binary mask corresponding to the j^{th} segment. Assuming that each segment is planar, we approximate its normal as

$$\bar{\mathbf{n}}_j = \frac{\sum_{\mathbf{x}\in\Omega} M_j(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x})}{\sum_{\mathbf{x}\in\Omega} M_j(\mathbf{x})} , \quad (1)$$

where Ω denotes the set of all pixel locations, and $\mathbf{n}(\mathbf{x})$ corresponds to the normal estimate at location $\mathbf{x}$. We then normalize $\bar{\mathbf{n}}_j$ to have unit norm.

A plane with normal $\bar{\mathbf{n}}_j$ can be defined by a vector $< \bar{n}^x_j, \bar{n}^y_j, \bar{n}^z_j, \bar{n}^d_j >$, such that any 3D point $\mathbf{Q}$ on the plane satisfies $\bar{\mathbf{n}}^T_j \mathbf{Q} + \bar{n}^d_j = 0$. While our average normal estimate provides us with the first 3 parameters, we still need to compute $\bar{n}^d_j$. To this end, let us consider the center of region j, with coordinates (c^x_j, c^y_j). We approximate the depth at the center location as

$$\bar{d}_j = \frac{\sum_{\mathbf{x}\in\Omega} M_j(\mathbf{x}) \cdot d(\mathbf{x})}{\sum_{\mathbf{x}\in\Omega} M_j(\mathbf{x})} , \quad (2)$$

where $d(\mathbf{x})$ corresponds to the depth estimate at location $\mathbf{x}$. This allows us to increase robustness to noise in the predicted depth map compared to directly using $d(c^x_j, c^y_j)$. Given the matrix of camera intrinsic parameters $\mathbf{K}$, the corresponding 3D point can be expressed as

$$\mathbf{Q} = \bar{d}_j \mathbf{K}^{-1}(c^x_j, c^y_j, 1)^T . \quad (3)$$

By making use of the plane constraint, we can estimate the last parameter $\bar{n}^d_j$ as $\bar{n}^d_j = -\bar{\mathbf{n}}^T_j \mathbf{Q}$.

Finally, let $\tilde{\mathbf{n}}_j = \bar{\mathbf{n}}_j / \bar{n}^d_j$. Given the relative rotation matrix $\mathbf{R}$ and translation vector $\mathbf{t}$ between the input and novel views, the homography for region j can be expressed as

$$\mathbf{H}_j = \mathbf{K}(\mathbf{R} - \mathbf{t}\tilde{\mathbf{n}}^T_j)\mathbf{K}^{-1}.$$

This lets us compute a homography for every seed region.

Inverse Image Warping. Each resulting homography can be applied to the pixels of the input (source) image. For each source pixel $\mathbf{x}^s$, this can be written as

$$\lambda \tilde{\mathbf{x}}^t_j = \mathbf{H}_j \tilde{\mathbf{x}}^s , \quad (4)$$

with $\tilde{\mathbf{x}}^s$ the pixel location in homogeneous coordinates, and λ the corresponding scalar. While the result of this operation will indeed correspond to a location in the target image (ignoring the fact that some will lie outside the image range), these locations will not correspond to exact, integer pixel coordinates. In our context of generating a novel view, this would significantly complicate the task of obtaining the intensity value at each target pixel, which would require combining the intensities of nearby transformed locations, whose number would vary for each target pixel.

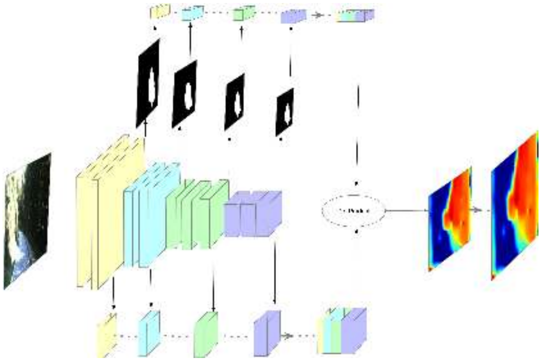

Figure 3. **Selection Network.** Instead of using hard segmentation masks to combine the candidate synthesized images, we train a network to generate a set of soft selection masks. The network structure follows that of the first 4 blocks of VGG16. We max-pool the corresponding 4 feature maps according to the seed masks and concatenate the resulting 4 feature vectors in a hypercolumn feature. We then convolve this hypercolumn feature with the concatenated complete feature maps at low resolution, which yields one global heatmap that we upsample to the original image size. Note that we normalize the pooled features and complete feature maps along the feature dimension.

To address this, instead of following a forward warping strategy (from source image to target image), we rely on an inverse warping (from target image to source image). Specifically, for every target pixel location $\mathbf{x}^t_i$, we obtain the corresponding source location by relying on the inverse homography $\mathbf{H}^{-1}_j$ as $\tilde{\mathbf{x}}^s_{i,j} \propto \mathbf{H}^{-1}_j \tilde{\mathbf{x}}^t_i$. We then compute the target intensity value at pixel $\mathbf{x}^t_i$ by bilinear interpolation as

$$\hat{I}^t_j(\mathbf{x}^t_i) = \sum_{q\in o^i_j} I^s(1 - |x^s_{i,j} - x^s_{q,j}|, 1 - |y^s_{i,j} - y^s_{q,j}|) , \quad (5)$$

where I^s is the input source image, and o^i_j denotes the 4-pixel neighborhood of $\mathbf{x}^s_{i,j}$, which itself is predicted by the inverse homography.

Selection Network. As discussed below, we generate the novel view by assembling the m candidate target images obtained as described above. To this end, we develop a *selection network* to predict m planar region masks from the input image and seed region masks (Section 3.1.2). More precisely, for each seed region, we aim to predict a soft selection map indicating the likelihood for every input pixel to be associated to the corresponding homography.

Specifically, the structure of our selection network follows that of the first 4 convolutional blocks of VGG16 [23].

As shown in Fig 3, each seed region mask is used to max-pool the corresponding 4 feature maps. We then concatenate the resulting 4 features to form a hypercolumn [11] feature, which we convolve with the concatenated complete feature maps at the lower resolution. This yields a low-resolution heat map, which we upsample to the original image size. The resulting heat map indicates a notion of similarity between the features at every pixel and the one pooled over the seed region. This procedure is performed individually for the m seed regions, but using shared network parameters. Note that the resulting m selection maps are defined in the input view, and we thus apply our inverse warping procedure to compute them in the novel view.

Novel View Prediction. Given the selection maps $\{\tilde{M}_j\}$, we first compute a normalized transformed mask for the novel view as

$$\hat{M}_j^t(\mathbf{x}_i^t) = \frac{\sum\limits_{q \in o_j^i} \tilde{M}_j(1 - |x_{i,j}^s - x_{q,j}^s|, 1 - |y_{i,j}^s - y_{q,j}^s|) + \epsilon}{\sum\limits_{k=1}^{m} \sum\limits_{q \in o_k^i} (\tilde{M}_k(1 - |x_{i,k}^s - x_{q,k}^s|)(1 - |y^s i, k - y_{q,k}^s|) + \epsilon)} \, . \tag{6}$$

Note that the resulting transformed masks are not binary, but rather provide weights to combine the m estimated target images. To account for the fact that some pixels will be warped outside the input image with all m homographies, we make use of a small constant ϵ, which prevents division by 0 in the normalization process and yields uniform weights for such pixels. In our experiments, we set $\epsilon = 0.0001$. We compute the novel view as

$$\hat{I}^t(\mathbf{x}_i^t) = \sum_{j=1}^{m} \hat{I}_j^t(\mathbf{x}_i^t) \cdot \hat{M}_j^t(\mathbf{x}_i^t) \, . \tag{7}$$

Note that some of the pixels in the output view will be mapped outside the input image by all homographies. In the simplest version of our approach, we fill in the intensity of each such pixel by using the value at the nearest pixel in the input image. In Section 3.2, we introduce a refinement network that produces more realistic predictions.

3.1.1 Learning

The novel view predicted using Eq. 7 is a function of the homographies, which themselves are functions of the normal and depth estimates, and of the selection masks, which in turn depend on the depth and normal branch parameters $\mathbf{W}_d, \mathbf{W}_n$, and selection network parameters $\mathbf{W}_s$, respectively. Altogether, the prediction can then be thought of as a function of the parameters $\mathbf{W} = \{\mathbf{W}_d, \mathbf{W}_n, \mathbf{W}_s\}$ given an input image I^s, and a relative pose $\mathbf{P}$, encompassing the 3D rotation, translation and camera intrinsics, and the segmentation seed region masks M.

All the operations described above are differentiable. The least obvious cases are the bilinear interpolations of Eqs. 5 and 6, and the use of the inverse homography. For the former ones, we refer the reader to [14], who showed that the (sub)-gradient of bilinear interpolation with respect to $\mathbf{W}$, could be efficiently computed. For the latter case, we propose to exploit the Sherman-Morrison formula [21], provided in the supplementary material, to avoid having to explicitly compute the inverse of the homography.

In our context, this formula lets us express the inverse of the homography analytically as follows. Let

$$\tilde{\mathbf{H}}^{-1} = \mathbf{R}^{\mathrm{T}} + \frac{\mathbf{R}^{\mathrm{T}}\mathbf{t}\tilde{\mathbf{n}}^{\mathrm{T}}\mathbf{R}^{\mathrm{T}}}{1 - \tilde{\mathbf{n}}^{\mathrm{T}}\mathbf{R}^{\mathrm{T}}\mathbf{t}} \, . \tag{8}$$

Then, we have $\mathbf{H}^{-1} = \mathbf{K}\tilde{\mathbf{H}}^{-1}\mathbf{K}^{-1}$. This formulation makes it easy to compute the gradient of the inverse homography w.r.t. the estimated depth and normals, and thus to train our model using backpropagation.

To this end, we make use of an ℓ_1 loss between the true target image and the estimated one. Given N training samples, learning can then be expressed as

$$\min_{\mathbf{W}} \frac{1}{N} \sum_{i=1}^{N} \|I_i^t - \hat{I}_i^t(I_i^s, P_i, M_i, \mathbf{W})\|_1 \, , \tag{9}$$

where I_i^t is the ground-truth novel view, and where, with a slight abuse of notation, we denote the segmentation mask for sample i as M_i. More details about optimization are provided in Section 4.

3.1.2 Obtaining Seed Regions

Throughout our framework, we assume to be given m segmentation masks as input, corresponding to the m planes we use to represent the scene. To extract these masks, we make use of the following simple, yet effective strategy. We first over-segment the image into superpixels using SLIC [1]. For each superpixel, we then extract its RGB value and center location as features and use *K-means* to cluster the superpixels into m regions. This strategy has the advantage over learning-based segmentation masks of generating compact regions, which are better suited to estimating the corresponding plane parameters. Furthermore, as evidenced by our experiments, it allows us to obtain accurate synthesized views that respect the scenes 3D structure.

3.2. Refinement Network

Our region-aware geometric-transformation network produces a novel view image that preserves the local geometric structures of the scene. While geometric transformations can synthesize regions that appear both in the input and novel views, it cannot handle the regions that are only present in the novel view, i.e., that were hidden in the input view. To address this, inspired by [20], we make use of the encoder-decoder refinement network depicted by Fig. 4. While the structure of this network is the same as in [20], we make use of a different, simpler loss function to train it.

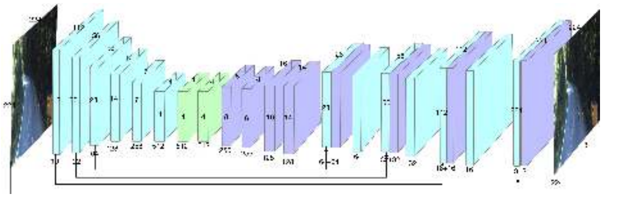

Figure 4. **Refinement Network.** Our refinement network adopts an encoder-decoder structure with skip connections. The blue blocks denote convolutions with stride two followed by batch normalization and leaky ReLU. The green blocks denote convolutions with stride one followed by batch normalization and leaky ReLU. The purple blocks denote deconvolutions followed by batch normalization and ReLU.

Specifically, let L_p denote the mean pixel ℓ_1 error. We then define the loss of our refinement network as

$$L_t = L_p + \lambda L_f \ , \tag{10}$$

where L_f is a feature ℓ_1 loss. That is, it corresponds to an ℓ_1 loss between features extracted from a fixed VGG-19 network, pre-trained for classification on ImageNet. In particular, we concatenate features from the 'conv1_2', 'conv2_2', 'conv3_2', 'conv4_2' and 'conv5_2' layers of VGG-19. This strategy has proven effective in [3] in the context of image-to-image translation. In particular, it has the advantage over [20] of not relying on a generative adversarial network, which are known to be hard to train. As shown in our results, this refinement network not only hallucinates the missing parts of the synthesized images, but it also removes the blur arising from combining multiple warped images.

4. Experiments

We evaluate our approach both quantitatively and qualitatively on the challenging urban KITTI odometry dataset [9], which depicts complex scenes with rich structure and dynamic objects, and on the large indoor scene ScanNet dataset [4], which covers diverse scene types. We compare our approach with the state-of-the-art single-image view synthesis algorithm of [29] for real-world scenes[1]. Furthermore, we also report the results of a depth-based baseline consisting of using the predictions of our depth stream warped to the new pose, followed by bicubic interpolation to obtain a complete image.

4.1. Experimental Setup

KITTI Dataset. For the comparison with [29] to be fair, we adopt the same data splits as them. Namely, we use the video sequences with index 0 to 8 as training set, and 9 to 10 as test set. We then generate our training and test pair in the following way, similar to that of [29]: For each image in a sequence, we randomly sample a frame number for the input image and for the target image such that they are separated by at most ±10 frames.

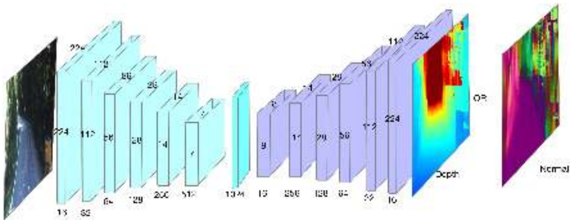

Figure 5. **Encoder-decoder network for depth or normal prediction on KITTI.** Both our depth and normal streams make use of this architecture. However, they rely on different parameters.

ScanNet Dataset. We make use of the training, validation and test splits provided with ScanNet. In particular, we use 405 training sequences to learn our model and 312 sequences from the test set for testing. We form the input-target pairs in the same manner as for KITTI. In total, we use 30000 training pairs and 5000 test pairs.

We resize the images from both datasets to $224 \times 224 \times 3$. to match that of [29]. To obtain the segmentation masks, we first oversegment each image into 400 SLIC [1] superpixels and cluster them into $m = 16$ regions, as described in Section 3.1.2. This represents a good trade-off between the accuracy of our piece-wise planar representation on the training data and the memory consumption of our method. In practice, this proved sufficient to yield realistic novel views.

4.2. Training Procedure

We train our model in a stage-wise manner: First, the depth and normal branches, then the selection network given fixed depth and normal branches, and finally the refinement network while rest of the framework is fixed. We tried to then fine-tune the entire network end-to-end, but did not observe any significant improvement.

Training the depth and normal networks. For the indoor ScanNet dataset, we were able to directly use the network of [5], which predicts both depth and normals. This network was pre-trained on NYU-v2 [19], and we simply fine-tuned it on our data. In particular, since ScanNet does not provide ground-truth normals, we fit a plane to each SLIC superpixel, and assigned the corresponding normal to all its pixels. The fine-tuned network yields a relative depth error of 0.236. We do not report the normal error, since the ground-truth normals were obtained from the depth maps.

For KITTI, we were unfortunately unable to train an equivalent model from scratch. Therefore, we relied on the simpler encoder-decoder network of Fig. 5, which is more compact and easier to train. To this end, we made use of the ℓ_1 loss for the inverse depth and of the negative inner product as a normal loss. Note that KITTI only provides sparse ground-truth depth maps. While this is sufficient to train the depth branch, it does not allow us to generate ground-truth normals as in ScanNet. To this end, we used the stereo framework of [27] to generate dense depth maps, which we

[1] Note that, as discussed in Section 2, the transformation-grounded network of [20] focuses on single-object novel view synthesis.

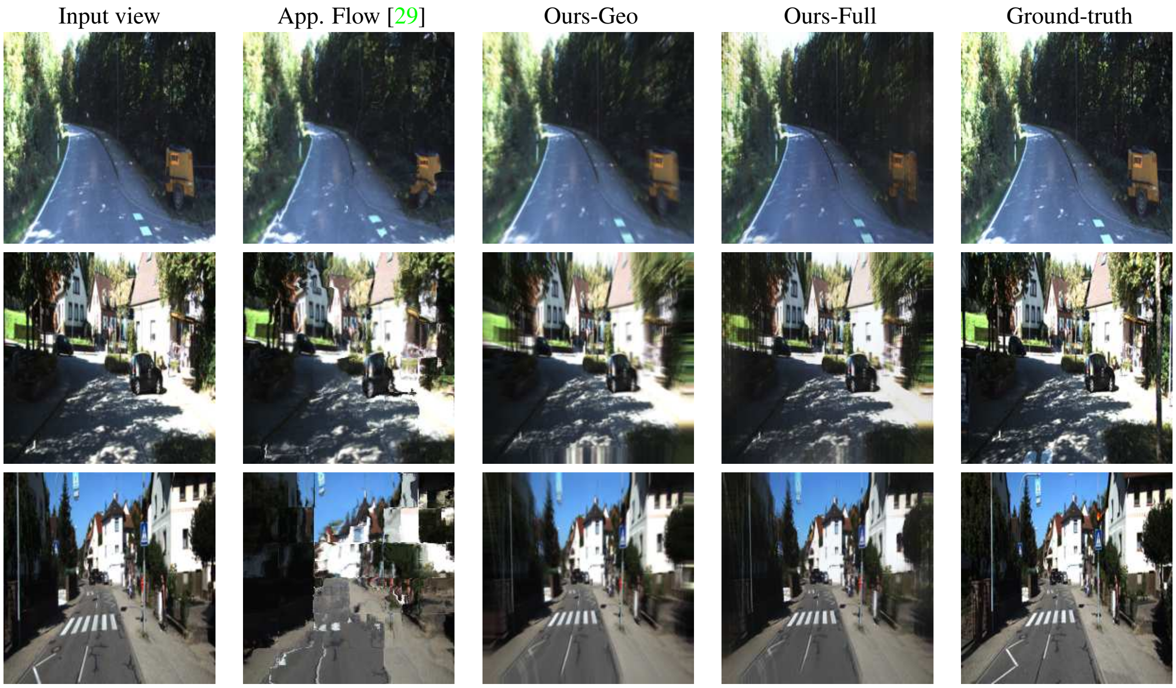

Figure 6. **Qualitative comparison of our approach with the appearance flow method of [29] on KITTI.** While appearance flow yields artifacts, our approach, which reasons about 3D geometry, yields more realistic results. This is noticeable, for instance, by looking at the bottom right part of the first image and at the buildings in the other images.

Method	ℓ_1-KITTI	ℓ_1-ScanNet
App. flow [29]	0.471	-
Depth-branch	0.668	0.217
Ours-Geo	**0.340**	**0.167**
Ours-Full	0.345	0.176

Table 1. **Quantitative evaluation on KITTI and ScanNet.** We compare our approach with the state-of-the-art method of [29] and our baseline based on our depth estimates. Our approach significantly outperforms the baselines, thus achieving state-of-the-art performance on these datasets.

used, in turn, to obtain normal maps using superpixels. The final depth network yields a relative error of 0.274.

Note that we analyze the influence of the depth and normal prediction accuracy on our final novel view synthesis results in our results section.

Training the selection network. The selection network takes the predicted depth and normals, together with the image, relative pose and seed regions, as input to synthesize the novel view. Since we do not have ground-truth labels for the selection maps, we therefore directly trained the selection network using the mean pixel ℓ_1 error as a loss.

Training the refinement network. The refinement network aims to improve an initial synthesized view. We train it using the loss of Eq. 10, with $\lambda = 0.01$.

We implemented our model in tensorflow and trained it on two NVIDIA Tesla P100, each with 16GB memory. We used mini-batches of size 10, and employed the ADAM solver with a learning rate of 0.0001,and the default values $\beta_1 = 0.9$ and $\beta_2 = 0.999$. We will make our code publicly available upon acceptance of the paper.

4.3. Results

In Table 1, we compare our approach, both without (Ours-Geo) and with (Ours-Full) refinement network, with the state-of-the-art appearance flow technique of [29] on KITTI and ScanNet, based on the mean pixel ℓ_1 error metric. Note that our approach outperforms the baseline that uses our depth estimates, without explicitly modeling the scene structure, by a large margin. This evidences the importance of accounting for 3D scene structure. Our approach also significantly outperforms the state-of-the-art appearance flow method on KITTI.[2] This again shows the benefits of modeling geometry, as done by our region-aware geometric-transform network. Interestingly, the refinement network tends to slightly degrade the novel view accuracy.

However, when looking at the qualitative comparison in Figs. 1, 6 and 7, we can see that our complete model (Ours-Full) yields more realistic novel views than both Ours-Geo and appearance flow [29]. Note that, by not leveraging structure, appearance flow yields to unrealistic artifacts. By contrast, the results of our approach that exploits 3D geometry look more natural. This, for instance, can be observed

[2] Note that, because the training code for appearance flow is not available, we had to re-implement it, and despite confirming that our implementation was correct using the KITTI dataset, we were unable to make training converge on ScanNet.

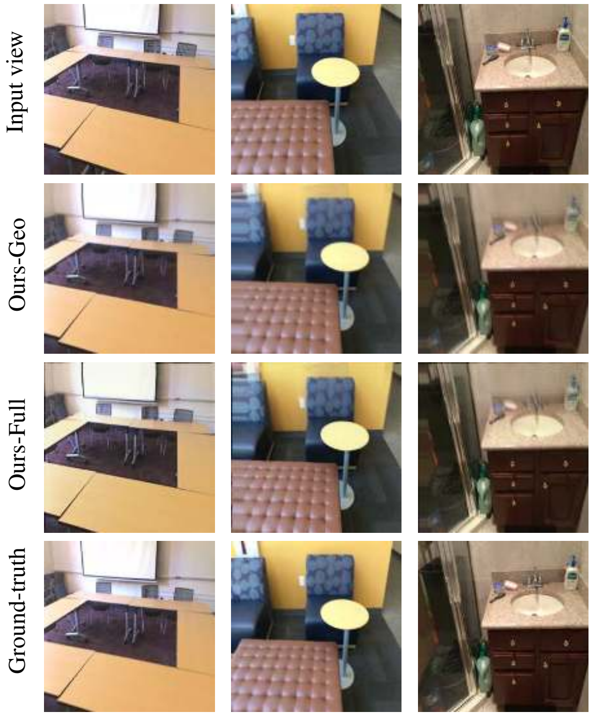

Figure 7. **Qualitative results of our approach on ScanNet.**

gtDep	gtNor	estDep	estNor	Seed	SelMap	ℓ_1
✓	✓	✗	✗	✓	✗	0.357
✓	✓	✗	✗	✗	✓	0.329
✗	✗	✓	✓	✓	✗	0.373
✗	✗	✓	✓	✗	✓	0.340

Table 2. **Influence of the quality of the depth and normal estimates and of learning the selection maps on KITTI.** From left to right: gtDep and gtNor denote the ground-truth depth and normals, respectively; estDep and estNor denote the estimated depth and normals, respectively; Seed and SelMap denote the hard-segmentations corresponding to the seed region and the selection maps obtained with our selection network, respectively.

by looking at the bottom-right corner of the first image in Fig. 6, where we better model the shape of the object, and at the buildings in the other images.

In Table 2, we analyze the influence of the quality of the depth and normal estimates, and the effect of learning the selection maps. In particular, we compare the error obtained when using the ground-truth depth and normals instead of the predicted ones, and when using the seed regions as 'hard' segmentation masks instead of the learnt selection maps. In both cases, the best results are obtained by using the ground-truth depth and normals in conjunction with our selection maps, followed by using the estimated depth and normals with our selection maps. This shows (i) the importance of learning the combination of the multiple synthesized candidates; and (ii) that the results of our approach will further improve as progress in single-image depth and normal prediction is made. A similar table for ScanNet is provided in the supplementary material.

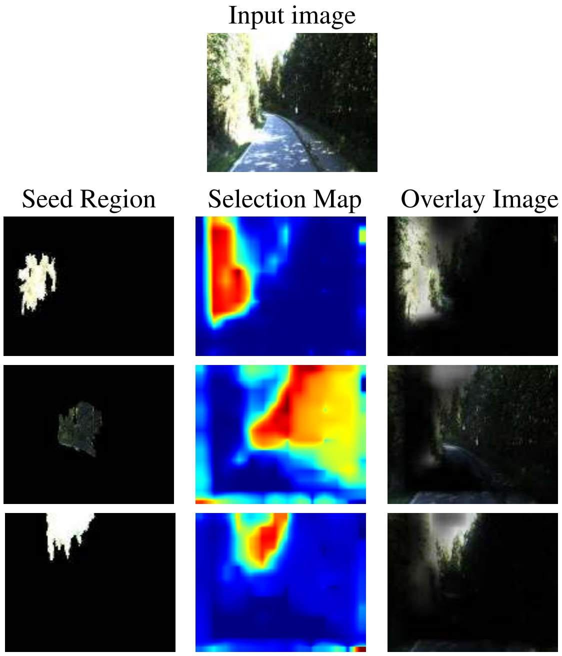

Figure 8. **Sample seed regions and predicted selection maps in the input view.** From left to right: the seed region, predicted selection map and selection map overlaid on the input image, showing that the corresponding region is close to planar. Red indicates a high likelihood for a pixel to belong to the plane defined by the seed region and blue to a low likelihood.

In Fig. 8, we illustrate what the selection network learns. To this end, we show the initial seed region overlaid with input image, and the likelihood of the pixels to be associated to this plane, predicted by the selection network. From the examples, we can see that the selection network extends the initial seed regions to larger planes of semantically and visually coherent pixels, such as a larger tree regions.

5. Conclusion

We have introduced a geometry-aware deep learning framework for novel view synthesis from a single image. Our approach models the scene with a fixed number of planes, and learns to predict homographies, which, in conjunction with a predicted selection map and a desired relative pose, let us generate the novel view. Our experiments on the challenging KIITI and ScanNet datasets have demonstrated the benefits of our approach; by leveraging 3D geometry, our method yields predictions that better match the scene structure, and thus outperforms the state-of-the-art single-image novel view synthesis techniques. Training the depth branch of our framework currently relies on ground-truth depth maps. In the future, we will investigate the use of weakly-supervised depth prediction methods [8, 10, 28] that only exploit two views to perform this task.

Acknowledgments This work was done when the first author was working in Data61, CSIRO, Australia. The Titan X used for this research was donated by the NVIDIA Corporation.

References

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*, 34(11):2274–2282, 2012. 5, 6

[2] G. Chaurasia, S. Duchene, O. Sorkine-Hornung, and G. Drettakis. Depth synthesis and local warps for plausible image-based navigation. *ACM Transactions on Graphics (TOG)*, 32(3):30, 2013. 1, 2

[3] Q. Chen and V. Koltun. Photographic image synthesis with cascaded refinement networks. In *ICCV*, 2017. 6

[4] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017. 2, 6

[5] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2650–2658, 2015. 6

[6] J. Flynn, I. Neulander, J. Philbin, and N. Snavely. Deepstereo: Learning to predict new views from the world's imagery. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5515–5524, 2016. 1, 2

[7] Y. Furukawa and J. Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE transactions on pattern analysis and machine intelligence*, 32(8):1362–1376, 2010. 2

[8] R. Garg, G. Carneiro, and I. Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *European Conference on Computer Vision*, pages 740–756. Springer, 2016. 3, 8

[9] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 2, 6

[10] C. Godard, O. Mac Aodha, and G. J. Brostow. Unsupervised monocular depth estimation with left-right consistency. *arXiv preprint arXiv:1609.03677*, 2016. 8

[11] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Hypercolumns for object segmentation and fine-grained localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 447–456, 2015. 5

[12] D. Hoiem, A. A. Efros, and M. Hebert. Automatic photo pop-up. *ACM transactions on graphics (TOG)*, 24(3):577–584, 2005. 1, 2

[13] Y. Horry, K.-I. Anjyo, and K. Arai. Tour into the picture: using a spidery mesh interface to make animation from a single image. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 225–232. ACM Press/Addison-Wesley Publishing Co., 1997. 1, 2

[14] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu. Spatial transformer networks. In *Advances in Neural Information Processing Systems*, pages 2017–2025, 2015. 5

[15] D. Ji, J. Kwon, M. McFarland, and S. Savarese. Deep view morphing. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017. 2

[16] T. D. Kulkarni, W. F. Whitney, P. Kohli, and J. Tenenbaum. Deep convolutional inverse graphics network. In *Advances in Neural Information Processing Systems*, pages 2539–2547, 2015. 2

[17] F. Liu, M. Gleicher, H. Jin, and A. Agarwala. Content-preserving warps for 3d video stabilization. *ACM Transactions on Graphics (TOG)*, 28(3):44, 2009. 1

[18] L. McMillan and G. Bishop. Plenoptic modeling: An image-based rendering system. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 39–46. ACM, 1995. 1, 2

[19] P. K. Nathan Silberman, Derek Hoiem and R. Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012. 6

[20] E. Park, J. Yang, E. Yumer, D. Ceylan, and A. C. Berg. Transformation-grounded image generation network for novel 3d view synthesis. In *CVPR*, 2017. 1, 2, 3, 5, 6

[21] W. H. Press. *Numerical recipes 3rd edition: The art of scientific computing*. Cambridge university press, 2007. 5

[22] K. Rematas, C. Nguyen, T. Ritschel, M. Fritz, and T. Tuytelaars. Novel views of objects from a single image. *arXiv preprint arXiv:1602.00328*, 2016. 2

[23] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 4

[24] M. Tatarchenko, A. Dosovitskiy, and T. Brox. Multi-view 3d models from single images with a convolutional network. In *European Conference on Computer Vision*, pages 322–337. Springer, 2016. 1, 2

[25] O. J. Woodford, I. D. Reid, P. H. Torr, and A. W. Fitzgibbon. On new view synthesis using multiview stereo. In *BMVC*, pages 1–10, 2007. 1

[26] J. Xie, R. Girshick, and A. Farhadi. Deep3d: Fully automatic 2d-to-3d video conversion with deep convolutional neural networks. In *European Conference on Computer Vision*, pages 842–857. Springer, 2016. 3

[27] J. Zbontar and Y. LeCun. Computing the stereo matching cost with a convolutional neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1592–1599, 2015. 6

[28] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe. Unsupervised learning of depth and ego-motion from video. *arXiv preprint arXiv:1704.07813*, 2017. 3, 8

[29] T. Zhou, S. Tulsiani, W. Sun, J. Malik, and A. A. Efros. View synthesis by appearance flow. In *European Conference on Computer Vision*, pages 286–301. Springer, 2016. 1, 2, 6, 7

[30] Z. Zhou, H. Jin, and Y. Ma. Plane-based content preserving warps for video stabilization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2299–2306, 2013. 1, 2