Multi-view to Novel view: Synthesizing novel views with Self-Learned Confidence

Shao-Hua Sun[1] Minyoung Huh[2] Yuan-Hong Liao[3]
Ning Zhang[4] Joseph J. Lim[1]

University of Southern California[1] Carnegie Mellon University[2]
National Tsing Hua University[3] Snap Inc.[4]

Abstract. In this paper, we address the task of multi-view novel view synthesis, where we are interested in synthesizing a target image with an arbitrary camera pose from given source images. We propose an end-to-end trainable framework that learns to exploit multiple viewpoints to synthesize a novel view without any 3D supervision. Specifically, our model consists of a flow prediction module and a pixel generation module to directly leverage information presented in source views as well as hallucinate missing pixels from statistical priors. To merge the predictions produced by the two modules given multi-view source images, we introduce a self-learned confidence aggregation mechanism. We evaluate our model on images rendered from 3D object models as well as real and synthesized scenes. We demonstrate that our model is able to achieve state-of-the-art results as well as progressively improve its predictions when more source images are available.

Keywords: Novel view synthesis, multi-view novel view synthesis

1 Introduction

With countless encounters of scenes and objects , humans learn to build a mental understanding of 3D objects and scenes just from 2D cross-sections, which in turn, allows us to imagine an unseen view with little effort. This is only possible because humans can integrate their statistical understanding of the world with the presented information. With more and more concrete prior information (e.g more viewpoints, shape understanding etc.), humans learn to consolidate all the information to predict with more confidence. This ability allows humans to make an amodal completion from just the presented data. In computer vision, these approaches are isolated and tackled separately, and the fusion of data is less well understood. Hence, we would like to develop an approach that not only learns to utilize what is given but also incorporate its 3D statistical understanding.

The task of synthesizing a novel view given an image or a set of images is known as *novel view synthesis*. The practical applications of it range from

Code is available on our website https://shaohua0116.github.io/Multiview2Novelview

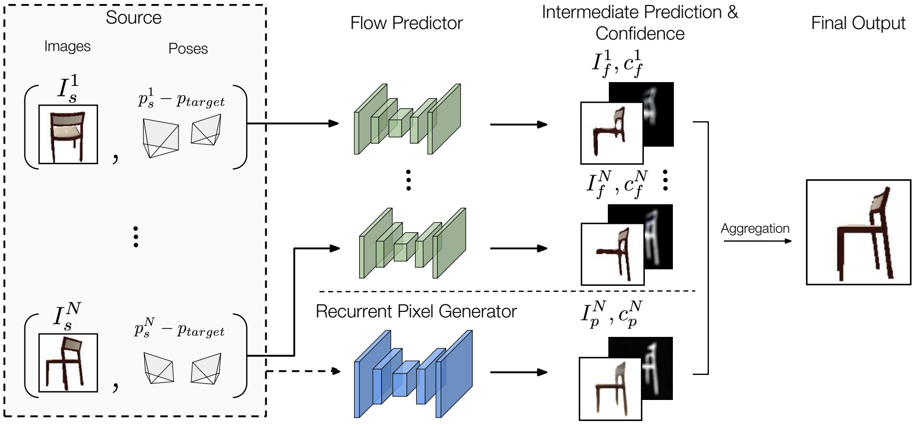

Fig. 1: Overview of our proposed network architecture. Given a set of N source images with different viewpoints and a target pose (on the left), *Flow Predictor* learns to predict a dense flow field to move the pixels presented in a source image to produce a target image for each source image. *Recurrent Pixel Generator* is trained to directly synthesize a target image given a set of source images. The two modules are trained to predict per-pixel confidence maps associated to their predictions. The final prediction is obtained by aggregating the $N + 1$ predictions with self-learned confidence maps.

but not limited to: computer vision, computer graphics, and virtual reality. Systems that perform on cross-view image inputs, such as action recognition [1–3] and 3D reconstruction [4–7], can leverage synthesized scenes to boost existing performance when the number of available views is limited. Furthermore, novel view synthesis can be used jointly on 3D Editing of 2D Photos [8–10] as well as rendering virtual reality environments using a history of frames [11, 12].

In this paper, we are interested in the task of *novel view synthesis* when multiple source images are given. Given a target camera pose and an arbitrary number of source images and their camera poses, our goal is to develop a model that can synthesize a target image and progressively improve its predictions. To address this task, a great amount of effort have been expended in geometry-based methods [13–17] aiming to directly estimate the underlying 3D structures by exploiting the knowledge of geometry. These methods, while successful with abundant source data, are unable to recover the desired target viewpoint with only a handful of images due to the inherent ambiguity of 3D structures.

With the emergence of neural networks, learning-based approaches have been applied to tackle this issue of data sparsity. A great part of this research was fueled by the introduction of a large-scale synthetic 3D model datasets such as [18]. The previous line of work that uses learning can be vaguely divided into two categories: pixel generation [19–21] and flow prediction [22, 23]. While directly regressing pixels can generate structurally consistent results, it is susceptible to generating blurry results largely in part of the inherent multi-modality of this task. Flow prediction, on the other hand, can generate realistic texture but is unable to generate regions that are not present in the source image(s). Furthermore, most of novel view synthesis frameworks focuses on synthesizing views from a single source image due to the difficulty of aggregating the understanding from multiple source images.

To step towards developing a framework that is able to address the task of multi-view novel view synthesis, we propose an end-to-end trainable framework (shown in Fig. 1) composed of two modules. The flow predictor estimates flow fields to move the pixels from a source view to a target view; the recurrent pixel generator, augmented with an internal memory, iteratively synthesizes and refines a target view when a new source view is given. We propose a self-confidence aggregation mechanism to integrate multiple intermediate predictions produced by the two modules to yield results that are both realistic and structurally consistent.

We compare our model against state-of-the-art methods on a variety of datasets such as 3D-object models as well as real and synthetic scenes. Our main contributions are as follows: we propose a hybrid framework which combines the strengths of two main lines of novel view synthesis methods and achieves significant improvement compared to existing work. We then demonstrate the flexibility of our method; we show that our model is able to synthesize views from a single source image as well as improve its predictions when additional source views are available. Furthermore, our model can be adapted to scenes rather than synthetic object data as it does not require 3D supervision.

2 Related Works

Geometry-based View Synthesis. A great amount of efforts have been dedicated to explicitly modeling the underlying 3D structure of both scenes and objects [13–16]. While appealing and accurate results are guaranteed when multiple source images are available, this line of work is fundamentally not able to deal with sparse inputs. Aiming to address this issue, a deep learning approach is proposed in [24] focusing on the multi-view stereo problem by regressing directly to output pixel values. On the other hand, [25] explicitly utilizes learned dense correspondences to predict the image in the middle view of a pair of source images. The above-mentioned methods are limited to synthesizing a middle view among source images and the number of source images is fixed; in contrast, our proposed framework focuses on arbitrary target views and is able to learn from source images vary in length.

Learning Dense Visual Correspondence. Discovering dense correspondences among images has been studied in [26–29] with a wide range of applications including depth estimation, optical flow prediction, image alignment, image retrieval, etc. Fundamentally differing from this task, novel view synthesis requires the ability to hallucinate pixels of the target image which are missing from source images.

Image Generation. A tremendous success in conditional image generation has been made with deep generative models. Given the style, viewpoint, and color of an object, the method proposed in [30] is able to render realistic results. However, their method is not able to generalize to novel objects or poses which are tackled in our proposed framework. Huang *et al.* [31] addressed the problem of synthesizing a frontal view face from a single side-view face image. The proposed model is specifically designed for face images. In contrast, our proposed framework is able to synthesize both scenes and objects.

Image-to-image Translation. The task of translating an image from a domain to another domain, known as image-to-image translation has recently received a significant amount of attention [32–35]. One can consider the task of novel view synthesis as an image-to-image translation problem where the target and source domains are defined by the camera poses. Not only are the view synthesis systems required to understand the representation of domain specifications *e.g.* camera poses, but also the numbers of source and target domains are possibly infinitely many due to the continuous representations of camera poses. Moreover, novel view synthesis requires the understanding of geometry while the task of image-to-image translation often only focuses on texture transfer.

3D Voxel/Point Cloud Prediction. Explicitly reconstructing 3D geometry has been intensively addressed in a multi-view setting, such as SfM and SLAM [13–16], in which we are interested in the case where plenty of images captured from different viewing angles are available. Recently, empowered by large-scale repositories of 3D CAD models such as ShapeNet [18], predicting 3D representations such as voxels and 3D point clouds from 2D views has achieved encouraging results [6,7]. By contrast, we are interested in synthesizing views instead of 3D representations of objects. Our approach requires no 3D supervision nor explicit 3D model.

Novel View Synthesis. [19,20] propose to directly generate pixels of a target view, while [22] re-casts the task of novel view synthesis as predicting dense flow fields that map the pixels in the source view to the target view, but it is not able to hallucinate the pixels which are missing from source view. [23] predicts a flow to move the pixels from the source to the target view, followed by an image completion network. There are three key differences between our work and [23]. First, [23] requires 3D supervision which limits the method to only objects; on the other hand, our model requires no 3D supervision and therefore is able to synthesize scenes. Second, we address the task where the source images vary in length while [23] focuses on a single source image. Third, we design our model to predict a flow and hallucinate pixels independently, which enables our framework to take advantage of both modules to produce structural consistent shape and sharper appearance. This design also makes our model end-to-end trainable. Instead, [23] considers it as a sequential process where the pixel generation network is only considered as a refinement network.

3 Approach

When synthesizing a novel view from multi-view input, we want our model to (1) directly reuse information from the source as well as hallucinate missing information; (2) progressively improve its prediction as more information is available. To put this idea into practice, we design a flexible neural network framework that progressively improves its prediction as more input information is presented. To put (1) into practice, we design our framework to be a two-stream model that consists of a flow predictor and a pixel generator (shown in Figure 1). The flow predictor learns to reuse the pixels presented in source images, while the pixel generator learns to hallucinate pixels. To take advantage of the strengths of both the modules as well as achieve (2), we aggregate intermediate predictions using a self-learned confidence aggregation mechanism.

3.1 Overview and Notations

Our goal is to synthesize a target image I_{target} given a target camera pose p_{target} and N (image, camera-pose) pairs $(I_s^1, p_s^1), (I_s^2, p_s^2)..., (I_s^N, p_s^N)$. We either use a one-hot vector to represent discrete camera-pose, or a 6DoF vector for continuous camera pose. We denote the flow predictor as $\mathcal{F}(\cdot)$, and denote the pixel generator as $\mathcal{P}(\cdot)$. We put a subscript f and p for predictions made by $\mathcal{F}(\cdot)$ and $\mathcal{P}(\cdot)$, respectively. Given t-th source image I_s^t and its corresponding pose p_s^t, the flow predictor generates a prediction $I_f^t, c_f^t = \mathcal{F}(p_{target}, I_s^t, p_s^t)$, where I_f^t is a predicted target image and c_f^t is the corresponding confidence map. The flow predictor independently produces N predictions from N source images since it learns to estimate the relative pixel movements from source viewpoint to the target viewpoint. The pixel generator, on the other hand, is designed as a recurrent model, which outputs a prediction $I_p^t, c_p^t = \mathcal{P}(p_{target}, I_s^1, p_s^1, ..., I_s^t, p_s^t)$ given t source images. I_p^t is the predicted target image and c_p^t is the corresponding confidence map. The final prediction $\hat{I}_{target}$ is generated by aggregating the $N+1$ predictions (N from the flow module and 1 from the pixel module).

3.2 Flow Predictor

Inspired by [22], we design a flow module that learns to predict dense flow fields. The output indicates the pixel displacement from the source image to the target image. Given t-th source image I_s^t, the model first predicts 2D dense flow fields from the original image in x and y-axis by $(x_t, y_t) = \mathcal{G}(I_s^t, p_s^t, p_{target})$. This flow field is then used to sample from the original image by $I_f^t = \mathcal{T}(x_t, y_t, I_s^t)$, where I_f^t denotes the predicted target image given I_s^t. Here $\mathcal{G}(\cdot)$ predicts the flow, and $\mathcal{T}(\cdot)$ bilinearly samples from the source image. This differentiable bilinear sampling layer was originally proposed by [36]. We optimize the flow predictor by minimizing the following equation:

$$\mathcal{L}_F = \frac{1}{N} \sum_{t=0}^{N} ||I_{target} - I_f^t||_1, \tag{1}$$

We use an encoder-decoder architecture with residual blocks and skip connections. The architecture details are left in the supplementary section. The encoder of this model takes a source image as well as its associated pose, where a pose vector is spatially tiled and concatenated to the source image channel-wise. The decoder upsamples the features to match the dimension of the input image. We empirically find that this architecture outperforms the architecture originally proposed in [22]. This comparisons can be found in Section 4.3.

3.3 Pixel Generator

The flow predictor is able to yield a visually appealing result when the source pose and the target pose are close – i.e. when the target is well represented by the source. Yet, it is not capable of generating pixels beyond the source pixels. Therefore, it is only natural to rely on the prior understanding of the underlying 3D structure.

The architecture of this module is very similar to our flow module. It is an encoder-decoder style network with an internal memory using Convolutional Long-Short Term Memory (ConvLSTM) [37], which is able to progressively improve its prediction with varying input lengths. Note that the ConvLSTMs are used only in the bottleneck layers and the mathematical formulation is left in the supplementary section. The pixel generator is trained to minimize the following equation:

$$\mathcal{L}_P = \frac{1}{N}\sum_{t=0}^{N} ||I_{target} - I_p^t||_1, \tag{2}$$

where I_p^t denotes a predicted target image by $I_p^t, c_p^t = \mathcal{P}(p_{target}, I_s^1, p_s^1, ..., I_s^t, p_s^t)$. To enforce our model to generate sharp images, we also incorporate an adversarial loss into our objective. We utilize the formulation proposed in [38], where an additional discriminator is trained to optimize:

$$\mathcal{L}_D = \mathbb{E}[(1 - D(I_{target}))^2] + \mathbb{E}[\frac{1}{N}\sum_{t=0}^{N}(D(I_p^t))^2]. \tag{3}$$

With the pixel generator minimizing the following additional loss:

$$\mathcal{L}_G = \mathbb{E}[\frac{1}{N}\sum_{t=0}^{N}(1 - D(I_p^t))^2]. \tag{4}$$

The final objective for the pixel module can be compactly represented as: $\mathcal{L}_P + \lambda\mathcal{L}_G$, where λ denotes the weight of the adversarial loss. The details of the discriminator architecture and GAN training can be found in the supplementary section.

3.4 Self-learned Confidence Aggregation

The flow module is able to produce visually realistic images by reusing the pixels from source images; however, synthesized images are often incomplete due to possible occlusions or pixels missing from source images. On the other hand, the pixel module is trained to directly hallucinate the target image and is able to produce structurally consistent results, their appearance is usually blurry due to the inherent ambiguity of minimizing a regression loss. Our key insight is to alleviate the disadvantages of the two modules by aggregating the advantages of both information. Inspired by the recent flourish of *Bayesian deep learning* [39, 40], where we are interested in modeling *uncertainty* of neural networks, we propose to train networks to predict *confidence*.

Specifically, we want an algorithm that is able to produce a per-pixel confidence map associated with its predictions. We formulate this confidence prediction objective as:

$$\mathcal{L}_C = \frac{1}{HW}\sum_{x,y} ||I_{target} - \hat{I}||^{\circ 2} \circ \frac{c}{\sum_{x,y} ||c||_2}, \tag{5}$$

where $\hat{I}$ is the predicted target image (either from flow or pixel module), and c is the estimated confidence map with a size of H by W. $||\cdot||^{\circ 2}$ is an element-wise square operator, $\circ$ is the Hadamard product. To minimize this objective, the models have to learn to put more weight on pixels where they are confident and less on regions it is not. Each module is augmented with an additional output layer to predict the confidence map. The confidence maps are optimized via the objective described in Equation 5.

We normalize the predicted confidences maps by applying a *Softmax* across $N+1$ confidence maps. The normalized confidence maps, denoted as $\hat{c}$, can then be used to aggregate the predictions: $\hat{I}_{target} = I_p^N \odot \hat{c}_p^N + \sum_{i=0}^{N} I_f^i \odot \hat{c}_f^i$. To iterate, $\hat{I}_{target}$ denotes the final aggregated image, I_p^N denotes the last output of the recurrent pixel generator, and I_f^i denotes the output of the flow predictor given the i-th source image. The reconstruction loss on the aggregated prediction is $\mathcal{L}_A = ||I_{target} - \hat{I}_{target}||_1$. The final objective of the full model is:

$$\min\ \beta\mathcal{L}_A + \overbrace{\mathcal{L}_F + \alpha_f\mathcal{L}_C}^{\text{Flow Prediction}} + \overbrace{\mathcal{L}_P + \lambda\mathcal{L}_G + \alpha_p\mathcal{L}_C}^{\text{Pixel Prediction}} \tag{6}$$

where α_f, α_p, are weights for confidence map predictions and β is the weight for the global confidence scale. The effectiveness and the gradual improvement of using confidence maps are demonstrated in Section 4. The architecture and training details can be found in Supplemental Material.

4 Experiments

We evaluate our model in multi-view and single-view settings on ShapeNet [18] objects, real-world scenes (KITTI Visual Odometry Dataset [41]), and synthesized scenes (Synthia dataset [42]). We benchmark against a pixel generation method [19], a flow prediction approach [22], and a state-of-the-art novel view synthesis framework [23]. We use L_1, and structural similarity measure (SSIM) as quantitative reconstruction metrics. Furthermore, to investigate whether our

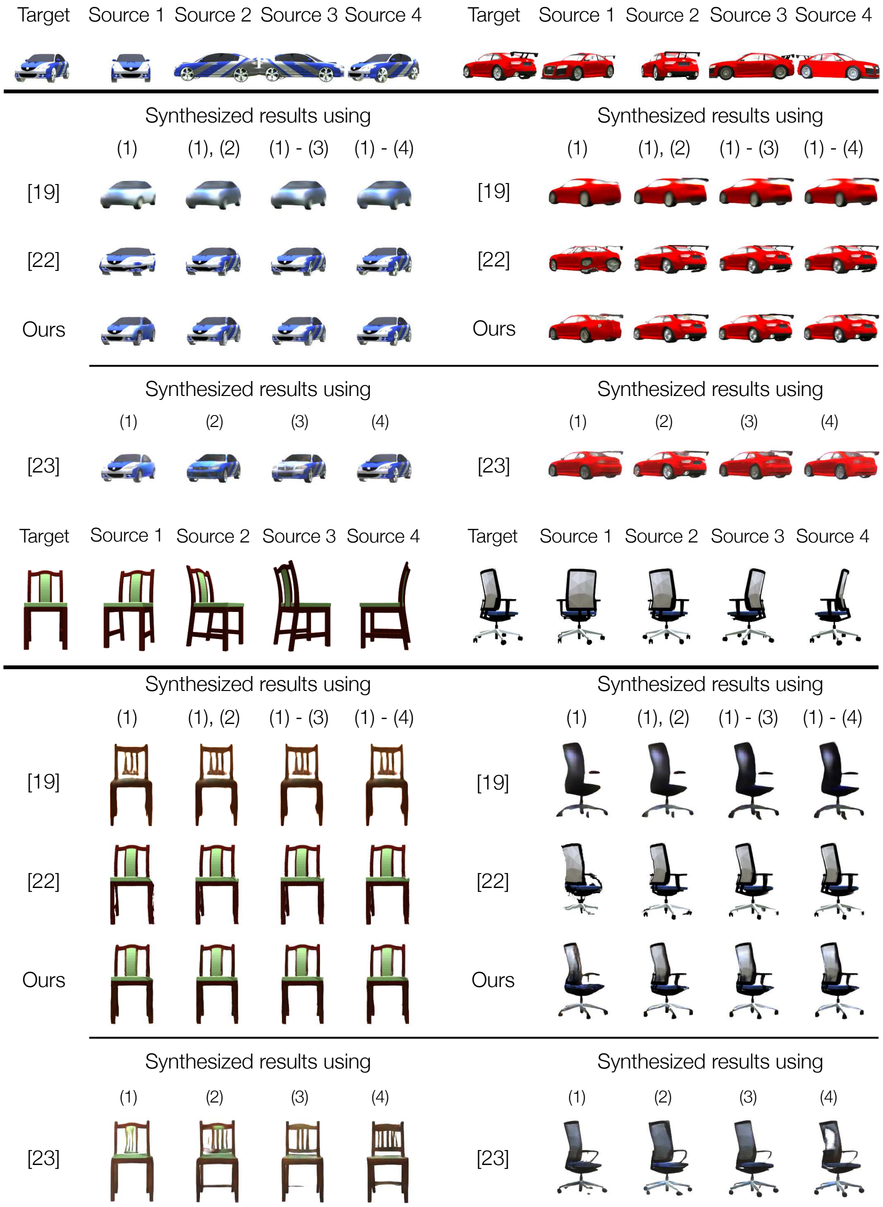

Fig. 2: Results on ShapeNet [18]. The proposed framework typically synthesized cars and chairs with correct shapes and realistic appearance. [19] generates structurally coherent but blurry images. [22] produces realistic results but suffers from distortions and missing pixels. [23] outperforms both [19] and [22] while sometimes produces unrealistic results.

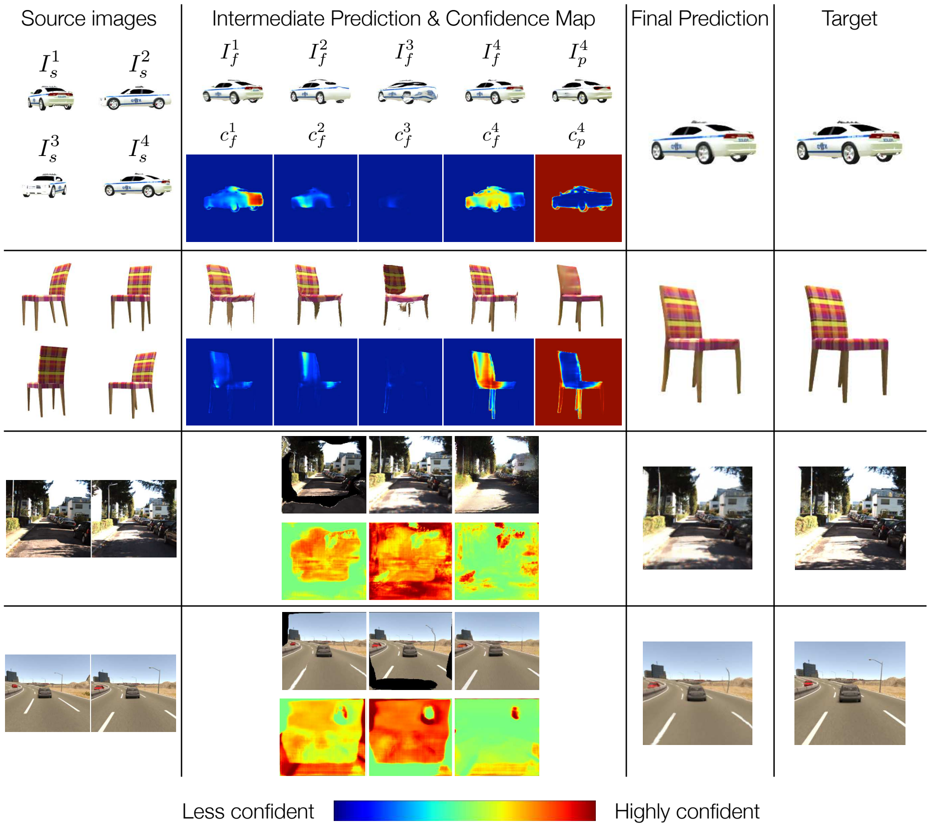

Fig. 3: Results and confidence maps generated from our proposed model. The first N intermediate predictions are produced by the flow predictor, and the last one is obtained by the pixel generator. Confidence maps are plotted with Jet colormap, where red means higher confidence and blue means lower confidence. This demonstrates that our model is able to adaptively exploit the information from different source poses with confidence maps.

model can synthesize semantically realistic images, we quantify our results using a segmentation score predicted by FCN [43] trained on Synthia dataset [42].

4.1 Novel view synthesis for objects

We train and test the proposed model on ShapeNet [18], where ground truth views of arbitrary camera poses are available.

Data setup We render images of 3D models from the car category and the chair category. For each model, we render images with the dimension of 256 × 256 for a total of 54 viewpoints, which corresponds to 18 azimuth angles (sampled in the range $[0, 340]$ with 20-degree increments) and the elevations of 0, 10, and 20.

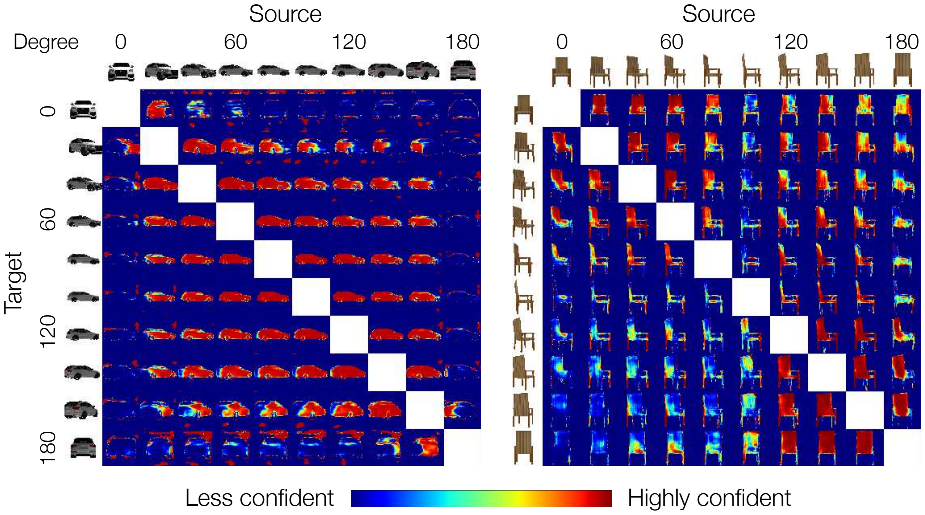

Fig. 4: Visualization of the predicted confidence maps for car and chair model. Each entry represents a predicted confidence map for a given source image and target pose. The confidence is represented using the jet-colormap, where red indicates highly confident, and blue indicating the otherwise.

The pose of each image is represented as a concatenation of two one-hot vectors: an 18 element vector indicating the azimuth angle and a 3 element vector indicating the elevation. We use the same training and testing splits used in [22,23] (80% of models for training and the rest 20% for testing). Each training/testing tuple is constructed by sampling a target pose as well as N source poses and their corresponding images $\langle I_{target}, p_t, I_s^1, p_s^1, .., I_s^N, p_s^N \rangle$. We randomly sample 20,000 tuples to create the testing split. N is set to 4 for this experiment.

Results The quantitative results are shown in Table 1 while the qualitative results can be found in Figure 2. The results demonstrate that our proposed model is able to reliably synthesize target images when single or multiple source images are available. Our model outperforms the three methods on both L_1 distance and SSIM. The pixel generation method [19] is capable of producing well-structured shapes but not appealing texture, while the flow prediction method [22] preserves realistic texture but is not able to hallucinate pixels missing from source. While the results produced by [23] are mostly satisfactory, when the flow module fails, the synthesized images generated by the refinement network usually either do not stay true to the source image – likely due to the adversarial loss – and is hugely distorted. Typically, our proposed framework is able to synthesize structurally consistent and realistic results by aggregating intermediate predictions with confidence maps.

We observe that the quality of the synthesized images of both cars and chairs improve as the number of source images increases. However, the marginal gain

Views	Methods	Car		Chair	
		L_1	SSIM	L_1	SSIM
1	[19]	.139	.875	.223	.882
	[22]	.148	.877	.229	.871
	[23]	.119	.913	.202	.889
	Ours	**.098**	**.923**	**.181**	**.895**
2	[19]	.124	.883	.209	.890
	[22]	.107	.901	.207	.881
	Ours	**.078**	**.935**	**.141**	**.911**
3	[19]	.116	.887	.197	.898
	[22]	.089	.915	.188	.887
	Ours	**.068**	**.941**	**.122**	**.919**
4	[19]	.112	.890	.192	.900
	[22]	.081	.924	.165	.891
	Ours	**.062**	**.946**	**.111**	**.925**

Table 1: ShapeNet objects: we compare our framework to [19], [22], and [23].

Views	Methods	Car		Chair	
		L_1	SSIM	L_1	SSIM
1	Pixel	.111	.911	.187	.892
	Flow	.119	.916	.208	.883
	Ours	**.098**	**.923**	**.181**	**.895**
2	Pixel	.095	.919	.148	.907
	Flow	.097	.927	.180	.890
	Ours	**.078**	**.935**	**.141**	**.911**
3	Pixel	.087	.923	.130	.915
	Flow	.086	.933	.164	.895
	Ours	**.068**	**.941**	**.122**	**.919**
4	Pixel	.082	.925	.119	.919
	Flow	.079	.938	.152	.900
	Oracle	.070	.941	.112	.923
	Ours	**.062**	**.946**	**.111**	**.925**

Table 2: Ablation study. We compare the performance of our full model to each module. Flow denotes the flow predictor and Pixel denotes the pixel generator.

decreases as the number of source images increases. This aligns with our intuition that each additional view contributes less new information since two random views are very likely to overlap with each other. Confidence maps and intermediate predictions shown in Figure 3 demonstrate that our model learns to adaptively exploit predictions produced by both of the two modules from multiple source images.

Learn to predict visibility maps without 3D supervision [23] trains the model to predict visibility maps, indicating which parts in a target image are visible from the source view. This requires prior 3D knowledge as it needs 3D coordinate and surface normal to produce ground truth visibility maps as supervision. With the predicted visibility maps, one is able to re-cast the remaining synthesis problem as image completion problem. On the other hand, as demonstrated in Figure 4, our model learns to predict confidence maps which share a similar concept of visibility maps without any 3D supervision. Specifically, our model is implicitly forced to comprehend which target pixels are presented in source images by learning to optimize the losses introduced in the proposed self-learned confidence aggregation mechanism. This is especially important in real life application where 3D supervision is most likely not available – allowing our model to be trained on not only objects but also scenes.

4.2 Novel view synthesis for scenes

While most of the existing novel view synthesis approaches only focus on ShapeNet, we are also interested in generalizing our proposed model to *scenes*, where 3D supervision is not available and training category-dependent models is not trivial. To this end, we train and test our framework on both real (KITTI Visual Odometry Dataset [41]) and synthetic (Synthia Dataset [42]) scenes.

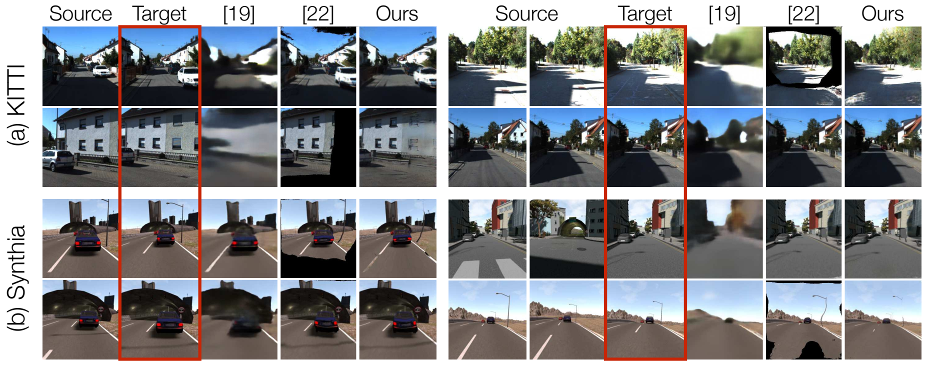

Fig. 5: Synthesized scenes on KITTI [41] and Synthia [42] datasets. Our framework typically produces structurally consistent and realistic results. Note that [22] struggles with distortions and missing pixels, and [19] is unable to generate sharp results.

KITTI The dataset [41] was originally proposed for SLAM evaluation. It contains frame sequences captured by a car traveling through urban city scenes with their camera poses. We use 11 sequences extracted from the dataset, whose ground truth camera poses are available, On average each sequence contains around two thousand frames. We use 80% frames for training and the rest of 20% for testing. We center-crop each frame to form an image with a dimension of 256×256. We convert each transformation matrix to its 6DoF representation (a translation vector and Euler angles) as a pose representation. We follow [22] to construct the training and testing set. We restrict the source frame and the target frame to be separated by at most 10 frames. To create the testing split, we randomly sample $20,000$ tuples. N is set to 2 for scene experiments.

Synthia The data was originally proposed for semantic segmentation in urban scenarios. Similar to [41], it contains realistic synthetic frame sequences captured by a driving car in a virtual world with their camera poses. We use sequences from all four seasons to train our model. We follow the same preprocessing procedures as KITTI to create the training and testing tuples.

Results As shown in Table 3, our proposed framework outperforms the two methods. Qualitative comparisons are shown in Fig. 5. Both [19] and [22] learn to infer the relative camera movements and synthesized scene accordingly. However, [19] hugely suffers from blurriness due to the uncertainty, while [22] is not able to produce satisfactory results when a camera pose changes drastically. Typically, our proposed framework is able to synthesize structurally consistent and realistic results. Also, the proposed framework does not suffer from the missing pixels by utilizing the scenes rendered by our proposed pixel generator. The two modules learn to leverage each others strength, as shown in Figure 3. We observed that none of the models perform well when some uncertainties are not able to be

Views	Methods	KITTI		Synthia	
		L_1	SSIM	L_1	SSIM
1	[19]	.295	.505	.175	.612
	[22]	.418	.504	.221	.636
	Ours	**.203**	**.626**	**.141**	**.697**
2	[19]	.283	.511	.172	.615
	[22]	.259	.626	.154	.702
	Ours	**.163**	**.691**	**.118**	**.737**

Table 3: Scenes: we compare our framework to [19] and [22] on KITTI and Synthia.

Views	Methods	KITTI		Synthia	
		L_1	SSIM	L_1	SSIM
1	Pixel	.259	.505	.183	.622
	Flow	.397	.539	.211	.652
	Ours	**.203**	**.626**	**.141**	**.697**
2	Pixel	.234	.525	.168	.628
	Flow	.249	.656	.149	.720
	Oracle	.199	.658	.140	.718
	Ours	**.163**	**.691**	**.118**	**.737**

Table 4: Ablation study. We compare the performance of our full model to each module. Flow denotes the flow predictor and Pixel denotes the pixel generator.

resolved purely based on source images and their pose. For instance, they include the speed of other driving cars, the lighting condition change, etc.

Semantic evaluation metrics Although the L_1 distance and SSIM are good metrics to measure the distance between a pair of images in the pixel domain, they often fail to capture the semantics of the generated images. Isola *et al.* [32] proposed to utilize a metric, similar to *inception score* [44], to measure the semantic quality of the synthesized images. Inspired by this, we evaluate our synthesized results using semantic segmentation score produced by a FCN [43] model trained on image semantic segmentation. We obtained the pretrained segmentation model trained on PASCAL VOC dataset [45] and then fine-tuned it on the sequences extracted from Synthia dataset [42] with the same training and testing split used in our view synthesis task. The FCN scores are shown in Table. 5 and the qualitative results are shown in Fig. 6.

4.3 Ablation Study

To investigate how different blocks of the framework affect the final outcomes, we conduct ablation studies on all the datasets. The qualitative results including intermediate predictions by the two modules can be found in Figure 3. The quantitative results can be found in Table 2 and Table 4, where *Flow* denotes aggregated predictions made by the flow predictor with its predicted confidence maps *e.g.* $\sum_{i=0}^{N} I_f^i \odot \hat{c}_f^i$, where $\hat{c}$ is softmaxed across only $c_f^1, ..., c_f^N$. Note that this does not use the image synthesized by the pixel generator. *Pixel* denotes the last results produced by the pixel generator, *e.g.* I_p^N.

One could argue that our model just learns to pick the best intermediate prediction. Hence, to investigate whether our model actually learns a meaningful self-confidence aggregation by comparing against an oracle. We quantify the best intermediate result from all $2N$ intermediate predictions produced by both modules. We denote this as the *Oracle* intermediate performance, *e.g.* $\min ||I_{target} - \hat{I}||_1 \quad \forall \hat{I} \in \{I_f^1, I_p^1, ..., I_f^N, I_p^N\}$.

We observed that our full model outperforms each module and the oracle. Also, our flow module with a fully convolutional architecture and residual blocks

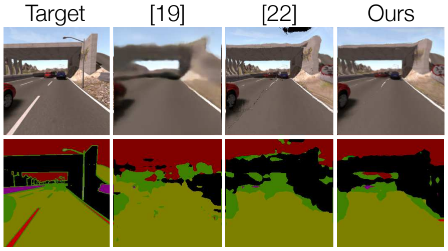

Fig. 6: Synthia FCN-results for the scenes synthesized by [19], [22], and our framework. The results produced by our framework yield better segmentation maps.

Methods	Per-pixel acc.	Per-class acc.	IOU
[19]	0.630	0.469	0.211
[22]	0.789	0.69	0.427
Ours	**0.803**	**0.695**	**0.441**
Ground Truth	0.868	0.783	0.586

Table 5: FCN-scores for different methods. The scores are evaluated by FCN-32 pretrained on PASCAL VOC and fine-tuned on Synthia dataset. The scores are estimated on synthesized scenes produced by [19], [22], and our proposed framework with one input view.

outperforms [22]. Our method is able to alleviate the issue of severe distortions reported in [22]. Our proposed recurrent pixel generator not only outperforms [19] but also show greater improvement (car: 26%, chair: 36%, KITTI: 10%, Synthia: 8%) when more source images are available compared to [19] (car: 19%, chair: 14%, KITTI: 4%, Synthia: 2%), which demonstrates the effectiveness of the recurrent pixel generator.

5 Conclusion

In this paper, we present an end-to-end trainable framework that is capable of synthesizing a novel view from multiple source views without utilizing 3D supervision. Specifically, we propose a two-stream model that integrates the strengths of the two main lines of existing view synthesis techniques: pixel generation and flow prediction. To adaptively merge the predictions produced by the two modules given multiple source images, we introduce a self-learned confidence aggregation mechanism. We evaluate our model on images rendered from 3D object models as well as real and synthesized scenes. We demonstrate that our model is able to achieve state-of-the-art results as well as progressively improve its predictions when more source images are available.

Acknowledgments

This project was supported by the center for super intelligence, Kakao Brain, and SKT. This research of Shao-Hua Sun and Minyoung Huh were partially supported by Snap Inc. The authors are grateful to Youngwoon Lee and Hyeonwoo Noh for helpful discussions about this work.

References

1. Poppe, R.: A survey on vision-based human action recognition. Image and vision computing (2010) 2
2. Gavrila, D.M., Davis, L.S.: 3-d model-based tracking of humans in action: a multi-view approach. In: Computer Vision and Pattern Recognition (CVPR). (1996) 2
3. Junejo, I.N., Dexter, E., Laptev, I., Pérez, P.: Cross-view action recognition from temporal self-similarities. In: European Conference on Computer Vision, Springer (2008) 293–306 2
4. Seitz, S.M., Curless, B., Diebel, J., Scharstein, D., Szeliski, R.: A comparison and evaluation of multi-view stereo reconstruction algorithms. In: Computer vision and pattern recognition (CVPR). (2006) 2
5. Furukawa, Y., Ponce, J.: Accurate, dense, and robust multiview stereopsis. IEEE transactions on pattern analysis and machine intelligence **32**(8) (2010) 1362–1376 2
6. Choy, C.B., Xu, D., Gwak, J., Chen, K., Savarese, S.: 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In: European Conference on Computer Vision (ECCV). (2016) 2, 4
7. Fan, H., Su, H., Guibas, L.: A point set generation network for 3d object reconstruction from a single image. In: Conference on Computer Vision and Pattern Recognition (CVPR). (2017) 2, 4
8. Remondino, F., El-Hakim, S.: Image-based 3d modelling: A review. The Photogrammetric Record (2006) 2
9. Zwicker, M., Pauly, M., Knoll, O., Gross, M.: Pointshop 3d: An interactive system for point-based surface editing. In: ACM Transactions on Graphics (TOG). (2002) 2
10. Seitz, S.M., Dyer, C.R.: View morphing. In: Special Interest Group on GRAPHics and Interactive Techniques (SIGGRAPH). (1996) 2
11. Chen, S.E.: Quicktime vr: An image-based approach to virtual environment navigation. In: Special Interest Group on GRAPHics and Interactive Techniques (SIGGRAPH). (1995) 2
12. Szeliski, R., Shum, H.Y.: Creating full view panoramic image mosaics and environment maps. In: Special Interest Group on GRAPHics and Interactive Techniques (SIGGRAPH). (1997) 2
13. Forsyth, D., Ponce, J.: Computer vision: a modern approach. Pearson (2011) 2, 3, 4
14. Sturm, P., Triggs, B.: A factorization based algorithm for multi-image projective structure and motion. (1996) 2, 3, 4
15. Montemerlo, M., Thrun, S., Koller, D., Wegbreit, B., et al.: Fastslam: A factored solution to the simultaneous localization and mapping problem. In: Aaai/iaai. (2002) 2, 3, 4
16. Durrant-Whyte, H., Bailey, T.: Simultaneous localization and mapping: part i. IEEE robotics & automation magazine (2006) 2, 3, 4
17. Debevec, P.E., Taylor, C.J., Malik, J.: Modeling and rendering architecture from photographs: A hybrid geometry-and image-based approach. In: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques. (1996) 2
18. Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., Xiao, J., Yi, L., Yu, F.: Shapenet: An

information-rich 3d model repository. Technical Report arXiv:1512.03012 [cs.GR] (2015) 2, 4, 7, 8, 9

19. Tatarchenko, M., Dosovitskiy, A., Brox, T.: Single-view to multi-view: Reconstructing unseen views with a convolutional network. CoRR abs/1511.06702 (2015) 2, 4, 7, 8, 10, 11, 12, 13, 14
20. Yang, J., Reed, S.E., Yang, M.H., Lee, H.: Weakly-supervised disentangling with recurrent transformations for 3d view synthesis. In: Advances in Neural Information Processing Systems. (2015) 1099–1107 2, 4
21. Rematas, K., Nguyen, C.H., Ritschel, T., Fritz, M., Tuytelaars, T.: Novel views of objects from a single image. IEEE transactions on pattern analysis and machine intelligence (2017) 2
22. Zhou, T., Tulsiani, S., Sun, W., Malik, J., Efros, A.A.: View synthesis by appearance flow. In: European Conference on Computer Vision (ECCV). (2016) 2, 4, 5, 6, 7, 8, 10, 11, 12, 13, 14
23. Park, E., Yang, J., Yumer, E., Ceylan, D., Berg, A.C.: Transformation-grounded image generation network for novel 3d view synthesis. In: CVPR. (2017) 2, 4, 7, 8, 10, 11
24. Flynn, J., Neulander, I., Philbin, J., Snavely, N.: Deepstereo: Learning to predict new views from the world's imagery. In: Computer Vision and Pattern Recognition (CVPR). (2016) 3
25. Ji, D., Kwon, J., McFarland, M., Savarese, S.: Deep view morphing. In: Computer Vision and Pattern Recognition (CVPR). (2017) 3
26. Zhou, T., Krahenbuhl, P., Aubry, M., Huang, Q., Efros, A.A.: Learning dense correspondence via 3d-guided cycle consistency. In: Computer Vision and Pattern Recognition (CVPR). (2016) 3
27. Godard, C., Mac Aodha, O., Brostow, G.J.: Unsupervised monocular depth estimation with left-right consistency. In: Computer Vision and Pattern Recognition (CVPR). (2017) 3
28. Ren, Z., Yan, J., Ni, B., Liu, B., Yang, X., Zha, H.: Unsupervised deep learning for optical flow estimation. In: AAAI. (2017) 1495–1501 3
29. Liu, C., Yuen, J., Torralba, A., Sivic, J., Freeman, W.T.: Sift flow: Dense correspondence across different scenes. In: European conference on computer vision (ECCV). (2008) 3
30. Dosovitskiy, A., Springenberg, J.T., Tatarchenko, M., Brox, T.: Learning to generate chairs, tables and cars with convolutional networks. IEEE transactions on pattern analysis and machine intelligence (PAMI) (2017) 4
31. Huang, R., Zhang, S., Li, T., He, R.: Beyond face rotation: Global and local perception gan for photorealistic and identity preserving frontal view synthesis. arXiv preprint arXiv:1704.04086 (2017) 4
32. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. arXiv preprint arXiv:1611.07004 (2016) 4, 13
33. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. arXiv preprint arXiv:1703.10593 (2017) 4
34. Liu, M.Y., Breuel, T., Kautz, J.: Unsupervised image-to-image translation networks. arXiv preprint arXiv:1703.00848 (2017) 4
35. Kim, T., Cha, M., Kim, H., Lee, J., Kim, J.: Learning to discover cross-domain relations with generative adversarial networks. arXiv preprint arXiv:1703.05192 (2017) 4
36. Jaderberg, M., Simonyan, K., Zisserman, A., et al.: Spatial transformer networks. In: Advances in Neural Information Processing Systems (NIPS). (2015) 5

37. Xingjian, S., Chen, Z., Wang, H., Yeung, D.Y., Wong, W.k., Woo, W.c.: Convolutional lstm network: A machine learning approach for precipitation nowcasting. In: Advances in Neural Information Processing Systems. (2015) 802–810 6
38. Mao, X., Li, Q., Xie, H., Lau, R.Y., Wang, Z., Smolley, S.P.: Least squares generative adversarial networks. In: ICCV. (2017) 6
39. Gal, Y., Ghahramani, Z.: Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In: international conference on machine learning. (2016) 7
40. Kendall, A., Gal, Y.: What uncertainties do we need in bayesian deep learning for computer vision? In: Advances in Neural Information Processing Systems. (2017) 7
41. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: Conference on Computer Vision and Pattern Recognition (CVPR). (2012) 7, 11, 12
42. Ros, G., Sellart, L., Materzynska, J., Vazquez, D., Lopez, A.M.: The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In: Computer Vision and Pattern Recognition (CVPR). (2016) 7, 9, 11, 12, 13
43. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 3431–3440 9, 13
44. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X.: Improved techniques for training gans. In: Advances in Neural Information Processing Systems. (2016) 2234–2242 13
45. Everingham, M., Eslami, S.M.A., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The pascal visual object classes challenge: A retrospective. International Journal of Computer Vision (2015) 13