Structure-Preserving Stereoscopic View Synthesis with Multi-Scale Adversarial Correlation Matching

Yu Zhang[1,2], Dongqing Zou[1]*, Jimmy S. Ren[1], Zhe Jiang[1], Xiaohao Chen[1]
[1]SenseTime Research [2]Tsinghua University
{zhangyu1,zoudongqing,rensijie,jiangzhe,chenxiaohao}@sensetime.com

Abstract

This paper addresses stereoscopic view synthesis from a single image. Various recent works solve this task by reorganizing pixels from the input view to reconstruct the target one in a stereo setup. However, purely depending on such photometric-based reconstruction process, the network may produce structurally inconsistent results.

Regarding this issue, this work proposes Multi-Scale Adversarial Correlation Matching (MS-ACM), a novel learning framework for structure-aware view synthesis. The proposed framework does not assume any costly supervision signal of scene structures such as depth. Instead, it models structures as self-correlation coefficients extracted from multi-scale feature maps in transformed spaces. In training, the feature space attempts to push the correlation distances between the synthesized and target images far apart, thus amplifying inconsistent structures. At the same time, the view synthesis network minimizes such correlation distances by fixing mistakes it makes. With such adversarial training, structural errors of different scales and levels are iteratively discovered and reduced, preserving both global layouts and fine-grained details. Extensive experiments on the KITTI benchmark show that MS-ACM improves both visual quality and the metrics over existing methods when plugged into recent view synthesis architectures.

1. Introduction

3D display is becoming universal nowadays. Automatic conversion of the rich 2D images and videos to 3D is now a demand that can benefit various industrial fields. To fulfill this demand, binocular views are rendered to form the stereoscopic format for an input scene, while only one of them is known beforehand. Such single-image based view synthesis problem, however, is still challenging.

In its early research, view synthesis is often based on at least two known views (or continuous video sequences), so that the 3D scene geometry is well-defined [30, 38, 14]. For a single input view, the gap of 3D understanding is filled very recently by the strong statistical modeling power of deep learning. Among these methods, 3D view transformations are formulated as 2D warping fields (*e.g.* pixel flows [42, 13, 23], spatially-variant kernels [39, 21], or homographies [15]), which guide the target view to "copy" pixels from the input image. Photometric reconstruction errors across views are usually adopted to supervise this process in training. However, as such loss functions optimize

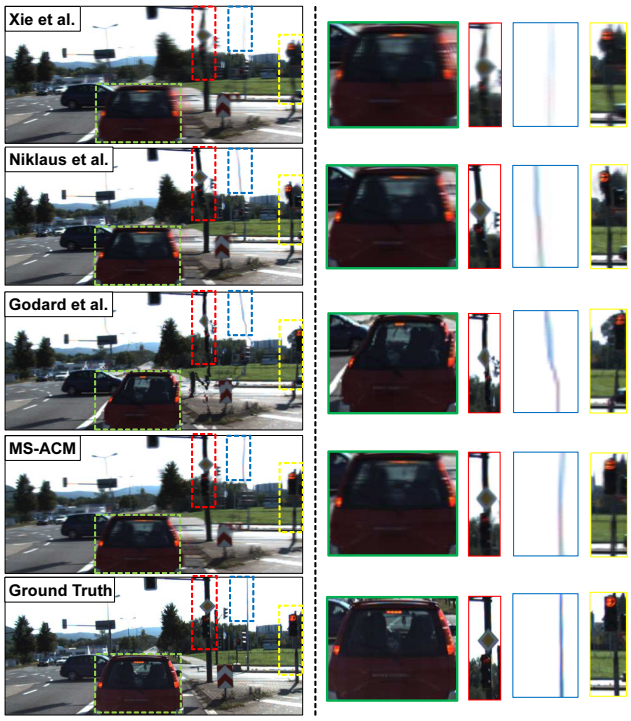

Figure 1. Structure preservation for view synthesis. Photometric losses commonly adopted by existing approaches (*e.g.*, Xie *et al.* [39], Niklaus *et al.* [21] and Godard *et al.* [9]) often lead to blurred and distorted structures, which is more severe for thin, unsalient objects. The proposed MS-ACM addresses this limitation via a novel adversarial training process that accounts for both large and fine-grained structures. Best viewed in color with zoom.

*Correspondence should be addressed to zoudonqging@sensetime.com

color consistency in average statistics, structure degeneration often happens as blurred, distorted details. It harms especially the objects from the "minority", *e.g*., the small and thin poles with ambiguous appearance shown in Fig. 1.

To maintain structural consistency during view synthesis, various methods leverage explicit supervisions from the 3D world in addition to the photometric consistency. It finds forms of scene depths/normals [15, 43], multi-view inputs [7, 13, 33], and 3D correspondences from CAD models [32, 23, 27]. Despite the rich 3D information, either of these is costly and difficult to obtain. Moreover, 3D supervision is only restricted to a small number of scene/object types, limiting the model's applicability in the wild.

In this paper we propose Multi-Scale Adversarial Correlation Matching (MS-ACM), a novel approach for learning stereoscopic view synthesis. MS-ACM learns the structural priors directly from data, instead of assuming any costly form of 3D supervisions. In the proposed approach, a *structure critic network* is appended to the view synthesis one, which transforms the synthesized and target views into latent feature spaces for structure matching. Each feature location computes normalized correlations within its surrounding window, whose responses serve as surrogates of local structural configurations. By training the critic network to maximize the distances of correlation coefficients between the synthesized and target views, it learns to amplify any structural mistakes it sees. This in turn guides the view synthesis network to correct its mistakes by asking it to minimize the same distance. Such adversarial training is performed on multi-scale feature maps, so as to be aware of both coarse-level and fine-grained structures. To avoid getting to bad minima, novel strategies are proposed to make the critic network adapted to high-level structures and robust to subtle noise. We show the effectiveness of MSACM by plugging it into two recent representative view synthesis architectures [39, 21]. Extensive results on the challenging KITTI benchmark [8] demonstrate that MS-ACM improves visual quality as well as quantitative metrics.

This paper makes the following contributions:

1) We propose a novel adversarial training framework for structure-preserving stereoscopic view synthesis. It is friendly to various existing view synthesis models, improving both their performance and generalizability.

2) Correlation based structure representation is proposed for adversarial training, which effectively captures scene structures at different scales. Various strategies are presented to avoid bad local minima as well.

2. Related Works

Rendering novel viewpoints of a given scene was solved with multi-view geometry for more than two decades. Performing this task with a single image, however, is relatively new. This section briefly reviews these related approaches.

Multiple-view based synthesis assumes the input scene is given from multiple known viewpoints. Rich physical 3D scene structure is provided in this manner, such that correspondences across views can be explicitly established. This idea arises since 90's [19, 30, 1]. Later works improved this pipeline by proposing stronger 3D scene representations [35, 24], better occlusion handling models [17, 5] and more powerful texture transfer techniques [25, 37]. Besides static scene modeling, view synthesis in videos was also extensively explored to facilitate stabilization tasks [14, 3]. Recent deep learning methods propose to learn direct multi-to-novel view synthesis functions [7, 33, 22, 20]. Although multi-view inputs provide more comprehensive understanding of the 3D structure, it does not fit many applications, especially those based on a single view.

Single-view based synthesis, on the other hand, generates novel views based on only a single image. Various approaches first infer the scene geometry (*e.g*. depths and normals [15, 43], then synthesize target views with geometry-grounded view transformations. CAD models as another form of geometrical signal for object-level novel view synthesis [27, 23, 32, 41]. However, while scene depth/normal is costly to collect, CAD models are limited to object categories and provide little knowledge to scene understanding.

On the other hand, several works advocate a self-taught learning process that directly reorganizes pixels from the input image to match the target one [42, 39, 34], without depending on explicit geometrical supervisions. The rationale behind is that the collective power of massive training data provides regularizations on the learned view transformations. Similar idea has also been explored for other tasks, including depth estimation [9] and visual tracking [36]. However, usually the only training signals are average photometric errors. Such errors focus on preserving the structures of majority cases but may neglect uncommon scenarios, leading to over-smoothed details distortion.

Structure regularization with adversarial training has been explored recently on image segmentation [18, 40, 11]. In these works, the network outputs and groundtruth segmentations are fed into a shared structure analysis network, which is adversarially trained to exaggerate prediction errors. The proposed idea is inspired from this line of works, but has two novel aspects. First, we process high-dimensional signals (*i.e*. the synthesized images), instead of low-dimensional segmentation maps. Novel strategies are introduced to stabilize training and get rid of bad local optimum. Second, rather than training on feature-space ℓ_1 distances, we propose to adopt feature correlations as the structure surrogate. In this manner, the network is encouraged to discover high-level edges in the scene, allowing structure-related mid-level representations to be more easily learned.

3. The Proposed Approach

3.1. Adversarial Correlation Matching

Before delving into our view synthesis framework, we first introduce Adversarial Correlation Matching (ACM), a novel adversarial training process for structure-aware learning. The proposed framework consists of a structure predictor $\mathcal{P}$ and a critic network $\mathcal{S}$. The predictor takes an input $\mathbf{x}$ and generates a structured output $\mathbf{y}$, *i.e.* $\mathbf{y} = \mathcal{P}(\mathbf{x}; \mathbf{w}_{\mathcal{P}})$, controlled by model parameters $\mathbf{w}_{\mathcal{P}}$. For example, in stereoscopic view synthesis the input is a left-view image, and the output is its right view. The structure critic network $\mathcal{S}$ takes responsibility of transforming $\mathbf{y}$ into a latent feature space for structure analysis, *i.e.* $\mathbf{f} = \mathcal{S}(\mathbf{y}; \mathbf{w}_{\mathcal{S}})$. We assume that $\mathbf{f}$ takes the form of convolutional feature maps with spatial information preserved. For a spatial location $\mathbf{p}$, its feature is accessed by $\mathbf{f}(\mathbf{p})$.

In this learned feature space, ACM models structure as mutual correlations among different spatial locations. More specifically, for each location $\mathbf{p}$, its local structure configuration is represented by the feature cosine distances computed with its spatial neighbours:

$$\mathbf{c}(\mathbf{p}) = \mathrm{vec}\left(\left\{ \frac{\mathbf{f}(\mathbf{p})^{\mathrm{T}} \mathbf{f}(\mathbf{q})}{\|\mathbf{f}(\mathbf{p})\|_2 \|\mathbf{f}(\mathbf{q})\|_2} \right\}_{\mathbf{q} \in \mathbb{N}_k(\mathbf{p})} \right), \quad (1)$$

where $\mathbb{N}_k(\mathbf{p})$ is the set of neighbour locations of $\mathbf{p}$ within a k-sized spatial window, and $\|\cdot\|_2$ denotes the ℓ_2 norm. The $\mathrm{vec}(\cdot)$ operation reorganizes input values into a vector. With the structure representation of the synthesized image $\mathbf{c}$, we can now quantize errors with that of groundtruth. To this end, groundtruth of $\mathbf{y}$, denoted by $\mathbf{y}_g$, is fed into the same $\mathcal{S}$ and produces structure representations $\mathbf{c}_g$. The structural error is thus measured by

$$d_s(\mathbf{y}, \mathbf{y}_g) = \frac{1}{|\mathbb{P}|} \sum_{\mathbf{p} \in \mathbb{P}} \|\mathbf{c}(\mathbf{p}) - \mathbf{c}_g(\mathbf{p})\|_1, \quad (2)$$

i.e. the average ℓ_1 distance over all the feature locations $\mathbb{P}$. For simplicity, we refer (2) to the corr-ℓ_1 distance.

In adversarial training, the structure critic network $\mathcal{S}$ pursues a feature space that best distinguishes between $\mathbf{y}$ and $\mathbf{y}_g$ by maximizing (2). Meanwhile, the prediction network $\mathcal{P}$ attempts to produce structured output $\mathbf{y}$ that can minimize it. In this manner, it is expected that any structural difference can be amplified during training, which in turn provides sufficient signals to supervise predictor training.

In the following, we provide several remarks on ACM.

Link to self-similarity. The proposed approach correlates with the concept of self-similarity for visual matching established before a decade [31]. Self-similarity assigns each image location a descriptor that characterizes its local layout patterns, computed by comparing a template window with a larger search region around the same location. In this manner, per-image textures are filtered out and only structural configurations are kept, making the matching process robust. Our structure representation (1) fits this idea and can be considered as normalized correlations between a size-1 template and a search window.

Intuitions behind corr-ℓ_1 distance. Previous works advocate using feature ℓ_1 distance for adversarial structure learning [40, 11], *i.e.* $\frac{1}{|\mathbb{P}|} \sum_{\mathbf{p} \in \mathbb{P}} \|\mathbf{f}(\mathbf{p}) - \mathbf{f}_g(\mathbf{p})\|_1$. Intuitively, corr-$\ell_1$ loss explicitly models local structural patterns, which should mitigate the difficulty of encoding structures directly into features. By computing cosine similarities among features, only feature-level "edges" are preserved while impact of other factors is reduced. This would save a great power of network capacities in learning textures, brightness, etc., that are irrelevant to scene structures. Another shortage of ℓ_1 loss, when applied for adversarial training, is its sensitiveness to the magnitude of features. It says that when $\mathcal{S}$ maximizes feature distance, it tends to scale the feature magnitudes up and make training unstable, as recognized in both [40] and [11]. Weight clipping was adopted to prevent this issue, introducing difficulty in parameter tuning and limiting the model's capacity. Instead, corr-ℓ_1 is a bounded, magnitude-insensitive loss. Thus, the network does not need to scale up features to conform the training objective. Recent findings also support this claim and show its positive effect for stabilizing training [16].

3.2. Getting Rid of Bad Minima

Discriminator in adversarial networks easily gets stuck into bad local minima when trained on high dimensional signals [26]. There is no exception for ACM as in tasks like view synthesis, the structure critic network operates on color images. We address this issue as follows.

Introducing robustness to noise. The prediction $\mathbf{y}$ and groundtruth $\mathbf{y}_0$ often have an inherent distribution gap depending on the generation process of the predictor $\mathcal{P}$. For example, the synthesized pixels of the predicted view are usually more correlated than those in groundtruth, due to the interpolation or warping operations during view synthesis. They can also differ in lighting and textures caused by camera len settings and the data capture environment. If the critic network notices them, it pushes the predictions and groundtruths into bad modes far away in feature space, and contributes nothing to learning.

In training GANs, such distribution gap problem was actively studied and a working trick is Instance Noise [2]. We adapt this idea into ACM as follows. When training $\mathcal{S}$, we add random noises into the groundtruth $\mathbf{y}_g$ to generate $\mathbf{y}_n$, and feed it into $\mathcal{S}$ to get the structure representation $\mathbf{c}_n$. We ask $\mathcal{S}$ to learn noise resistant features, by constraining $\mathbf{c}_n$ to

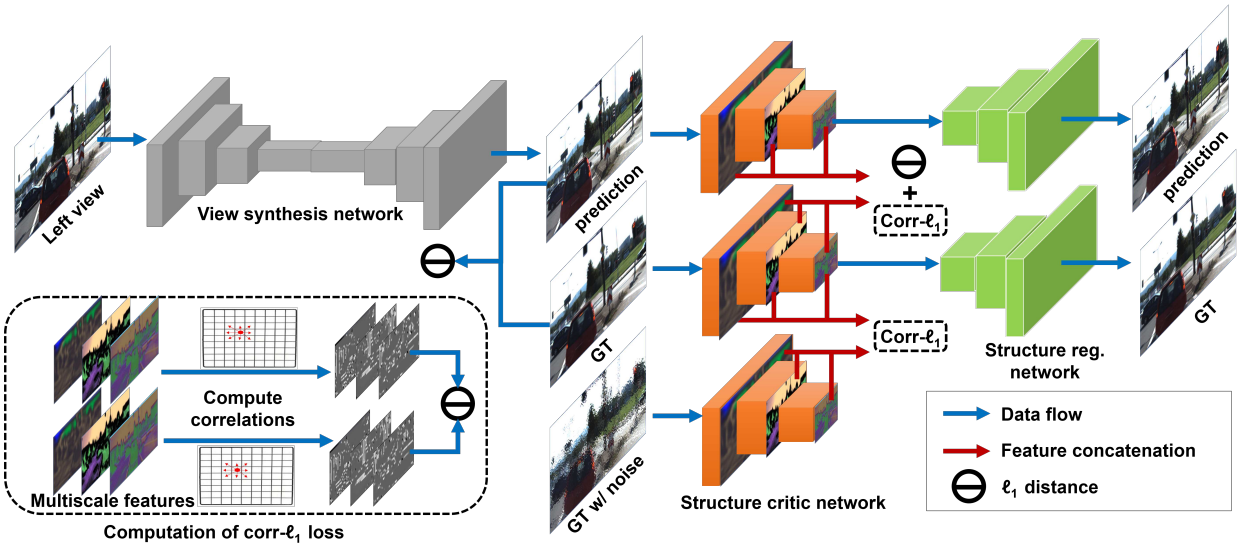

Figure 2. The proposed framework for stereoscopic view synthesis. The view synthesis network predicts the synthesized view of the input image, which is fed into the structure critic network along with its groundtruth to produce mutli-scale feature maps. Meanwhile, a noisy version of the groundtruth image goes through the same procedure. During training, the view synthesis network minimizes the pixel ℓ_1 distance, the ℓ_1 and corr-ℓ_1 distances of extracted feature maps between the synthesized image and groundtruth. The structure critic network maximizes the same corr-ℓ_1 distance, while minimizing it between the groundtruth and its noisy transform. At the same time, the extracted feature maps reconstruct the inputs with a regularization network jointly trained with the critic. Best viewed in color.

be close to $\mathbf{c}_g$. It equals to minimizing

$$d_n\left(\mathbf{y}_g, \mathbf{y}_n\right) = \frac{1}{|\mathbb{P}|}\sum_{\mathbf{p}\in\mathbb{P}} \left\|\mathbf{c}_g\left(\mathbf{p}\right) - \mathbf{c}_n\left(\mathbf{p}\right)\right\|_1. \tag{3}$$

In this manner, predictor/dataset-specific characteristics are broken by noise, forcing $\mathcal{S}$ aware of the real image content.

Making features content-aligned. Although in principle $\mathcal{S}$ finds any differences between two images, it is better to make learned features align with the inputs. This idea was originally proposed by Hwang *et al*. [11], which facilitates the network to learn good structure basis more effectively. To this end, a structure regularization network $\mathcal{R}$ is appended behind $\mathcal{S}$, which consumes its output features and reconstructs the input image. Networks $\mathcal{R}$ and $\mathcal{S}$ are jointly trained, minimizing the ℓ_1 reconstruction loss

$$d_r\left(\mathbf{y}, \mathbf{y}_g\right) = \left\|\mathbf{y} - \mathcal{R}\left(\mathbf{c}; \mathbf{w}_{\mathcal{R}}\right)\right\|_1 + \left\|\mathbf{y}_g - \mathcal{R}\left(\mathbf{c}_g; \mathbf{w}_{\mathcal{R}}\right)\right\|_1. \tag{4}$$

Closing the gap of feature scaling. Since corr-ℓ_1 is insensitive of feature magnitudes, there exists a potential risk of overfitting. Imagine that $\mathcal{S}$ pushs the predictions and groundtruths into different feature spaces with their own scale of magnitude, but correlation values are still the same. If this happens, optimizing structure distance in two different feature spaces may generate unpredictable results. To prevent this from happening, we train the predictor $\mathcal{P}$ to pursue the feature space of groundtruth:

$$d_f\left(\mathbf{y}, \mathbf{y}_g\right) = \frac{1}{|\mathbb{P}|}\sum_{\mathbf{p}\in\mathbb{P}} \left\|\mathbf{f}\left(\mathbf{p}\right) - \mathbf{f}_g\left(\mathbf{p}\right)\right\|_1. \tag{5}$$

In summary, the ACM training objective for $\mathcal{C}$ is

$$\begin{aligned}\max_{\mathbf{w}_{\mathcal{C}}, \mathbf{w}_{\mathcal{R}}} L_{\mathcal{C}}\left(\mathbf{y}, \mathbf{y}_g, \mathbf{y}_n\right) = &-\lambda_n d_s\left(\mathbf{y}_n, \mathbf{y}_g\right) \\ &- \frac{\lambda_r}{2} d_r\left(\mathbf{y}, \mathbf{y}_g\right) + d_s\left(\mathbf{y}, \mathbf{y}_g\right),\end{aligned} \tag{6}$$

where λ_n and λ_r are positive weights. For $\mathcal{P}$, the training objective is defined by

$$\min_{\mathbf{w}_{\mathcal{P}}} L_{\mathcal{P}}\left(\mathbf{y}, \mathbf{y}_g\right) = d_s\left(\mathbf{y}, \mathbf{y}_g\right) + d_f\left(\mathbf{y}, \mathbf{y}_g\right). \tag{7}$$

In the rest of this section, we show how ACM is instantiated in solving stereoscopic view synthesis.

3.3. View Synthesis with Multi-Scale ACM

The proposed training framework for stereoscopic view synthesis is summarized in Fig. 2. In this framework, the view synthesis network takes a left view as input and reorganizes its pixels to generate a predicted right view. The predicted view, groundtruth, and a noisy version of groundtruth are fed into the critic network for structure analysis. During testing, only the view synthesis network is kept and other parts are discarded.

The view synthesis network can be implemented with various existing architectures [42, 39, 21]. It is trained with the ℓ_1 photometric reconstruction loss as well as the ACM loss (7). The structure critic network $\mathcal{S}$ and regularization network $\mathcal{R}$ from a encoder-decoder structure, for which we

adopt U-Net [28]. It consists of three downsampling stages, and three upsampling ones. Each downsampling stage has two convolution layers interleaved with Leaky ReLU nonlinearity. Average pooling is applied after each stage. As such, the structure critical network actually provides feature maps of three scales. We perform ACM at each scale to capture structures at different granularities. We refer this extended version of ACM to Multi-Scale ACM (MS-ACM).

The training algorithm. Following the practice of training GANs [10], we alternate updating $\mathcal{P}$ and $\mathcal{S}$ till convergence. At each training step, the groundtruth is transformed by three types of noises: additive Gaussian noise, Gaussian blur and random pixel shifts, as well as their combinations. For random pixel shifting, we generate a small local random offset field at all pixel locations, and apply bilinear warping [12, 44]. The strength of noise is decayed overtime. In this manner, we expect $\mathcal{S}$ to focus on high-level coarse structures and neglect other details at first to avoid bad minima. We summarize the training algorithm in Alg. 1.

Algorithm 1 Training algorithm of MS-ACM for stereoscopic view synthesis.

Require: training set: left views $\mathbb{X}$, and right views $\mathbb{Y}_g$

repeat

1. Sample a batch $\left\{\mathbf{x}^{(i)}\right\}_{i=1}^{m} \in \mathbb{X}$, $\left\{\mathbf{y}_g^{(i)}\right\}_{i=1}^{m} \in \mathbb{Y}_g$;
2. Get predictions $\mathbf{y}^{(i)} = \mathcal{P}\left(\mathbf{x}^{(i)}; \mathbf{w}_{\mathcal{P}}\right)$, and generate noisy groundtruth $\mathbf{y}_n^{(i)}$, $i \in \{1, 2, \cdots, m\}$;
3. Compute feature correlations $\mathbf{c}^{(i)}$, $\mathbf{c}_g^{(i)}$, $\mathbf{c}_n^{(i)}$ by (1);
4. Update $\mathcal{S}$, $\mathcal{R}$ by ascending their gradients: $\nabla_{\mathbf{w}_{\mathcal{S}}, \mathbf{w}_{\mathcal{R}}} \frac{1}{m} \sum_{i=1}^{m} L_{\mathcal{C}}\left(\mathbf{y}^{(i)}, \mathbf{y}_g^{(i)}, \mathbf{y}_n^{(i)}\right)$;
5. Update $\mathcal{P}$ by descending its gradients: $\nabla_{\mathbf{w}_{\mathcal{P}}} \frac{1}{m} \sum_{i=1}^{m} \left(\left\|\mathbf{y}^{(i)} - \mathbf{y}_g^{(i)}\right\| + L_{\mathcal{P}}\left(\mathbf{y}^{(i)}, \mathbf{y}_g^{(i)}\right)\right)$;
6. (Optionally) decay learning rate and noise;

until maximum training iteration is reached.

4. Experiments

4.1. Experimental Settings

Dataset and evaluation metrics. To benchmark existing approaches for stereoscopic view synthesis, we set up experiments on the challenging KITTI dataset [8]. The raw-form KITTI contains a total of 42382 rectified stereo pairs captured from 61 scenes. We benchmark models on the 400 pairs provided as the official training set in KITTI's 2015 challenge. These images span across 28 scenes, which are excluded and the rest 33 ones are kept for training, resulting into 34071 training pairs in total. The Eigen split [6] is also included in evaluation. It provides a test split covering 697 pairs from 29 scenes, and suggests training with the 23488 pairs sampled from the rest 32 scenes. Across this section, these two splits will be referred to KITTI-Raw and KITTI-Eigen, respectively.

We follow previous works on view synthesis [15, 42] and adopt Root Mean Square Deviation (RMSE), Peak Signal-to-noise Ratio (PSNR) and Structure Similarity Index (SSIM) as evaluation metrics. As this work aims to improve the quality of structures, we also perform evaluations in gradient space. Specifically, the metrics *Grad. x* and *Grad. y* measure the mean squared errors between the gradients of the synthesized and groundtruth images in horizontal and vertical directions, respectively.

Baselines. We integrate MSACM into two recent representative architectures, Deep3D [39] and SepConv [22]. SepConv is originally designed for video frame interpolation, which requires two frames as input. We tailor it for stereoscopic view synthesis by removing one image input and keeping other layers fixed. We choose these two baselines for their concise designs and strong performance. However, it should be noted that the proposed approach is general and not restricted to certain architectures.

Besides Deep3D and SepConv, we also compare with LRDepth [9]. All these approaches do not assume additional inputs such as scene depths or multi-view images, thus are directly comparable. For LRDepth, we make use of the models released by the authors. As Deep3D and SepConv do not report results on KITTI or release the training scripts, we retrain them by integrating the authors' source codes into our training framework, as described as follows. We ensure that our integrations keep their original details of model definition that can reproduce their released results.

Implementation details. During training, the high-resolution KITTI images are firstly downsampled by half at resolution 188×621. Patches of size 128×256 are randomly cropped on the downsampled images, which form mini-batches of 8 images. We apply Adam optimizer with the first and second moment decay equal 0.5 and 0.999, respectively. Training lasts for 50 epochs, with a learning rate 10^{-4} that is exponentially decayed by half every 20 epochs. In training MS-ACM, noise is decayed every epoch with exponential factor 0.95. During testing, the image is downsampled to a size 188×621, on which a 160×608 region is cropped from the top-left corner, to meet the aspect ratio requirement of baselines.

Throughout the evaluations, the weights λ_r and λ_n in (6) are set to 10, while the window size for computing correlations is set to 3, if not specifically explained.

4.2. Comparisons with Existing Approaches

Benchmarking results on KITTI. The results are summarized in Table 1. Besides the baselines trained with the ℓ_1 pixel reconstruction loss, we also compare with a variant trained with multi-scale SSIM, an extensively adopted structure-aware loss. As the table shows, the proposed ap-

Table 1. Benchmarking results on the KITTI-Raw (left) and KITTI-Eigen (right) datasets. Arrow ↑ (↓) denotes the larger (smaller) number, the better results. Bold highlights the top place while underline the second.

Models	RMSE ↓	PSNR ↑	Grad. x ↓	Grad. y ↓	SSIM ↑	RMSE ↓	PSNR ↑	Grad. x ↓	Grad. y ↓	SSIM ↑
LRDepth	28.052	19.590	205.124	131.621	0.751	29.868	19.103	203.210	138.895	0.737
Deep3D	19.466	22.854	137.803	81.960	0.829	22.694	21.400	162.112	111.935	0.775
+MS-SSIM	19.520	22.790	135.494	82.256	0.833	23.017	21.295	**156.849**	**110.052**	0.782
+MS-ACM	**18.062**	**23.577**	**120.626**	**75.248**	**0.844**	**22.159**	**21.624**	158.053	110.584	**0.787**
SepConv	19.556	22.861	141.467	83.520	0.827	23.796	21.010	174.754	119.061	0.764
+MS-SSIM	19.825	22.709	142.557	93.204	0.832	23.801	20.987	171.366	119.858	0.766
+MS-ACM	18.370	23.467	128.214	79.415	0.835	23.519	21.120	170.658	119.543	0.768

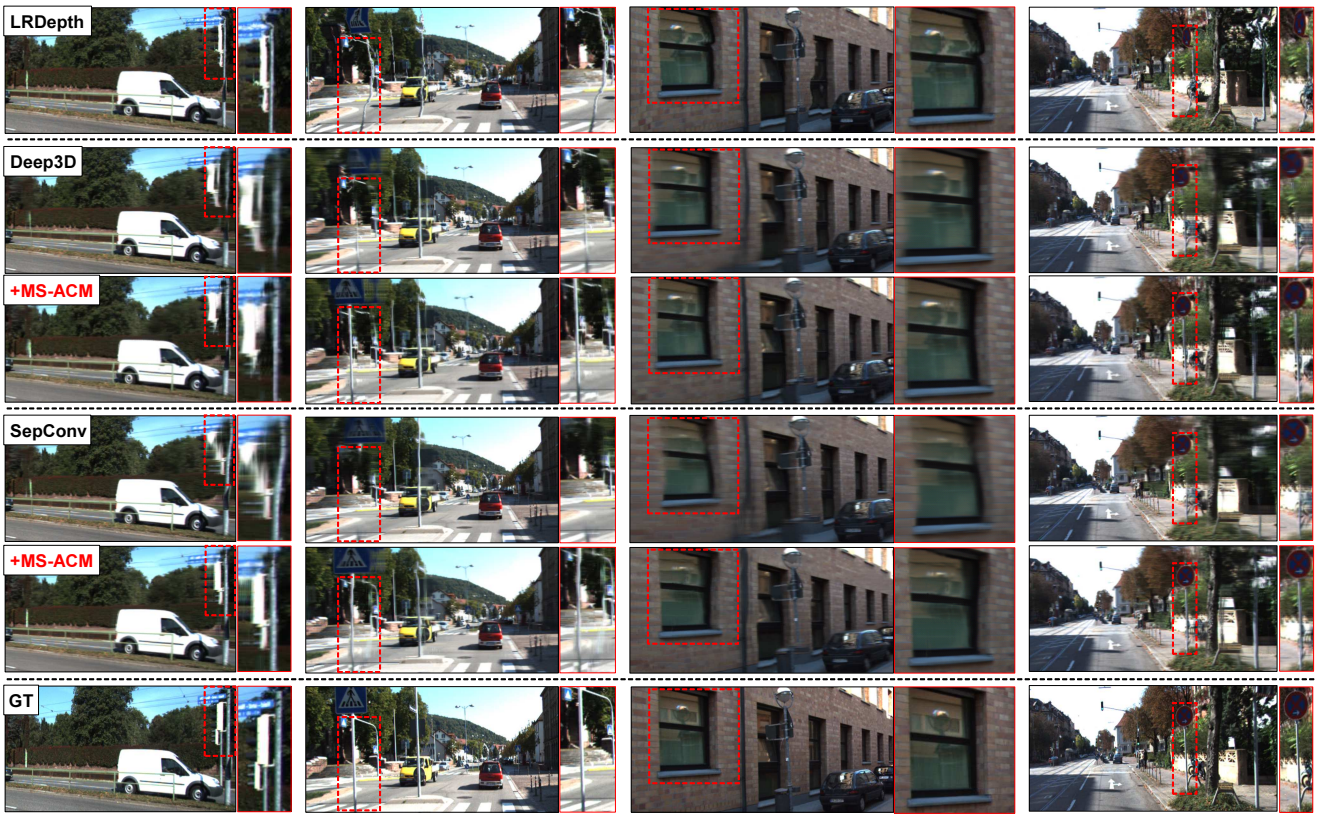

Figure 3. Qualitative results on the KITTI dataset. In each example, red rectangle marks the regions for comparison.

proach improves over baseline approaches consistently on nearly all the metrics. On the KITTI-raw dataset, a large improvement is achieved on the gradient-specific measures, illustrating that the proposed approach makes model training sensitive to scene boundaries.

Besides result comparisons, Table 1 also suggests several observations that worth to discuss. First, although MS-ACM does not apply SSIM as a training loss, it achieves better SSIM numbers even than training directly with SSIM. It seems strange at the first glance, as the model should devote its capacity to optimizing this specific metric and it indeed gets a lower SSIM loss during training. We attribute this improvement to the stronger generalization ability of MS-ACM, which leads to better testing behavior. In the next subsection, we further demonstrate this point.

Second, although the proposed approach still achieves the best results on KITTI-Eigen, the gap is closer than that on KITTI-Raw. We suspect that it is caused by the bias of dataset sampling. As the distributions of training and testing data of KITTI-Raw are more different (the sites where the data are captured do not overlap), it requires the model to have a better generalization ability. For KITTI-Eigen, on the contrary, training and testing distributions overlap much and the improvement is relatively small.

Qualitative results. In Fig. 3, we show representative results generated by different approaches. With adversarial training, MS-ACM pays attention to any noticeable structural differences. As one can see, it preserves object shapes better, recovers over-smoothed details and successfully handles deformation caused by occlusions. In con-

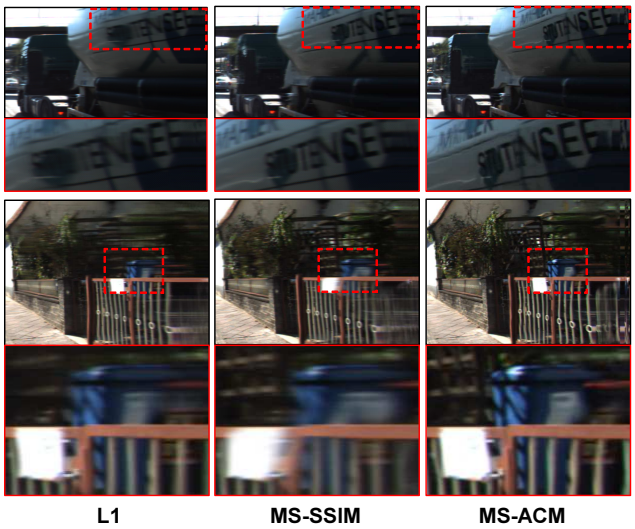

Figure 4. Visual comparisons between MS-ACM and MS-SSIM. See text for details.

Table 2. Analyzing different window parameters on KITTI-Raw dataset. Arrow ↑ (↓) denotes the larger (smaller) number, the better results. Bold highlights the top place while underline the second.

Multi-Scale?	Win. Size	RMSE ↓	PSNR ↑	SSIM ↑
✗	3	20.870	22.257	0.813
✗	7	22.124	21.660	0.773
✗	11	20.393	22.470	0.802
✓	3	**18.370**	**23.467**	**0.835**
✓	7	<u>18.500</u>	<u>23.371</u>	<u>0.829</u>
✓	11	18.848	23.167	0.826

trary, the baselines either sacrifice the small and thin details to achieve a better average quality (*e.g*. Deep3D and SepConv), or exhibit large distortions due to the errors in disparity estimation (*e.g*. LRDepth).

Comparisons with SSIM criterion. SSIM is a differentiable structure-aware criterion, thus is widely adopted for training. Essentially, SSIM optimizes the consistency of first and second-order moments within multi-scale local windows between the predicted and groundtruth images. Such statistical matching, however, renders it not sensitive to local deformations and small details [29]. As shown in Fig. 4, although SSIM fixes coarse structural mistakes but leaves the fine-grained errors unaddressed. As a result, blurred boundaries and over-smoothed details still happen. MS-ACM, on the contrary, does not have such limitation.

Visualization of disparities. The Deep3D or SepConv architectures estimate for each output pixel the likelihoods that it equals to the input pixels at several fixed horizontal offsets. The disparities could be thus produced by aggregating the offsets weighted by the learned likelihoods, which we show in Fig. 5. As one can see, the disparities trained with SSIM are more visually smooth, but not accurate along object boundaries. In contrary, for MS-ACM disparities are

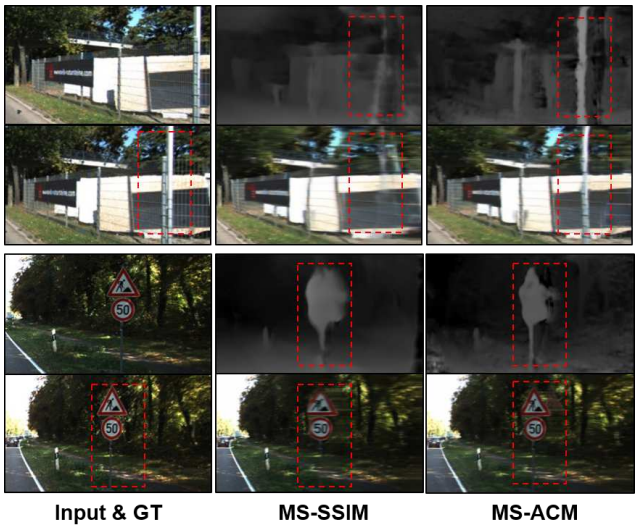

Figure 5. Comparing the learned disparities. For each example, we show disparities and the synthesized views trained with MS-SSIM and MS-ACM, respectively.

Table 3. Parameter study on λ_n and λ_r.

λ_n/λ_r	0.1/0.1	0.1/1	1/0.1	1/10	10/1	10/10
PSNR	22.96	23.06	22.97	23.59	23.62	23.92
SSIM	0.83	0.83	0.84	0.84	0.84	0.85

adapted to scene edges and exhibits sharp depth boundaries. However, in textureless regions (*e.g*. road), they are not that accurate and smooth. Adding smoothness constraint solves this problem, but is not desired for view synthesis as it may smooth out object boundaries and cause distortions.

4.3. Performance Analysis

In this section, we conduct extensive experiments to see how the proposed approach works under various situations. All the experiments are based on the SepConv baseline.

Parameter analysis. At first, we study how different window sizes impact the proposed approach. We also consider a single-scale variant, where only the deepest scale is involved for structure matching. From the results in Table 2, we conclude that multi-scale matching is consistently beneficial, as learning different feature scales enables both local and global structural mistakes to be fixed. However, larger window sizes do not necessarily help improve the results. We suspect that as deep representations already capture sufficient local context, a small window would suffice.

In Table 3, we evaluate different combinations of parameters λ_n and λ_r in Eqn. (6). We find that they both improve they both improve results as a stable behaviour: as long as they are large enough (*i.e*. $\lambda_r, \lambda_n \geq 1$), the final results are not very sensitive to them.

Ablation study of design choices. In the second experiment, we show empirically the necessity of several impor-

Table 4. Ablation study of design choices on the KITTI 2015 split. Arrow ↑ (↓) denotes the larger (smaller) number, the better results. Bold highlights the top place while underline the second.

Loss	Noise?	Feat. Reg.?	Self Recon.?	RMSE ↓	PSNR ↑	Grad. x ↓	Grad. y ↓	SSIM ↑
Corr-ℓ_1	✗	✗	✗	44.662	15.272	386.909	338.504	0.491
Corr-ℓ_1	✓	✗	✗	19.558	22.841	141.227	87.518	0.819
Corr-ℓ_1	✓	✓	✗	19.280	22.961	137.666	86.353	<u>0.825</u>
Corr-ℓ_1	✓	✓	✓	**18.370**	**23.461**	**128.214**	**79.415**	**0.835**
ℓ_1	✓	✓	✓	<u>18.921</u>	<u>23.111</u>	<u>132.578</u>	<u>85.043</u>	0.819

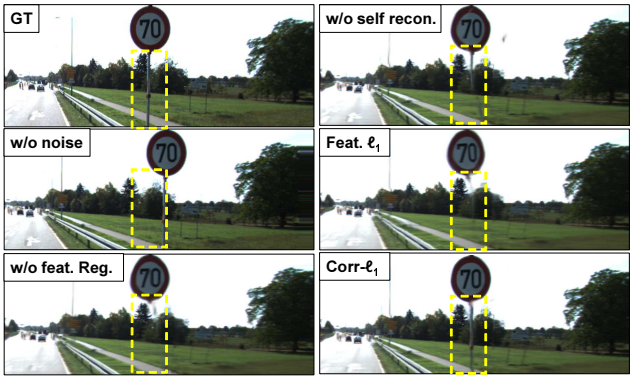

Figure 6. Studying different components of the proposed approach by visual comparisons. See text for details.

tant design choices. The numbers are reported in Table 4, and a visual comparison is provided in Fig. 6. Without enforcing noise resistance (w/o noise), the model simply does not learn much. The structure critic network notices the inherent distribution differences between the synthesized and real input, thus the view synthesis network tends to copy the input to make them look real. After adding noise (w/o feat. reg.), trainings succeeds, but details are missing. Feature regularization (w/o self recon.) improves the details, but does not address overall distortion. Incorporating self-reconstruction (corr-ℓ_1) helps a lot by learning features tightly correlated with the spatial context of the scene.

We also replace the corr-ℓ_1 loss with the standard feature ℓ_1 loss for adversarial training, and it gets worse performance. We believe that explicit modeling of structures in MS-ACM eases the difficulty of encoding them with feature learning. As shown in Fig. 6, ℓ_1 loss does not learn the thin structure although equipped with the same other strategies.

Generalizability to unseen dataset. As mentioned previously, we believe that an advantage of MS-ACM is its better generalizability over classic metrics. The intuition is that adversarial training provides easy-to-hard dynamic training signals, which may prevent the model from continuously optimizing a fixed objective and getting overfitting. To illustrate this point, we evaluate the model trained on KITTI-raw dataset to the test set of Cityscapes benchmark [4], without further finetuning. The input image is resized to resolution 192×384, which matches the scale of the trained model. In Table 5, it shows that while MS-SSIM does not apparently improves over the baseline, MS-ACM significantly boosts

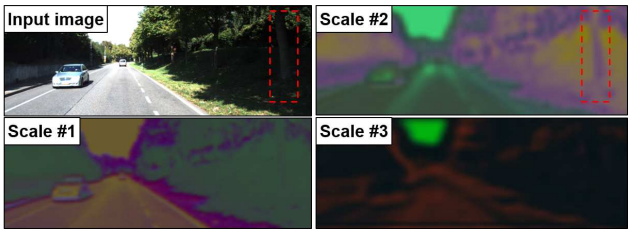

Figure 7. The features learned by the structure critic network, visualized by PCA projection.

Table 5. Model generalizability on the Cityscapes test set. Arrow ↑ (↓) denotes the larger (smaller) number, the better results. Bold highlights the top place while underline the second.

Models	SepConv	+MS-SSIM	+MS-ACM
RMSE ↓	<u>19.547</u>	19.586	**17.731**
PSNR ↑	<u>22.620</u>	22.603	**23.465**
SSIM ↑	0.650	<u>0.661</u>	**0.693**

the performance in nearly all metrics.

Visualization of learned features. Finally, we visualize the learned features in the structure critic network by PCA projection, and show them in Fig. 7. As expected, the first scale learns local edges to represent fine-level information. From the second scale, the model seems to filter out low-level colors and emphasize more on region shapes (see the marked regions). The third scale, as it shows, captures more complex structural patterns that the model finds best to represent the global layout of the scene.

5. Conclusion

This paper proposes Multi-Scale Adversarial Correlation Matching for stereoscopic view synthesis. MS-ACM transforms the synthesized results and groundtruths into multi-scale feature spaces, in which feature correlations are computed as structural representation. By adversarial training on the distances of such representations, errors of different scales are discovered and reduced, enabling structure preservation at various granularities.

In the future work, we are interested in introducing high-level cues (*e.g.* semantics, object contours) to incorporate scene-level knowledge for better structure learning.

References

[1] S. Baker, R. Szeliski, and P. Anandan. A layered approach to stereo reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1998.

[2] C. Kaae Sønderby, J. Caballero, L. Theis, W. Shi, and F. Huszár. Amortised MAP Inference for Image Super-resolution. *ArXiv 1610.04490 [cs.CV]*, 2016.

[3] C.-H. Chu. Video stabilization for stereoscopic 3d on 3d mobile devices. In *IEEE International Conference on Multimedia and Expo (ICME)*, 2014.

[4] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[5] A. Criminisi, A. Blake, C. Rother, J. Shotton, and P. H. S. Torr. Efficient dense stereo with occlusions for new view-synthesis by four-state dynamic programming. *International Journal of Computer Vision (IJCV)*, 71(1):89–110, 2007.

[6] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in Neural Information Processing Systems (NIPS)*, 2014.

[7] J. Flynn, I. Neulander, J. Philbin, and N. Snavely. Deepstereo: Learning to predict new views from the world's imagery. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[8] A. Geiger, P. Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013.

[9] C. Godard, O. Mac Aodha, and G. J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[10] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems (NIPS)*, 2014.

[11] J.-J. Hwang, T.-W. Ke, J. Shi, and S. X. Yu. Adversarial Structure Matching Loss for Image Segmentation. *ArXiv 1805.07457 [cs.CV]*, 2018.

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

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

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

[15] M. Liu, X. He, and M. Salzmann. Geometry-aware deep network for single-image novel view synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[16] C. Luo, J. Zhan, L. Wang, and Q. Yang. Cosine Normalization: Using Cosine Similarity Instead of Dot Product in Neural Networks. *ArXiv 1702.05870 [cs.ML]*, 2017.

[17] G. Luo, Y. Zhu, Z. Li, and L. Zhang. A hole filling approach based on background reconstruction for view synthesis in 3d video. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[18] G. Máttyus and R. Urtasun. Matching adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[19] L. McMillan and G. Bishop. Plenoptic modeling: an image-based rendering system. In *Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, 1995.

[20] S. Niklaus and F. Liu. Context-aware synthesis for video frame interpolation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[21] S. Niklaus, L. Mai, and F. Liu. Video frame interpolation via adaptive separable convolution. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[22] S. Niklaus, L. Mai, and F. Liu. Video frame interpolation via adaptive separable convolution. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[23] E. Park, J. Yang, E. Yumer, D. Ceylan, and A. C. Berg. Transformation-grounded image generation network for novel 3d view synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[24] E. Penner and L. Zhang. Soft 3d reconstruction for view synthesis. *ACM Transactions on Graphics (TOG)*, 36(6):235:1–235:11, 2017.

[25] S. Pujades, F. Devernay, and B. Goldluecke. Bayesian view synthesis and image-based rendering principles. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[26] A. Radford, L. Metz, and S. Chintala. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. *ArXiv 1511.06434 [cs.ML]*, 2015.

[27] K. Rematas, C. H. Nguyen, T. Ritschel, M. Fritz, and T. Tuytelaars. Novel views of objects from a single image. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 39(8):1576–1590, 2017.

[28] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2015.

[29] M. P. Sampat, Z. Wang, S. Gupta, A. C. Bovik, and M. K. Markey. Complex wavelet structural similarity: A new image similarity index. *IEEE Transactions on Image Processing (TIP)*, 18(11):2385–2401, 2009.

[30] D. Scharstein. Stereo vision for view synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1996.

[31] E. Shechtman and M. Irani. Matching local self-similarities across images and videos. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[32] H. Su, F. Wang, E. Yi, and L. J. Guibas. 3d-assisted feature synthesis for novel views of an object. In *IEEE International Conference on Computer Vision (ICCV)*, 2015.

[33] S.-H. Sun, M. Huh, Y.-H. Liao, N. Zhang, and J. J. Lim. Multi-view to novel view: Synthesizing views with self-learned confidence. In *European Conference on Computer Vision (ECCV)*, 2018.

[34] S. Tulsiani, R. Tucker, and N. Snavely. Layer-structured 3d scene inference via view synthesis. In *European Conference on Computer Vision (ECCV)*, 2018.

[35] G. Vogiatzis, P. H. S. Torr, and R. Cipolla. Multi-view stereo via volumetric graph-cuts. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[36] C. Vondrick, A. Shrivastava, A. Fathi, S. Guadarrama, and K. Murphy. Tracking emerges by colorizing videos. In *European Conference on Computer Vision (ECCV)*, 2018.

[37] O. J. Woodford, I. D. Reid, and A. W. Fitzgibbon. Efficient new-view synthesis using pairwise dictionary priors. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[38] O. J. Woodford, I. D. Reid, P. H. S. Torr, and A. W. Fitzgibbon. On new view synthesis using multiview stereo. In *British Machine Vision Conference (BMVC)*, 2007.

[39] J. Xie, R. B. Girshick, and A. Farhadi. Deep3d: Fully automatic 2d-to-3d video conversion with deep convolutional neural networks. In *European Conference on Computer Vision (ECCV)*, 2016.

[40] Y. Xue, T. Xu, H. Zhang, L. R. Long, and X. Huang. Segan: Adversarial network with multi-scale l1 loss for medical image segmentation. *Neuroinformatics*, 16(3):383–392, 2018.

[41] J. Yang, S. E. Reed, M.-H. Yang, and H. Lee. Weakly-supervised disentangling with recurrent transformations for 3d view synthesis. In *Advances in Neural Information Processing Systems (NIPS)*, 2015.

[42] T. Zhou, S. Tulsiani, W. Sun, J. Malik, and A. A. Efros. View synthesis by appearance flow. In *European Conference on Computer Vision (ECCV)*, 2016.

[43] H. Zhu, H. Su, P. Wang, X. Cao, and R. Yang. View extrapolation of human body from a single image. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[44] X. Zhu, Y. Xiong, J. Dai, L. Yuan, and Y. Wei. Deep feature flow for video recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.