Future Video Synthesis with Object Motion Prediction

Yue Wu HKUST
Rongrong Gao HKUST
Jaesik Park POSTECH
Qifeng Chen HKUST

Abstract

We present an approach to predict future video frames given a sequence of continuous video frames in the past. Instead of synthesizing images directly, our approach is designed to understand the complex scene dynamics by decoupling the background scene and moving objects. The appearance of the scene components in the future is predicted by non-rigid deformation of the background and affine transformation of moving objects. The anticipated appearances are combined to create a reasonable video in the future. With this procedure, our method exhibits much less tearing or distortion artifact compared to other approaches. Experimental results on the Cityscapes and KITTI datasets show that our model outperforms the state-of-the-art in terms of visual quality and accuracy.

1. Introduction

Can an artificial intelligence system predict a photorealistic video conditioned on past visual observation? With an accurate video prediction model, an intelligent agent can plan its motion according to the predicted video. Future video generation techniques can also be used to synthesize a long video by repeatedly extending the future of the video. Video prediction has been adopted in various applications such as sensorimotor control, autonomous driving, and video analysis [9, 35, 25, 39].

Video prediction has not been solved yet, especially if we need to synthesize frames of an extended period. Existing methods tend to generate blurry and distorted images where rigid objects are usually bent and spread. This issue indicates that it is necessary to consider several aspects: forecasting the motion of dynamic objects, creating new visual data for unveiled regions, finding spatio-temporal relationships when two objects overlap, and so on. Therefore, to generate realistic future video, understanding essential information such as semantics, shape, or dynamics of the scene is necessary.

Most existing methods tackle the video prediction task by generating future video frames one by one in an unsupervised fashion [24, 50, 41, 6]. These approaches synthesize future frames at the pixel level without explicit modeling of the motions or semantics of the scene. Thus, it is difficult for the model to grasp the concept of object boundaries to create different movements for different objects. For instance, a moving car should be treated individually instead of modeling a car and background scene as a whole. Recently, Wang et al. [44] propose a general video-to-video translation model (vid2vid) that demonstrates future video prediction as a sub-task. The model takes semantic maps in the past and estimates future semantic maps to synthesize the next video frame. With this idea, the generated video can preserve the structure of objects better, but the shape of objects deforms unnaturally in the long term. To synthesize more realistic future videos, we find that the explicit modeling of object trajectories is highly beneficial.

The key idea of our video prediction model is that we synthesize future video frames conditioned on predicted object trajectories. The trajectory of an object is defined as its 2D pixel location in each video frame. In particular, we identify each dynamic object and predict its moving path, scale change, and shape in the future. Object appearance in the next few frames can be roughly approximated by applying an affine transformation on the object segment in the last input frame. In this way, appearance is highly regularized and avoids unexpected deformation. For the background with static objects, we directly predict a motion field between the last frame and each future frame. Then we warp the background image with the estimated motion field. In this background image in the future, dynamic objects are located. Since the future background images may contain missing regions due to occlusion, we apply refinement steps to complete missing areas and harmonize components. Our experiments indicate that our approach can synthesize future videos that are more photo-realistic than state-of-the-art video prediction methods.

2. Related Work

Future frame synthesis is initially studied at the patch level [37]. Recent advances in the future prediction from image sequence can be classified into the three-fold.

Single image prediction. This class of works synthesizes a single frame for the next time step. Patraucean *et al*. [30] use a convolutional version of long short-term memory. Lot-

Figure 1. Results of predicting the frames $t+1$, $t+5$, and $t+10$ on the Cityscapes dataset [5].

ter *et al*. [24] introduce a predictive coding network, and Byeon *et al*. [3] improve image quality using parallel multi-dimensional long short-term memory (LSTM). Liang *et al*. [21] design the generative adversarial loss [12] for both on predicted optical flow and synthesized image to enforce consistency explicitly. Liu *et al*. [22] introduce an efficient atomic operator to predict the next frame in an unsupervised manner.

Long-term prediction. More recently, long-term video prediction becomes an active research area. Srivastava *et al*. [35] use LSTM to encode and decode video sequence. Denton and Fergus *et al*. [6] introduce an approach that produces plausible frame predictions with stochastic latent variables to generate sharp frames. Mathieu *et al*. [29] propose a multi-scale approach by reducing blur artifact with the aid of mean squared error loss. Lee *et al*. [19] combine the latent variable for stochastic reasoning and adversarial loss for photo-realistic image synthesis. Wichers *et al*. [47] introduce a hierarchical approach without using ground truth annotation of high-level structures. A probabilistic approach by Xue *et al*. [50] synthesizes various motions from a single image. Villegas *et al*. [40] and Reda *et al*. [32] involve motion encoder to explicitly regard foreground motion. Wang *et al*. [44] propose an advanced framework that can synthesize long-term video. The power of this approach comes from a concrete design of generative adversarial loss for image domain and temporal domain. Ye *et al*. [52] proposed a pixel-level future prediction approach given a single image with the prediction of future states of independent entities. Liu *et al*. [4] used variational recurrent neural networks with higher capacity likelihood models. Hang *et al*. [10] introduced a confidence-aware warping operator to predict occluded area and disoccluded area separately. Ho *et al*. [15] proposed a parametric video prediction approach based on a sparse motion field.

Other tasks. In addition to the next image or long-term image synthesis, additional tasks, including scene semantics or motion of dynamic objects, have been studied. A method by Walker *et al*. [43] predicts the movement of the foreground object from a single image, and recent works [1, 2, 13, 18, 27, 51, 8, 33] demonstrate that human movements or trajectories can be estimated successfully from the real dataset. Vondrick *et al*. [42] synthesize a one-second video from weakly annotated natural videos using a network understanding dynamics of foreground object and motion classes. Jin *et al*. [17] propose a new fully convolutional network for predicting semantic label and optical flow for a next frame.

Our approach is in line with recent works [7, 40, 26, 31] that decouple stationary and moving part of the scene. We incorporate high-level semantics and instances to consider movements of individual foreground objects explicitly. As shown in the paper, the synthesized frames are more realistic than previous state-of-the-art on complex scenes, such as real-world driving videos.

3. Model

Problem definition. Let x_i be the video frame at time step i, s_i and e_i be the corresponding semantic and instance map of x_i, and f_i be the motion field (or optical flow) from frame x_i to frame x_{i+1}. Then our video prediction task could be formulated as follows. Given input video frames x_i, semantic maps s_i, instance maps e_i from time-steps $i = \{1, \cdots, t\}$ and optical flows between consecutive frames, predict the future video frames x_i for $i = \{t+1, \cdots, T\}$. t is the index to the last input frames, and T is the index of the last prediction frame. To solve this problem, we propose a separate-predict-composite approach to produce realistic future frames.

Overview. To begin with, we attempt to classify objects into dynamic and static ones to trace and handle various motions in the scene effectively. We train a moving object detection network to classify moving objects and static scenes. This idea is different from previous approaches that divide frames to the foreground and background region based on semantic class. After obtaining dynamic and static regions, we predict the optical flow of the static scene and warp the last input video frame to get future frames of the static scene. Then, we use a background-aware spatial transformation network (STN) to predict the motion of dynamic objects. The holes in warped static scenes are filled using an image inpainting method [54]. The warped images serve as the future background information for the STN network. The estimated static and dynamic scene is composed in the last stage to generate a seamless image. Fig. 2 illustrates the proposed pipeline.

3.1. Moving object detection

There are two major causes for the appearance change between continuous frames. The first one is the dynamic motion of moving objects, and the second one is the ego-motion of the camera. To handle such a scene effectively, we train a moving object detection network to identify moving objects and static scenes. Based on Cityscapes-Motion dataset and KITTI-Motion dataset [38] that provide annotations of moving areas, we build an encoder-decoder architecture to detect moving regions, with ResNet50 as the backbone [14]. The input of the network is observed sequences of frames, semantic maps, instance maps, and optical flow between consecutive frames. The output of the network is a binary mask to indicate the region of the moving object.

3.2. Background prediction

With the identified moving objects, the pipeline handles the static motion of the scene that is predicted by an optical flow network. The network predicts the forward and backward optical flow between the last observed frame x_t and each future frame[1]. Note that our pipeline does not predict frame-wise motion recursively. Instead, the batch prediction of flow maps alleviates the effect of accumulated error and possible blur artifacts.

Generative model. We propose a conditional generative adversarial network to predict the future optical flow. The pipeline has one generator $\mathcal{G}_{back}$ and two types of discriminators, one for evaluating single frame $\mathcal{D}_f$ and the other for temporal coherence of multiple video frames $\mathcal{D}_v$. The generator $\mathcal{G}_{back}$ is an encoder-decoder structure with skip connections. The encoder follows the structure of ResNet50 [14] with activation function replaced with Leaky ReLU [28]. The input of encoder is a tensor that collates sequential input images $\{x_i\}_{i=1}^{t}$, sequential semantic layout $\{s_i\}_{i=1}^{t}$ generated by [57], sequential instance maps $\{e_i\}_{i=1}^{t}$ computed by [48], and sequential optical flow between consecutive input frames $\{f_i\}_{i=1}^{t-1}$ using PWC-Net [36].

The decoder consists of several upsample modules. We employ the multi-scale strategy to predict optical flow at different spatial resolutions. The input to each module is a concatenation of feature maps produced at the corresponding resolution by the encoder, feature maps provided by the preceding module, and optical flow prediction result. Each upsample module consist of a bilinear upsample layer and a convolutional layer to recover the spatial resolution.

A loss function of frame discriminator $\mathcal{L}_f$ checks if estimated flow creates weird artifacts by warping x_t using

[1] Backward optical flow is used for warping x_t to the future because this can avoid warping artifacts.

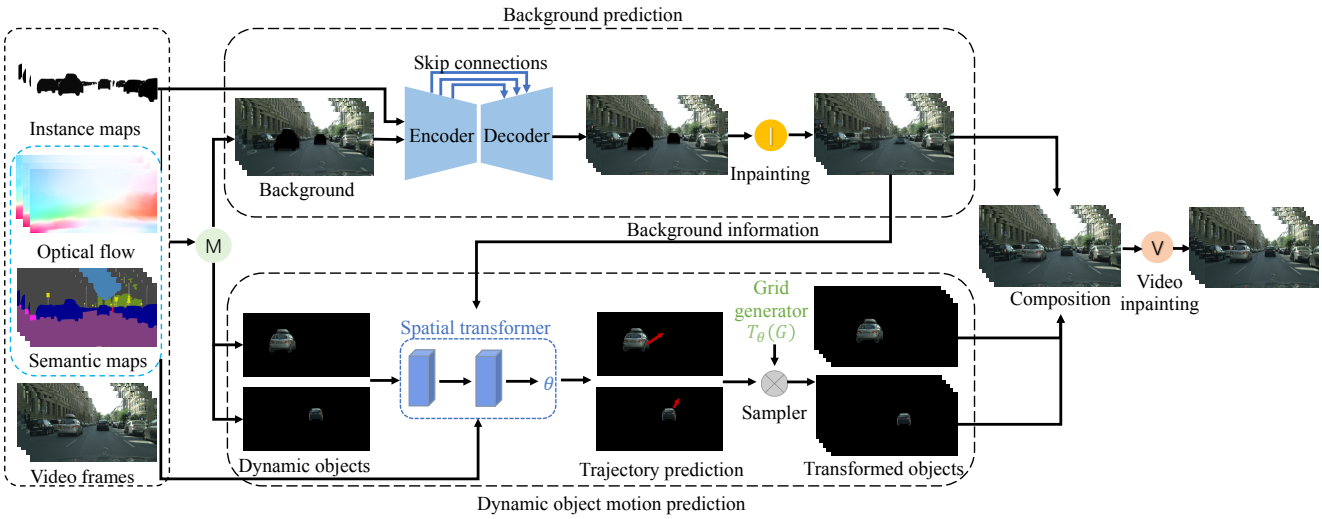

Figure 2. Overview of the proposed architecture. We use a dynamic object detection model M to separate moving objects and static background. The missing foreground area in the generated future background is inpainted using the inpainting model I. By providing the background images for the future, we apply a spatial transformer to predict moving objects. After that, we composite the foreground and background images and use a video inpainting module V to inpaint occluded area.

predicted optical flow:

$$\mathcal{L}_f = \sum_{i=t+1}^{T} \Big(\log \mathcal{D}_f(x_i, f_{i\to t}) + \log(1 - \mathcal{D}_f(\tilde{x}_i, \tilde{f}_{i\to t})) \Big), \tag{1}$$

where $\tilde{f}_{i\to t}$ is the optical flow from frame i to frame t predicted by $\mathcal{G}_{back}$, $\tilde{x}_i$ is inversely warped image of x_t using $\tilde{f}_{i\to t}$, and $f_{i\to t}$ is the ground-truth optical flow from frame i to frame t. The loss $\mathcal{L}_v$ on $\mathcal{D}_v$ is defined as:

$$\begin{aligned} \mathcal{L}_v \;=\; & \log \mathcal{D}_v\big(\{x_i\}_{i=1}^{T}, \{f_{i\to t}\}_{i=t+1}^{T}\big) \;+ \\ & \log\Big(1 - \mathcal{D}_v\big(\{\tilde{x}_i\}_{i=1}^{T}, \{\tilde{f}_{i\to t}\}_{i=t+1}^{T}\big)\Big), \end{aligned} \tag{2}$$

where $\{x_i\}_{i=1}^{T}$ concatenates images $\{x_1, \cdots, x_T\}$ in the channel-wise manner, $\{f_{i\to t}\}_{i=t+1}^{T}$ is concatenated optical flow, and others are defined similarly. In contrast to $\mathcal{L}_f$, this function penalizes unrealistic image and motion by directly analyzing a range of image frames and flow maps. This is realized by concatenating frames to learn temporal changes. In this way, unrealistic temporal behavior is discouraged.

Flow evaluation. We have an additive loss $\mathcal{L}_{flow}$ to evaluate estimated flow. $\mathcal{L}_{flow}$ is linear combination of multiple criterions $\mathcal{L}_{flow} = \sum(\lambda_{data}\mathcal{L}_{data} + \lambda_{perc}\mathcal{L}_{perc} + \lambda_{smooth}\mathcal{L}_{smooth} + \lambda_{cons}\mathcal{L}_{cons})$, where $(\lambda_{data}, \lambda_{perc}, \lambda_{smooth}, \lambda_{cons})$ is empirically set to (1.0, 15.0, 1.0, 1.0), respectively.

$\mathcal{L}_{data}$ is a data term that penalizes the discrepancy between predicted flow and the flow from real images:

$$\mathcal{L}_{data} = \sum_{i=t+1}^{T} C_{i\to t} \left\| \tilde{f}_{i\to t} - f_{i\to t} \right\|_1, \tag{3}$$

where a confidence map C indicates whether the optical flow on this pixel is valid.

We also compute a perceptual loss between warped image and ground truth image. We use VGG19 model [34] for feature extraction and define a L_1 loss between warped images and ground truth images in the feature domain:

$$\mathcal{L}_{perc} = \sum_{i=t+1}^{T} \left(\sum_{j=1}^{n} \frac{1}{N_j} \left\| \Phi_j(\tilde{x}_i) - \Phi_j(x_i) \right\|_1 \right), \tag{4}$$

where n is the number of VGG feature layers. where Φ_j denote feature map from the j-th layer in the VGG-19 network having a number of feature parameter N_j. To make the predicted optical flow coherent with the structure of x_i, we adopt smoothness loss for optical flow weighted by image gradient ∇x_i:

$$\mathcal{L}_{smooth} = \sum_{i=t+1}^{T} \left\| \nabla \tilde{f}_{i\to t} \right\|_1 e^{-\|\nabla x_i\|_1}, \tag{5}$$

where ∇ indicates the gradient operator. To make the training more stable, we use a forward-backward consistency loss [53]:

$$\mathcal{L}_{cons} = \sum_{i=t+1}^{T} \sum_{\mathbf{p}} \delta(\mathbf{p}) \left\| \Delta \tilde{f}_{i\to t}(\mathbf{p}) \right\|_1, \tag{6}$$

where $\Delta \tilde{f}_{i\to t}(\mathbf{p})$ is the discrepancy obtained from forward and backward flow check at pixel location $\mathbf{p}$. It is defined as $\Delta \tilde{f}_{i\to t}(\mathbf{p}) = \mathbf{p} - (\mathbf{p}' + \tilde{f}_{t\to i}(\mathbf{p}'))$, where $\mathbf{p}' = \mathbf{p} + \tilde{f}_{i\to t}(\mathbf{p})$. $\delta(\mathbf{p})$ is a conditional scalar for robustness. $\delta(\mathbf{p})$ is 1 if

$\left\|\Delta\tilde{f}_{i\to t}(\mathbf{p})\right\|_2 < \max\left(a, b\left\|\tilde{f}_{i\to t}(\mathbf{p})\right\|_2\right)$ or 0 otherwise. (a, b) is empirically set to (3, 0.05). Pixels where the forward and backward flows contradict seriously are regarded as possible outliers.

As a result, we train the flow prediction network using a combination of proposed losses[2]:

$$\min_{\mathcal{G}_{back}}\left(\max_{\mathcal{D}_f}\lambda_f\mathcal{L}_f + \max_{\mathcal{D}_v}\lambda_v\mathcal{L}_v + \mathcal{L}_{flow}\right). \tag{7}$$

The weight for frame discriminator λ_f and video discriminator λ_v is empirically set to 1.0 and 2.0. Here we use the multi-scale loss that is defined as the sum of the losses when images are evaluated at different resolutions: full resolution, half resolution, $\frac{1}{4}$ resolution, and so on.

Background inpainting. For better future prediction, we decompose moving objects and static scenes from the input image. After extracting moving objects, the area where moving objects were placed remains blank. Such a blank region is filled with an inpainting network based on Wasserstein GANs with a contextual attention layer [54]. To make the inpainting network even better, we feed randomly cropped patches from background classes (such as buildings, trees, or roads in traffic scenes) and perform fine-tuning. This procedure makes an inpainted background image b_i from the original image with holes.

The background inpainting operation is necessary. It is because the inpainted background is used as the extra guidance for dynamic object trajectories prediction. Without background inpainting, the regions for moving objects are denoted as black. Then the dynamic object trajectories prediction module will overfit to predict motion to match the black pixels, which is not desirable.

3.3. Dynamic object motion prediction

Our approach identifies dynamic objects in the scene and handles their motion explicitly. Instead of treating cluttered scenes as a whole, this scheme helps to understand the history of an individual object so that it can predict the future better. We presume the motion of dynamic objects can be adequately approximated with 2D affine transformation. Due to this rigid motion constraint, predicted appearance does not show distortion or unrealistic texture that are common problems in previous approaches. Our model detects all the moving objects, and each object is treated separately using our transformation network.

Network. The input to the motion prediction network is a sequence of binary object masks m, optical flow f, semantic maps s, objects o, and inpainted background images b. The network produces a series of 2D affine transformation $\mathbf{A}$ that expresses the predicted object motion. Note that the network takes background images as input because the location of objects is highly related to the background. For example, a car should be placed on the road, and trees should not block the road, etc. Without background information, the network may predict unrealistic trajectories because the prediction is purely based on past motions.

[2]We also define the similar losses with opposite flow direction to improve consistency.

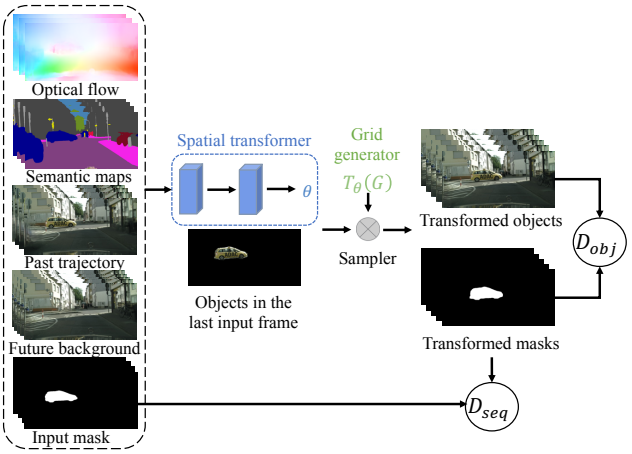

Figure 3. Training loss for dynamic motion prediction. Our approach puts the predicted objects (by spatial transformer) on the predicted background images and generates virtual images. We also use two discriminators to ensure the locations of predicted objects are spatially and temporally coherent.

The network is an encoder architecture and outputs the parameters of a series of 2D affine transformation $\mathbf{A}$. Then the following grid sampler transforms coordinate of object's pixels in the last frame using the estimated parameters. By combining the estimated background image b and transformed object o, we can build a composition image c.

Similar to the background prediction module, the motion prediction network is equipped with two discriminators: single object discriminator $\mathcal{D}_{obj}$ and object sequence discriminator $\mathcal{D}_{seq}$. The input of $\mathcal{D}_{obj}$ is a pair of an object mask and a composed image to determine whether the predicted location is natural. This discriminator is used to suppress some unreasonable areas, such as cars on a building. The produced image is made by placing a transformed object on an inpainted background. The input of $\mathcal{D}_{seq}$ takes a sequence of masks representing the object trajectory as input and determines whether the predicted object trajectory is reasonable.

We define the discriminator loss $\mathcal{L}_{obj}$ on single object discriminator and the discriminator loss $\mathcal{L}_{seq}$ on object sequence discriminator as follows:

$$\mathcal{L}_{obj} = \sum_{i=t+1}^{T}\Big(\log\mathcal{D}_{obj}(c_i, m_i) + \log\big(1 - \mathcal{D}_{obj}(\tilde{c}_i, \tilde{m}_i)\big)\Big), \tag{8}$$

$$\mathcal{L}_{seq} = \log \mathcal{D}_{seq}\big(\{m_i\}_{i=1}^{T}\big) + \log\Big(1 - \mathcal{D}_{seq}\big(\{\tilde{m}_i\}_{i=1}^{T}\big)\Big), \tag{9}$$

where $\mathcal{L}_{obj}$ is the GAN loss on mask and synthetic image pair defined by single object discriminator $\mathcal{D}_{obj}$, $\mathcal{L}_{seq}$ is the GAN loss on sequential masks defined by object sequence discriminator $\mathcal{D}_{seq}$. $\tilde{c}_i$ is the composite of transformed object and background information, and $\tilde{m}_i$ is a binary mask of moving object.

Another loss $\mathcal{L}_r$ consists of three terms, and it is equivalent to $\lambda_{rgb}\mathcal{L}_{rgb} + \lambda_{reg}\mathcal{L}_{reg} + \lambda_{smooth}\mathcal{L}_{smooth}$, where $(\lambda_{rgb}, \lambda_{reg}, \lambda_{smooth})$ is set to (1.0, 1.0, 2.0). $\mathcal{L}_{rgb}$ is the L_1 difference between appearance of a j-th object in i-th frame and its ground truth: $\mathcal{L}_{rgb} = \sum_{i=t+1}^{T} m_{(i,j)} \odot \left\|\tilde{o}_{(i,j)} - o_{(i,j)}\right\|_1$, where $\tilde{o}$ is transformed object. $\mathcal{L}_{smooth}$ is the smoothness loss to improve the temporal coherency of predicted parameters: $\mathcal{L}_{smooth} = \sum_{i=t+3}^{T}\sum_j \left\|\left(\mathbf{A}_{(i,j)} - \mathbf{A}_{(i-1,j)}\right) - \left(\mathbf{A}_{(i-1,j)} - \mathbf{A}_{(i-2,j)}\right)\right\|_1$.

$\mathcal{L}_{reg}$ is a regularization term on predicted parameters to prevent abrupt change from original state, or identity transform $\mathbf{I}$: $\mathcal{L}_{reg} = \sum_{i=t+1}^{T}\sum_j \|\mathbf{A}_{(i,j)} - \mathbf{I}\|_2$.

As a result, for each moving object, we a separate motion estimation network and the loss for training this network is:

$$\min_{\mathcal{G}_{fore}} \big(\max_{\mathcal{D}_{obj}} \lambda_{obj}\mathcal{L}_{obj} + \max_{\mathcal{D}_{seq}} \lambda_{seq}\mathcal{L}_{seq} + \lambda_r\mathcal{L}_r\big), \tag{10}$$

where $(\lambda_{obj}, \lambda_{seq}, \lambda_r)$ is set to $(4.0, 4.0, 1.0)$ and $\mathcal{G}_{fore}$ is the foreground object generator.

Training data generation. The Cityscapes dataset [5] and KITTI dataset [11] does not provide tracking information for each instance. Therefore, we employ a tracking algorithm to produce data for training the proposed network. We first generate an instance mask using the approach by Xiong *et al*. [48]. Then, few-shot tracking algorithm [20] is employed to obtain bounding boxes of the tracked objects in a video sequence. After getting the bounding boxes of the tracked objects, we compute the intersection of bounding boxes and instance maps to obtain the corresponding binary masks. We employ several strategies to delete some failure tracking samples. For instance, we compute the SSIM [45] score of objects being tracked to determine whether they are the same object.

3.4. Background-foreground composition

After predicting motion for the background scene and moving objects, the composition module fuses the scene components to create future video frames. We determine the relative depth order of moving objects according to the relative depth obtained by GeoNet [53]. Then we place the moving objects one by one onto the predicted background. Note that we have a hole-filled background image b_i, we may directly use those frames for producing output, but it does not have temporal coherence.

Therefore, we adopt a video inpainting approach to minimize flickering artifact. Following the method [49], we utilize forward and backward optical flow between consecutive frames, and employ a consistency check to find valid optical flow. With adequate optical flow, we build a connection between pixels across continuous frames. Pixels with the valid flow are propagated bidirectionally to fill the missing regions. This procedure repeats to minimize holes in the video. If there are still missing regions, the image inpainting method [54] is employed to fill such areas.

4. Experiments

We conduct both quantitative and qualitative experiments on real-world datasets concerning the capability of predicting future video. We compare our approach with other approaches that produce the next-frame or multiple-frames for the future.

4.1. Datasets

We conducted our experiments on Cityscapes dataset [5] and KITTI dataset [11]. Cityscapes dataset contains 2048×1024 resolution image sequences for city scene captured at 17 FPS. For the fair comparison with other approaches that do not produce such resolution, we experiment at the 1024×512 resolution. KITTI dataset contains 375×1242 resolution image sequences for driving scenes captured at 10 FPS. The semantic maps are generated using the method of [57]. For the experiment, we get instance maps using UPSNet [48] and obtain optical flow fields with PWCNet [36]. For the fair comparison, we experiment at the 256×832 resolution. We apply techniques such as random horizontal flipping to augment data.

Cityscapes dataset contains 2975 video sequences for training and 500 video sequences for testing. KITTI dataset for our training and evaluation includes 28 video sequences. We randomly select four sequences for assessment.

4.2. Implementation

We use the multi-scale PatchGAN discriminator [16] architecture for all the discriminators in our framework. For the Cityscapes dataset, the input frame length is set to 4, and the prediction length is set to 5. We first train a model at the 256×512 resolution, then train a 512×1024 resolution model by adding an upsampling module. By recurrently test our model twice, we obtain future predictions for the next 10 frames.

For the KITTI dataset, the input frame length is set to 4. Because the KITTI dataset has a more substantial motion, generating optical flow between two long period frame is difficult using PWCNet [36]. The prediction length for the background prediction model is set to 3. And the prediction length for the dynamic object motion prediction model is set to 5. We experiment at the 256×832 resolution. By

Figure 4. Results of predicting the frames $t+1$, $t+3$, and $t+5$ on the KITTI dataset [11].

	Cityscapes						KITTI					
	Next frame		Next 5 frames		Next 10 frames		Next frame		Next 3 frames		Next 5 frames	
	MS-SSIM	LPIPS	MS-SSIM	LPIPS	MS-SSIM	LPIPS	MS-SSIM	LPIPS	MS-SSIM	LPIPS	MS-SSIM	LPIPS
PredNet [24]	0.8403	0.2599	0.7521	0.3603	0.6633	0.5221	0.5626	0.5535	0.5147	0.5866	0.4756	0.6295
MCNET [40]	**0.8969**	0.1888	0.7058	0.3734	0.5971	0.4513	0.7535	0.2405	0.6352	0.3171	0.5548	0.3739
Voxel Flow [23]	0.8385	0.1737	0.7111	0.2879	0.6341	0.3655	0.5393	0.3247	0.4699	0.3743	0.4262	0.4159
Vid2vid [44]	0.8816	0.1058	0.7513	0.2014	0.6690	0.2705	-	-	-	-	-	-
Ours-WC	0.8792	0.0903	0.7430	0.1718	0.6593	0.2411	0.6853	0.2252	0.5850	0.2897	0.5217	0.3482
Ours-WM	0.8866	0.0899	0.7537	0.1694	0.6727	0.2351	0.7634	0.1987	0.6504	0.2588	0.5839	0.3136
Ours	0.8910	**0.0850**	**0.7568**	**0.1650**	**0.6741**	**0.2328**	**0.7928**	**0.1848**	**0.6765**	**0.2461**	**0.6077**	**0.3049**

Table 1. Comparison with state-of-the-art methods on the Cityscapes and KITTI datasets. The table shows the image quality of the synthesized images. The higher MS-SSIM is better. The lower LPIPS is better.

recurrently test the model twice, we obtain predicted images in the next 5 frames.

All parts of our model are implemented with Pytorch 1.1.0, and we use the ADAM optimizer. For the background prediction model, we train 200 epochs, with learning rate 2e-4 for the first 100 epochs, then linearly decrease the learning rate. For the dynamic trajectory prediction model, we train 60 epochs with a learning rate of 3e-5. Training takes about three days for a 512×1024 resolution model. The experiment is done with Nvidia RTX 2080 Ti.

4.3. Evaluation metrics

We evaluate our model using several metrics measuring the accuracy of video frames in the future. We use a multi-scale structure similarity (MS-SSIM) [46] index and perceptual image patch similarity (LPIPS) [55]. Higher

MS-SSIM scores and lower LPIPS distances suggest better performance.

4.4. Baselines

To evaluate our model for future prediction, we compare our model with the following baselines, where the first several are state-of-the-art approaches, and the rest are variants of our model.

PredNet [24]. PredNet is a prior approach for next-frame prediction. We fine-tune their model on our dataset, and recurrently perform next-frame prediction to get multiple-frame results.

MCNet [40]. This is a state-of-the-art approach for the next frame prediction. We re-train their model on our datasets using their public source code. The multiple frames are generated by recurrently applying the pipeline.

Voxel-Flow [23]. This is a video synthesis approach with optical flow fields across space and time. This approach can be applied for video extrapolation. We re-train their model on our dataset for evaluation.

Vid2vid [44]. This is a video-to-video translation framework. The approach can generate a video conditioned on a sequence of semantic layouts. For future prediction, their approach predicts the semantic layout and converts a sequence of semantic layouts into a real video. We directly compare our method with their provided video prediction results on the Cityscapes dataset.

Ours-WC. Our ablated model without foreground-background composition. To demonstrate the effectiveness of our foreground-background separation approach, we train a model to directly output the optical flow prediction for a full image using the same model with the background prediction.

Ours-WM. Our ablated model without moving object detection. For this model, we remove the moving object detection module and use an STN to predict the trajectories of all possible moving objects (cars, pedestrians) based on semantic classes.

4.5. Evaluation on Cityscapes and KITTI

We evaluate the capability of our model to predict future video frames in both the next-frame and multiple-frames prediction. Our result on Cityscapes and KITTI dataset is shown in Table 1. The frame rates of Cityscapes dataset and KITTI dataset are 17 FPS and 10 FPS, respectively. Then we predict the next 10 frames on the Cityscapes dataset and the next 5 frames on the KITTI dataset, about 0.5 seconds.

On the Cityscapes dataset, in terms of MS-SSIM score, our model achieves comparable scores with MCNet [40] and vid2vid [44]. The performance of our model in LPIPS is 20%, 18%, 14% better than the second-best model for the evaluation of the next frame, next five frames, and the next ten frames. On the KITTI dataset, our model outperforms all state-of-the-art methods in all metrics. Our model's improvement in terms of LPIPS in the next frame, next three frames, next five frames, is 23%, 22%, 18%, respectively against the second-best result. Our improvement in terms of MS-SSIM is 5%, 7%, 10%, respectively. It demonstrates that our method can achieve better performance in both short-term and long term prediction. It is because our approach highly keeps the rigidity of objects. The current state-of-the-art method appears to have significant distortion artifact around object boundary, while our approach alleviates this phenomenon a lot and makes the result more realistic.

We also perform an ablation study on Ours-WC and Ours-WM. From the results, we can see that all the strategies in our model are helpful. The foreground-background decomposition keeps the rigidity of objects and makes the background prediction easier. The moving object detection strategy classifies objects into dynamic or static and predicts separately based on the motion type.

As demonstrated in Fig. 1 and 4, our model produces more realistic results over state-of-the-art methods. Our method keeps the rigidity of objects even in long-term prediction, while the state-of-the-art techniques suffer from distortion around motion boundaries. Also, our method produces a result with less blurriness because we predict the motion of multiple frames together. This strategy alleviates the accumulated error by recurrent prediction. More visual comparisons are shown in the supplement.

4.6. Additional experiments

We also conduct experiments beyond driving scenes on the BAIR robot pushing dataset [9] and the Penn Action dataset [56]. The BAIR dataset consists of videos about a robot arm pushing multiple objects. The Penn dataset has videos with various non-rigid human actions. The results are presented in the supplement.

5. Conclusion

We have presented a separate-predict-composite model for future frame prediction. Our method produces future frames by firstly decomposing possible moving objects into currently-moving or static objects. Then for moving objects, we employ a spatial transformer network to predict the trajectories of objects. This helps to preserve the structure of objects while producing reliable future motion. For background, we use an optical flow prediction network to predict the background of multiple frames at once. Then we integrate the foreground and background and add a video inpainting module to help alleviate the artifact in composition. The experiments have shown that our approach outperforms prior work on future video prediction.

References

[1] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Fei-Fei Li, and Silvio Savarese. Social LSTM: human trajectory prediction in crowded spaces. In *CVPR*, 2016. 3

[2] Apratim Bhattacharyya, Mario Fritz, and Bernt Schiele. Long-term on-board prediction of people in traffic scenes under uncertainty. In *CVPR*, 2018. 3

[3] Wonmin Byeon, Qin Wang, Rupesh Kumar Srivastava, and Petros Koumoutsakos. ContextVP: Fully context-aware video prediction. In *ECCV*, 2018. 2

[4] Lluıs Castrejon, Nicolas Ballas, and Aaron Courville. Improved conditional vrnns for video prediction. In *ICCV*, 2019. 2

[5] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016. 2, 6

[6] Emily Denton and Rob Fergus. Stochastic video generation with a learned prior. In *ICML*, 2018. 1, 2

[7] Emily L. Denton and Vighnesh Birodkar. Unsupervised learning of disentangled representations from video. In *NeurIPS*, 2017. 3

[8] Nemanja Djuric, Vladan Radosavljevic, Henggang Cui, Thi Nguyen, Fang-Chieh Chou, Tsung-Han Lin, and Jeff Schneider. Motion prediction of traffic actors for autonomous driving using deep convolutional networks. *arXiv:1808.05819*, 2018. 3

[9] Chelsea Finn, Ian Goodfellow, and Sergey Levine. Unsupervised learning for physical interaction through video prediction. In *NeurIPS*, 2016. 1, 8

[10] Hang Gao, Huazhe Xu, Qi-Zhi Cai, Ruth Wang, Fisher Yu, and Trevor Darrell. Disentangling propagation and generation for video prediction. In *ICCV*, 2019. 2

[11] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *I. J. Robotics Res.*, 2013. 6, 7

[12] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, Warde-Farley, David, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. In *NeurIPS*, 2014. 2

[13] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social GAN: socially acceptable trajectories with generative adversarial networks. In *CVPR*, 2018. 3

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 3

[15] Yung-Han Ho, Chuan-Yuan Cho, Wen-Hsiao Peng, and Guo-Lun Jin. Sme-net: Sparse motion estimation for parametric video prediction through reinforcement learning. In *ICCV*, 2019. 2

[16] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017. 6

[17] Xiaojie Jin, Huaxin Xiao, Xiaohui Shen, Jimei Yang, Zhe Lin, Yunpeng Chen, Zequn Jie, Jiashi Feng, and Shuicheng Yan. Predicting scene parsing and motion dynamics in the future. In *NeurIPS*, 2017. 3

[18] Kris M Kitani, Brian D Ziebart, James Andrew Bagnell, and Martial Hebert. Activity forecasting. In *ECCV*, 2012. 3

[19] Alex X. Lee, Richard Zhang, Frederik Ebert, Pieter Abbeel, Chelsea Finn, and Sergey Levine. Stochastic adversarial video prediction. *arXiv:1804.01523*, 2018. 2

[20] Bo Li, Wei Wu, Qiang Wang, Fangyi Zhang, Junliang Xing, and Junjie Yan. Siamrpn++: Evolution of siamese visual tracking with very deep networks. *arXiv:1812.11703*, 2018. 6

[21] Xiaodan Liang, Lisa Lee, Wei Dai, and Eric P Xing. Dual motion gan for future-flow embedded video prediction. In *ICCV*, 2017. 2

[22] Wenqian Liu, Abhishek Sharma, Octavia Camps, and Mario Sznaier. Dyan - a dynamical atoms-based network for video prediction. In *ECCV*, 2018. 2

[23] Ziwei Liu, Raymond Yeh, Yiming Liu Xiaoou Tang, , and Aseem Agarwala. Video frame synthesis using deep voxel flow. In *ICCV*, 2017. 2, 7, 8

[24] William Lotter, Gabriel Kreiman, and David Cox. Deep predictive coding networks for video prediction and unsupervised learning. In *ICLR*, 2017. 1, 2, 7, 8

[25] Pauline Luc, Natalia Neverova, Camille Couprie, Jacob Verbeek, and Yann LeCun. Predicting deeper into the future of semantic segmentation. In *ICCV*, 2017. 1

[26] Zelun Luo, Boya Peng, De-An Huang, Alexandre Alahi, and Li Fei-Fei. Unsupervised learning of long-term motion dynamics for videos. In *CVPR*, 2017. 3

[27] Yuexin Ma, Xinge Zhu, Sibo Zhang, Ruigang Yang, Wenping Wang, and Dinesh Manocha. Trafficpredict: Trajectory prediction for heterogeneous traffic-agents. In *AAAI*, 2019. 3

[28] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. Rectifier nonlinearities improve neural network acoustic models. In *ICML*, 2013. 3

[29] Michaël Mathieu, Camille Couprie, and Yann LeCun. Deep multi-scale video prediction beyond mean square error. In *ICLR*, 2016. 2

[30] V. Patraucean, A. Handa, and R. Cipolla. Spatio-temporal video autoencoder with differentiable memory. In *ICLR workshop*, 2016. 1

[31] Xiaojuan Qi, Zhengzhe Liu, Qifeng Chen, and Jiaya Jia. 3d motion decomposition for RGBD future dynamic scene synthesis. In *CVPR*, 2019. 3

[32] Fitsum A Reda, Guilin Liu, Kevin J Shih, Robert Kirby, Jon Barker, David Tarjan, Andrew Tao, and Bryan Catanzaro. Sdcnet: Video prediction using spatially-displaced convolution. In *ECCV*, 2018. 2

[33] Amir Sadeghian, Vineet Kosaraju, Ali Sadeghian, Noriaki Hirose, and Silvio Savarese. Sophie: An attentive GAN for predicting paths compliant to social and physical constraints. *arXiv:1806.01482*, 2018. 3

[34] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 4

[35] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhutdinov. Unsupervised learning of video representations using lstms. In *ICML*, 2015. 1, 2

[36] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. In *CVPR*, 2018. 3, 6

[37] Ilya Sutskever, Geoffrey E Hinton, and Graham W. Taylor. The recurrent temporal restricted boltzmann machine. In *NeurIPS*, 2009. 1

[38] Johan Vertens, Abhinav Valada, and Wolfram Burgard. Smsnet: Semantic motion segmentation using deep convolutional neural networks. In *IROS*, 2017. 3

[39] Ruben Villegas, Arkanath Pathak, Harini Kannan, Dumitru Erhan, Quoc V. Le, and Honglak Lee. High fidelity video prediction with large stochastic recurrent neural networks. In *NeurIPS*, 2018. 1

[40] Ruben Villegas, Jimei Yang, Seunghoon Hong, Xunyu Lin, and Honglak Lee. Decomposing motion and content for natural video sequence prediction. In *ICLR*, 2017. 2, 3, 7, 8

[41] Ruben Villegas, Jimei Yang, Yuliang Zou, Sungryull Sohn, Xunyu Lin, and Honglak Lee. Learning to generate long-term future via hierarchical prediction. In *ICML*, 2017. 1

[42] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Generating videos with scene dynamics. In *NeurIPS*, 2016. 3

[43] Jacob Walker, Carl Doersch, Abhinav Gupta, and Martial Hebert. An uncertain future: Forecasting from static images using variational autoencoders. In *ECCV*, 2016. 3

[44] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. Video-to-video synthesis. In *NeurIPS*, 2018. 1, 2, 7, 8

[45] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. 6

[46] Zhou Wang, Eero P. Simoncelli, and Alan C. Bovik. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, volume 2, pages 1398–1402. IEEE, 2003. 7

[47] Nevan Wichers, Ruben Villegas, Dumitru Erhan, and Honglak Lee. Hierarchical long-term video prediction without supervision. In *ICML*, 2018. 2

[48] Yuwen Xiong, Renjie Liao, Hengshuang Zhao, Rui Hu, Min Bai, Ersin Yumer, and Raquel Urtasun. Upsnet: A unified panoptic segmentation network. In *CVPR*, 2019. 3, 6

[49] Rui Xu, Xiaoxiao Li, Bolei Zhou, and Chen Change Loy. Deep flow-guided video inpainting. In *CVPR*, June 2019. 6

[50] Tianfan Xue, Jiajun Wu, Katherine Bouman, and Bill Freeman. Visual dynamics: Probabilistic future frame synthesis via cross convolutional networks. In *NeurIPS*, 2016. 1, 2

[51] Takuma Yagi, Karttikeya Mangalam, Ryo Yonetani, and Yoichi Sato. Future person localization in first-person videos. In *CVPR*, 2018. 3

[52] Yufei Ye, Maneesh Singh, Abhinav Gupta, and Shubham Tulsiani. Compositional video prediction. In *ICCV*, 2019. 2

[53] Zhichao Yin and Jianping Shi. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In *CVPR*, 2018. 4, 6

[54] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S. Huang. Generative image inpainting with contextual attention. In *CVPR*, 2018. 3, 5, 6

[55] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 7

[56] Weiyu Zhang, Menglong Zhu, and Konstantinos Derpanis. From actemes to action: A strongly-supervised representation for detailed action understanding. In *ICCV*, 2013. 8

[57] Yi Zhu, Karan Sapra, Fitsum A. Reda, Kevin J. Shih, Shawn Newsam, Andrew Tao, and Bryan Catanzaro. Improving semantic segmentation via video propagation and label relaxation. In *CVPR*, 2019. 3, 6