Temporal Scene Completion with Generative Adversarial Networks

Minyoung Huh (Audit)
Robotics Institute
jmhuh@cs.cmu.edu

Martin Li
Robotics Institute
mtli@cs.cmu.edu

Ishan Nigam
Robotics Institute
inigam@cs.cmu.edu

1 Introduction

What I cannot build, I do not understand.
- Richard Feynman

Adversarial learning [6] has significantly influenced the landscape of data-driven representation learning. Classical representation learning requires the task to be explicitly defined, so is the objective function used to learn representations. Adversarial learning, on the other hand, relies on modeling the data distribution via an *adversarial* game. The generator network learns to generate samples of data that are close to the underlying true data distribution, while the discriminator network learns to determine if the generated data samples are close to this underlying distribution. Richard Feynman's famous quote (mentioned above) rings true several decades later - the ultimate expression to understanding something is the ability to generate it from the ground up. If a Generative Adversarial Network(GAN) is trained successfully, *which by itself is not a trivial task*, then the network will learn how to apply its learned knowledge of the data distribution to myriad tasks. In the past, GANs have been utilized for learning style transfer [7], anime character generation [8], 3D object generation and reconstruction [9], and image super-resolution [10]. Recently, Isola et al. [11] showed brilliant results about transferring knowledge at the pixel-level conditionally onto output data to generate structurally and semantically coherent output images. A few other works have also followed this line of inquiry and are discussed in the Related Work section.

Understanding the dynamics of a video sequence requires an innate understanding of the low-level, mid-level, and high-level representations in a video. Thus, pixel-level video understanding is an important task for data-driven Computer Vision. Contextual temporal scene completion has been a class of works in Computer Vision and Graphics that tries to fill in the missing content in a video. In the past for images, optimization using nearest-neighbor approaches have shown great results. However, nearest-neighbor methods are expected to break down if the scene evolves temporally. Since conditional generation of images from images proves to be such a well-conditioned problem, conditional generation of image frames from video sequences is possible because it is a natural extension of conditional image-to-image generation.

Data-driven Computer Vision techniques may be used to generate missing or corrupted frames for broken video sequences or to generate high FPS video sequences from corresponding low FPS video sequences. We propose to perform the former of these two tasks - conditional image generation for multiple missing image frames - using Generative Adversarial Networks. It is important to draw a distinction here from a related branch of research - video prediction. In the past, Variational Autoencoders [12] and GANs [1, 2, 13] have been used to predict future frames from past frames. Our problem differs from this in the sense that we strictly intend to generate intermediate frames for which both past and future frames are available. Thus, our problem is a natural extension of using more temporal context to model video sequences.

Our research explores the use of two recent data-driven representation learning frameworks for conditional generation of image frames. *First*, we perform image sequence to image sequence translation with conditional adversarial networks, extending the popular approach by Isola et al.

10707 Class Project (2017).

[11] (hereon, refered to as *pix_to_pix*) to condition multiple image frame generation on past and future image frames. *Second*, we perform image sequence synthesis by appearance flow, extending the wonderful framework proposed by Zhou et al. [24] (hereon refered to as *appearance_flow*) to synthesize multiple image frames from past and future image frames. While we struggle with the realities of training Generative Adversarial Networks with novel architectures, we discuss the lessons we learned and the (limited) conclusions we were able to draw, hoping to continue improving upon this body of work in the future.

Section 2 describes the literature related to our task. Section 3 describes the datasets that we worked with. Section 4 follows with the proposed frameworks. Section 5 describes our experimental results. Section 6 discusses the lessons we learned during the course of our project. Finally, Section 7 provides the conclusions and possible future extensions of our research.

2 Related Work

Adversarial learning was first described by Goodfellow et al. [6]. This was followed up by the Deep Convolutional GAN [15] which popularized the technique with realistic visualizations and stabilized training efforts. The Laplacian pyramid method [31] for stablizing GAN image generation allowed for non-trivial image resolutions to be synthesized. While the literature on adversarial learning is quite vast, we focus on studying conditional GAN methods.

While Pix2Pix forms the backbone for one of the proposed methods for our study for conditional generation of video frames, there were prior studies GANs conditioned on various forms of supervision. Prior works learnt conditioned representations based on information such as discrete labels [17] and text [18]. Image-conditional models have also tackled inpainting [19], image prediction from a normal map [20], and style transfer [21]. However, Pix2Pix is the first robust method that is agnostic of the application task. It's follow-up work - Cycle GANs [22] goes one step further where images are translated in the absence of paired examples.

Disentangling an image into various factors of variation has been a well documented form of representation learning. Hinton et al. [25] proposed to learn a hierarchy of *capsules* that represent local transformations of the image. Jaderberg et al. [26] proposed spatial transformer networks that perform global transformation over input features. Interestingly, Jayaraman et al. [27] study the synthesis of ego-motion transformed features as an auxiliary supervisory signal for learning better semantic representations. While the above-mentioned approaches demonstrate the ability to disentangle factors of variation, the view manipulations demonstrated are typically restricted to mild transformations with limited visual appearance variance. A CNN may learn [28] to function as a renderer by generating chairs given structural and viewpoint information in the form of explicit graphics code. Tatarchenko et al. [29] and Yang et al. [30] follow up this wonderful work by observing that the structural and viewpoint information may not necessarily be provided as explicit graphics code. Instead, we may implicitly implicitly capture the desired transformation in the image itself. Our second proposed framework is the approach to synthesizing novel views of an image using appearance flow [24]. We extend this framework to utilizing multiple frames and the associated optical flow towards synthesizing multiple novel image frames.

Our work corresponds closely to the task of video prediction. There have been several studies that rely on data driven methods to tackle this problem. Mathieu et al. [1] train a convolutional network to generate future frames given an input sequence, using a multi-scale architecture utilizing adversarial training, and an image gradient difference loss function instead of standard MSE. Vondrick et al. [13] use large amounts of unlabeled video to learn a model of scene dynamics with a spatio-temporal convolutional architecture that untangles the scene's foreground from the background. Recently, Walker et al. [23] have suggested modeling the forecasting problem at a higher level of abstraction, exploiting human pose as supervision to model the forecasting as modeling the high level structure of active objects and using a variational auto-encoder to model the possible future movements of humans, and then using the generated poses as conditional information for a GAN to predict the future frames of the video in pixel space. While all these are wonderful works, we wish to distinguish our work with the argument that we intend to perform video interpolation, and not video prediction.

3 Dataset

Generative Adversarial Networks are known to be able to learn to model the data distribution from even a few hundred conditional labels. We work with a subset of the Human Motion Database [4], which consists of a total of 6849 clips of resolution 320×240. The videos include scenarios such as general facial actions, facial actions with object manipulation, general body movements, body movements with object interaction, body movements for human interaction.

Figure 1: Image frames from four video sequences from the walk category utilized in our project from the HMDB dataset. The wide range of motions as well as distincy backgrounds demonstrate the diversity of visual appearances present within the dataset.

Specifically, we work with the *Walk* subcategory of the HMDB dataset. We choose this category in the hopes that such an action is likely to be performed both indoors and outdoors and is general enough in itself (as compared to, say "draw sword" or "brush hair") that it captures a wide variety of visual appearances. Table 1 describes a few statistics for our dataset. Below, we provide a few representative samples of the dataset. The total number of frames extracted from 549 video sequences are 49315 frames at 30 FPS.

Figure 2: Image frames from four video sequences from the Penn Action dataset, utilized in our project for the *appearance_flow* architecture.

Since the HMDB dataset did not provide a lot of valuable insights into the process of conditionally generating image sequences, we also explored static background images from Penn Action Dataset [32], which contains 2326 video sequences of 15 different actions for each sequence. We extract all the data at 30FPS and perform the same operations on the Penn Action dataset as for the HMDB dataset. Due to lack of space, we do not provide any visual samples of the Penn Action Dataset. It suffices to say that the visual samples are much more simpler than the HMDB dataset samples provided above.

4 Proposed Framework

We perform conditional image generation for video sequences using two methods:

1. Image-to-Image Translation with Conditional Adversarial Networks by Isola et al. [11],
2. View Synthesis by Appearance Flow by Zhou et al. [24].

We assume multiple frames in the video are lost during transmission or compression. The loss may occur once or periodically occur multiple times. It is prudent to point out that the periodic case of multiple image frames being lost is equivalent to temporal super-resolution, colloquially known in the community as frame rate up-conversion (FRUC). Our goal is to generate the missing frames based on prior and posterior visual information present in the immediately preceding and the immediately following available image frames. For simplicity, we assume the the image frame is missing as a whole, which implies that we cannot rely on spatial information within the same frame to perform temporal scene completion.

[As an aside, we believe this would be an interesting line of inquiry since this would possibly incorporate attention mechanisms to decide what spatial locations are required to be used for performing the reconstruction. While this would be an interesting research thread to pursue, we believe this lies outside of the scope of the project.]

We implement an optical flow and a simple image frame interpolation method as simple, yet powerful, baselines. Below, we briefly describe the qualitative nature of these methods. Next, we discuss the two proposed extensions for *pix_to_pix* and *appearance_flow*.

4.1 Baseline Methods

Interpolation is a natural way to fill in the missing values in between data points. The first baseline approach is naive interpolation. Next, we use state-of-art optical flow estimation methods to perform interpolation of video sequence image frames to fill in the missing frames. Let n be the number of frames dropped, $\mathbf{I}_s$ be the last frame before the loss, and $\mathbf{I}_e$ be the first frame after the loss. The interpolation method generates the i-th dropped frame in the following way:

$$\alpha = i/(n+1) \tag{1}$$

$$\mathbf{I}_i = (1-\alpha)\mathbf{I}_s + \alpha\mathbf{I}_e \tag{2}$$

A more involved approach towards performing interpolation is to estimate the optical flow between the frames to guide the interpolation. We first estimate the optical flow using the Farnback method [32] from $\mathbf{I}_s$ to $\mathbf{I}_e$, then we use the flow vectors multiplied by the α (which is the bilinear sampling factor), to offset the sampling grid, before passing it to a standard interpolation function.

4.2 Image to Image Translation with Conditional Adversarial Networks (*pix_to_pix*)

The *pix_to_pix* proposes conditional adversarial networks to solve conditional image translation problems. The network learns the mapping from input image to output image, but also learn a loss function to train this mapping. We extend this framework by conditioning on a stack of either RGB mage frames or the associated optical flow from these image frames. Below, we briefly discuss the underlying architecture and optimization we are trying to perform. The basic architecture of *pix_to_pix* is based on the DCGAN architecture by Radford et al. [15]. The generator consists of skip connections and is comprised of a 9-block residual architecture. This is a departure from the original *pix_to_pix* architecture which employed a U-Net style architecture.

4.2.1 Generator

The input is passed through a series of layers that progressively downsamples until a bottleneck layer at which point the process is reversed. Such a network requires that all information flow pass through all the layers. For many image translation problems, there is a great deal of low-level information shared between the input and output, and it would be desirable to shuttle this information directly across the net. To give the generator a means to circumvent the bottleneck for information iskip connections are introduced.

4.2.2 Discriminator

The L_2 and L_1 losses produce blurry results since they tend to capture low-frequency information and penalize the absence of low-level structural similarities. Thus, the correctness of the low frequencies is automatically encoded into the L_1 metric. This motivates restricting the GAN discriminator to only model high-frequency structure, relying on an L_1 term to force low-frequency correctness. Thus, the final objective resembles the following:

$$G^1 = \arg\min_G \max_D L_{cGAN}(G, D) + \lambda L_{L1}(G) \tag{3}$$

However, we still have not considered how high-frequency information may be captured. The structure in local image patches comes to the rescue; thus, the discriminator architecture termed as PatchGAN penalizes structure at the scale of patches. Thus, instead of making decisions about the global image, we make multiple decisions, once for each patch. The discriminator runs as a convolutional operator to determine whether each patvch is real or fake, averaging the responses to determine the final output of the discriminator.

4.3 View Synthesis by Appearance Flow (*appearance_flow*)

Generative adversarial models have recently gained immense popularity in generating images; yet for video generation, even the smallest deviation in color can look anomalous. Hence, we used a method by Zhou et al. [24] that predicts a flow field given an image. Analogous to optical flow, the model predicts the displacement between the original pixels and target pixels. Bilinear sampling from the original image using the predicted flow field, the model can generate photo-realistic results while preserving the original pixel content. To adopt this method to video hole filling, we trained a CNN parameterized by weights θ that takes both the start frame and the end frame, and predicts flow fields for both images.

Figure 3: Range of movements in the HMDB dataset.

$$f_s, f_e = F_\theta(I_s, I_e) \tag{4}$$

If we define a bilinear sampler as B that takes 2 arguments: image and the flow field, we can generate the predicted images $\hat{I}_s, \hat{I}_e$ using the predicted flow:

$$\hat{I}_s = B(I_s, f_s) \tag{5}$$

$$\hat{I}_e = B(I_e, f_e) \tag{6}$$

Since we have two candidates for the prediction, we generate a confidence mask using a CNN parameterized by weights ϕ:

$$m = G_\phi(I_s, I_e, f_s, f_e) \tag{7}$$

By weighting the prediction using the mask, we can minimize the L_1 loss with the ground truth.

$$\min |I_t - ((m)\hat{I}_s + (1-m)\hat{I}_e)| \tag{8}$$

In Figure 3, we show some results using the flow field method on Penn Action Dataset [5]. While the generated results look more photo-realistic than the previous approach, we observed that our model

Penn Action	Baseline_Optic_Flow	Baseline_Interpolation	ganWithAppearanceFlow
SSIM	0.8504	0.8781	0.8608
Euclidean Loss	2182.2	2024.9	6620.1
HMDB	Baseline_Optic_Flow	Baseline_Interpolation	GAN
SSIM	0.8962	0.9397	0.9014
Euclidean Loss	1985.3	1766.8	1941.2

Table 1: We provide here the baseline optical flow method, the interpolation method on both the Penn Action and the HMDB datasets. The GAN method for Penn Action dataset is our *appearance_flow* architecture. The GAN method for HMDB dataset is our *pix_to_pix* dataset.

learns an auto-encoder instead because the discrepancy between the input frames and the target frame was too small. For example, the area of motion is significantly smaller than the static background.

5 Results

We first provide baseline visual results for the interpolation methods. Next, we discuss the performance of the *pix_to_pix* architecture on the HMDB dataset and the performance of the *appearance_flow* method on the Penn Action dataset.

5.1 Modes of evaluation

Since human evaluations for determining quality of generated image frames is beyond the scope of the project, we describe the modest methods of our evaluation protocol. Conditional GAN formulations rely on modeling a structured loss, which jointly models the output feature space. We propose to use:

- the SSIM metric [14] to understand how well our method captures the high level image statistics, and
- the Euclidean loss to understand how well our method captures the low level image statistics

5.2 Baseline

The visual results for generating a single dropped frame are shown in Figure 4 while the results for generating two consecutive dropped frames are shown in Figure 5. We can see from those figures that interpolation blurs the frames, but does not introduce any deformation of the objects even with the number of dropped frames increasing. The optical flow method on the other hand, performs really well with small motion. If only the object in focus is moving in the video, it can accurately reconstruct the frame. However, when there are motions every where in the frame and with more dropped frames, we start to see deformation of the objects. This is expected, as the pixels of the same object does not necessarily move coherently due to the movement of the camera. We hope our proposed method can fix those problems by trying to understand the frame content through deep learning and adversarial training.

Figure 4: Baseline methods generating a single dropped frame based on the temporal context.

5.3 Conditional Image Generation - *pix_to_pix*

The results for two different experiments are shown:

Figure 5: Baseline methods generating two consecutive dropped frames based on the temporal context.

- Generating images conditioned on stacks of 2 preceding and 2 following image frames. This is shown in Figure
- Generating images conditioned on stacks of 1 preceding and 1 following pairs of image frames and optical flow.

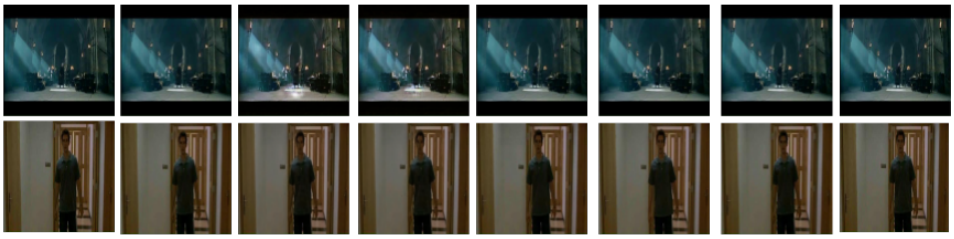

Figure 6: The pix_to_pix method with image frames. From left: frame1, frame2, generatedFrame3, generatedFrame4, frame5, frame6, groundTruthFrame3, groundTruthFrame4.

Figure 7: The pix_to_pix method with image frames and optical flow. From left: opticalFlowframe12, opticalFlowFrame23, frame1, generatedFrame2, generatedFrame3, frame4, opticalFlowframe34, opticalFlowFrame45, ,groundTruthFrame2, groundTruthFrame3

5.4 Image Synthesis with Appearance Flow - *appearance_flow*

6 Discussion and Analysis

In all of our experiments we have observed that our model fails to capture the motion of the video correctly. We suspect that this is an issue in the loss function. The current loss formulation does not encourage the model to understand motion, as the motion between subsequent frames are very subtle. Our results indicate that using the reconstruction loss as the main learning signal simply encourages the model to reconstruct the original input, similar to that of an autoencoder. While using a carefully annotated data on a static video could alleviate the problem by forcing the model to only incur loss

Figure 8: The Appearance Flow method with image frames. From left: frame1, generatedFrame2, frame3, groundTruthFrame2

on the region of motion, videos in real life are not static. We wish to further investigate how we could incur loss on the salient region of the data.

Unfortunately, our architectures do not do better than the interpolation baseline. We believe this is likely due to our inexperience in training GANs, and

7 Conclusion

In this paper, we studied unsupervised data-driven approaches to perform conditional hole filling in video sequences. We explored 2 popular generative models: generative adversarial networks and apperance flow using flow-field. We extend these works to conditionally generate multiple image frames. We found that both methods provided competitive generations compared to the ground truth. In addition, we found that using appearance flow approach tends to perform slightly better. While both of the models explored can generate photo-realistic results, we observed that our model has difficulty in understanding objects in images and its corresponding physics. While the solution to the problem in uncertain, we hypothesize 2 things: 1) We believe that using reconstruction loss does not provide a strong enough learning signal for model to learn object motion, and a better loss formulation is required. 2) training a model from scratch may never learn rich features to understand the context and the physics of the image, and pre-training our model on object classification such as ImageNet or using feature loss may be required. We wish to further investigate this problem in the future.

References

[1] Mathieu, Michael & Couprie, Camille & LeCun, Yann "Deep Multi-scale Video Prediction Beyond Mean Square Error", *International Conference on Learning Representations* (2016).

[2] Vondrick, Carl & Pirsiavash, Hamed & Torralba, Antonio "Generating Videos with Scene Dynamics". *Advances in Neural Information Processing Systems* (2016).

[3] Vondrick, Carl & Torralba, Antonio "Generating the Future with Adversarial Transformers". *IEEE Conference on Computer Vision and Pattern Recognition* (2017).

[4] Kuehne, H & Jhuang, H & Garrote, E & Poggio, T & Serre, T "HMDB: A Large Video Database for Human Motion Recognition". *International Conference on Computer Vision* (2011).

[5] H. Kiani Galoogahi & A. Fagg & C. Huang & D. Ramanan & S.Lucey "Need for Speed: A Benchmark for Higher Frame Rate Object Tracking". *International Conference on Computer Vision* (2017).

[6] Goodfellow, Ian & Jean Pouget-Abadie & Mehdi Mirza & Bing Xu & David Warde-Farley & Sherjil Ozair & Aaron Courville & Yoshua Bengio. "Generative adversarial nets." *Advances in Neural Information Processing Systems* (2014).

[7] Gatys, L. A. & Ecker, A. S. & Bethge, M. "A neural algorithm of artistic style". *ArXiv:1508.06576*. (2015)

[8] Yanghua Jin & Jiakai Zhang & Minjun Li & Yingtao Tian & Huachun Zhu & Zhihao Fang. "Towards the Automatic Anime Characters Creation with Generative Adversarial Network". *ArXiv:1708.05509*. (2017)

[9] Choy, Christopher B & Xu, Danfei & Gwak, JunYoung & Chen, Kevin & Savarese, Silvio. "3D-R2N2: A Unified Approach for Single and Multi-view 3D Object Reconstruction". *European Conference on Computer Vision*. (2016)

[10] Christian Ledig & Lucas Theis & Ferenc Huszar & Jose Caballero & Andrew Cunningham & Alejandro Acosta & Andrew Aitken & Alykhan Tejani & Johannes Totz & Zehan Wang & Wenzhe Shi. "Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network". *ArXiv:1609.04802* (2016)

[11] Phillip Isola & Jun-Yan Zhu & Tinghui Zhou & Alexei Efros, "Image-to-Image Translation with Conditional Adversarial Networks" *Convference on Computer Vision and Pattern Recognition* (2017).

[12] Jacob Walker & Carl Doersch & Abhinav Gupta & Martial Hebert. "An Uncertain Future: Forecasting from Variational Autoencoders", *European Conference on Computer Vision*. (2016)

[13] Carl Vondrick & Hamed Pirsiavash & Antonio Torralba. "Generating Videos with Scene Dynamics", *Advances in Neural Information Processing Systems*. (2016)

[14] Z. Wang & A. C. Bovik & H. R. Sheikh & E. P. Simoncelli. "Image quality assessment: from error visibility to structural similarity". *IEEE Transactions on Image Processing*. (2004)

[15] Alec Radford & Luke Metz & Soumith Chintala. "Unsupervised Representation Learning with Deep Convolutonal GANs". *International Conference on Learning Representations*. (2016)

[16] Pix2Pix PyTorch implementation - https://github.com/junyanz/pytorch-CycleGAN-and-pix2pix

[17] M. Mirza & S. Osindero. "Conditional generative adversarial nets". *ArXiv:1411.1784*. (2014)

[18] S. Reed & Z. Akata & X. Yan & L. Logeswaran, & B. Schiele & H. Lee. "Generative adversarial text to image synthesis". *ArXiv:1605.05396*. (2016)

[19] D. Pathak & P. Krahenbuhl & J. Donahue & T. Darrell & A. Efros. "Context encoders: Feature learning by inpainting". *Conference on Computer Vision and Pattern Recognition*. (2016)

[20] X. Wang & A. Gupta. "Generative image modeling using style and structure adversarial networks". *European Conference on Computer Vision*, (2016)

[21] C. Li & M. Wand. "Precomputed real-time texture synthesis with markovian generative adversarial networks". *Europena Conference on Computer Vision*. (2016)

[22] Zhu, Jun-Yan & Park, Taesung & Isola, Phillip & Efros, Alexei, "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks". *ArXiv:1703.10593*. (2017)

[23] Jacob Walker & Kenneth Marino & Abhinav Gupta & and Martial Hebert. "The Pose Knows: Video Forecasting by Generating Pose Futures". *International Conference on Computer Vision*. (2017)

[24] Tinghui Zhou & Shubham Tulsiani & Weilun Sun & Jitendra Malik & Alexei A. Efros. "View Synthesis by Appearance Flow". *European Conference on Computer Vision*. (2016)

[25] G. Hinton & A. Krizhevsky & S. Wang. "Transforming auto-encoders". *Artificial Neural Networks and Machine Learning–ICANN*. (2011)

[26] M. Jaderberg & K. Simonyan & A. Zisserman, A. "Spatial transformer networks". *Advances in Neural Information Processing Systems*. (2015)

[27] D. Jayaraman & K. Grauman. "Learning image representations tied to egomotion". *IEEE International Conference on Computer Vision*. (2015)

[28] A. Dosovitskiy & J. Springenberg,& T. Brox. "Learning to generate chairs with convolutional neural networks". *IEEE International Conference on Computer Vision and Pattern Recognition*. (2015)

[29] M. Tatarchenko, & A. Dosovitskiy & T. Brox. "Single-view to multi-view: Reconstructing unseen views with a convolutional network". *arXiv: 1511.06702*. (2015)

[30] J. Yang & S. Reed & M. Yang & H. Lee. "Weakly-supervised disentangling with recurrent transformations for 3d view synthesis". *Advances in Neural Information Processing Systems*. (2015)

[31] E. Denton & S. Chintala & Rob Fergus. "Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks". *Advances in neural information processing systems*. (2015)

[32] G. Farnebäck. "Two-frame motion estimation based on polynomial expansion". *Image analysis*. (2003)