ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2.
Available from http://www.i6doc.com/en/.

Motion Segmentation using Frequency Domain Transformer Networks

Hafez Farazi and Sven Behnke

University of Bonn, Computer Science Institute VI, Autonomous Intelligent Systems
Endenicher Allee 19a, 53115 Bonn, Germany
{farazi, behnke}@ais.uni-bonn.de

Abstract. Self-supervised prediction is a powerful mechanism to learn representations that capture the underlying structure of the data. Despite recent progress, the self-supervised video prediction task is still challenging. One of the critical factors that make the task hard is motion segmentation, which is segmenting individual objects and the background and estimating their motion separately. In video prediction, the shape, appearance, and transformation of each object should be understood only by predicting the next frame in pixel space. To address this task, we propose a novel end-to-end learnable architecture that predicts the next frame by modeling foreground and background separately while simultaneously estimating and predicting the foreground motion using Frequency Domain Transformer Networks. Experimental evaluations show that this yields interpretable representations and that our approach can outperform some widely used video prediction methods like Video Ladder Network and Predictive Gated Pyramids on synthetic data.

1 Introduction

Many of the recent models for video prediction use a huge number of parameters, which results in scalability issues and lack of interpretability. Furthermore, these large networks take days to train on even synthetic datasets, which makes exploring new ideas more difficult in comparison with lightweight differentiable models, which only need minutes for training. More importantly, due to the high number of parameters used in heavy models, they tend to overfit the training set and do not easily generalize to novel data. One way to address these issues it to prestructure the models based on domain knowledge. Of course, manually engineering every aspect of video prediction is not possible, and one has to find a good balance between nature—inductive bias, which is optimized on an evolutionary time scale—and nurture—learning from own experience.

In this work, we propose a model for motion segmentation that has zero trainable parameters and is fully interpretable. It models foreground and background separately. Our model estimates and predicts foreground motion using Frequency Domain Transformer Networks [1]. We extend this model by adding a few learnable parameters. The improvements made by the added parameters are fully explainable and rational. The code and dataset of this paper are publicly available.[1]

[1] `https://github.com/AIS-Bonn/MotionSegmentation`.

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2.
Available from http://www.i6doc.com/en/.

2 Related Work

Despite much progress in the field, self-supervised video prediction is still a challenging task. One fundamental issue in video prediction is that the predictor has to segment the scene into individual objects and background and to infer corresponding motions. One attempt to address segmentation of static images is Tagger [2]. The Tagger network learns to group the representations of different objects and backgrounds iteratively in a self-supervised way. Hsieh et al. [3] proposed Spatial Transformer Network [4] to decompose video frames into individual objects and model their motion separately. Other works do not explicitly model moving segments and rely on unstructured recurrent models to learn these bindings. For example, Cricri et al. [5] added recurrent lateral connections in Ladder Networks to capture the temporal dynamics of video. Recurrent connections and lateral shortcuts relive the deeper layers from modeling spatial detail. The VLN network achieves competitive results to Video Pixel Networks, the state-of-the-art on Moving MNIST dataset, using fewer parameters.

Some works try to learn image relations by separating content and transformation. For instance, PGP [6], which is based on a gated autoencoder model [7], has the assumption that two temporally consecutive frames can be modeled as a linear transformation of each other. In the PGP model, by using a bi-linear model, the hidden layer of mapping units encodes the transformation. These transformation encodings are then used in a hierarchy to predict the next frame. Conv-PGP [8] significantly reduces the number of parameters by utilizing convolutional layers. When predicting video that has location-dependent features, Azizi et al. [9] proposed location-dependent convolutional layers that can model, for example, bouncing on the borders.

Another related work is Predictive-Corrective networks [10], which sequentially make top-down predictions and then correct those predictions with bottom-up observations for the action recognition task. This model adaptively focuses on surprising images where predictions require significant corrections. More recently, Hur et al. [11] proposed the Iterative Residual Refinement network for jointly predicting optical flow and estimating occlusions.

3 Motion Segmentation Network

3.1 Prediction-correction State Estimation

Similar to [10], we were inspired by classic linear dynamical systems theory and Kalman filters. In a Kalman filter, x_t is a noisy linear function of the previous time step state x_{t-1}. The observation z_t is modeled as a noisy linear function of the state x_t:

$$\begin{aligned} \mathbf{x}_t &= \mathbf{F}\mathbf{x}_{t-1} + \text{Noise} \\ \mathbf{z}_t &= \mathbf{H}\mathbf{x}_t + \text{Noise} \end{aligned} \quad , \tag{1}$$

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2. Available from http://www.i6doc.com/en/.

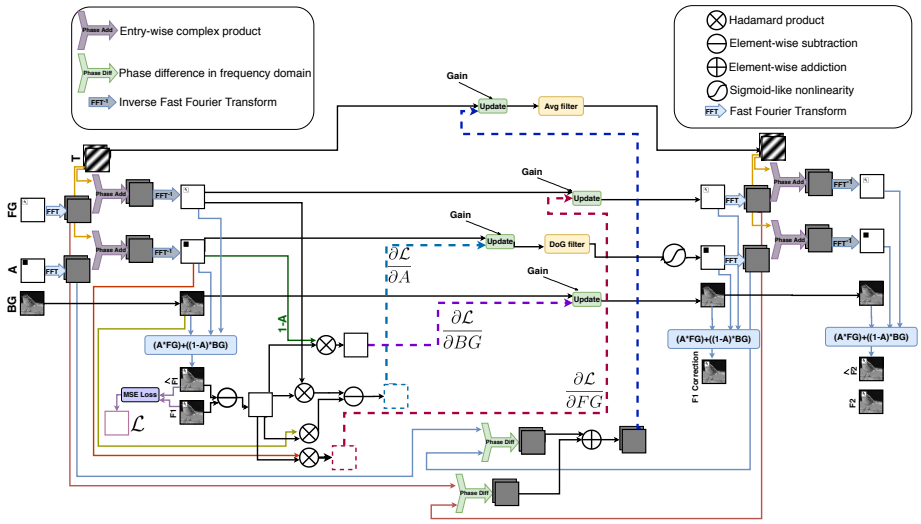

Fig. 1: Motion segmentation model. Foreground (FG) and background (BG) are modeled separately and combined using an alpha mask (A) to the predicted frame $\hat{\mathsf{F}}1$, which is compared to the input frame F1. The prediction error is used to update FG, BG, and A. For FG and A, motion is estimated by computing phase differences in Fourier space (T). This motion estimate is added to the phases of FG and A to move them accordingly. After a few steps of this prediction-correction cycle, the model does not need the input frames anymore and can continue predicting using only the estimated state (FG, BG, A, T).

where F is the state-transition matrix, and H is the measurement matrix. Under these assumptions, the posterior estimate of the state x_t is calculated by:

$$\hat{\mathbf{x}}_t = \underbrace{\hat{\mathbf{x}}_{t|t-1}}_{\text{prediction}} + \underbrace{\mathbf{K}(\mathbf{z}_t - \hat{\mathbf{z}}_{t|t-1})}_{\text{correction}}, \tag{2}$$

where $\hat{x}_{t|t-1}$ and $\hat{z}_{t|t-1}$ are the predictions of x_t and z_t, respectively, given observations $z_1, \ldots, z_{t-1}$. K is the Kalman gain matrix, which controls how much we rely on the current prediction $\hat{x}_{t|t-1}$ versus the observation z_t.

3.2 Frequency Domain Motion Segmentation

Fig. 1 illustrates our model for self-supervised motion segmentation. We model foreground (FG_t) and background (BG_t) separately as images having the same size as the observed frames (F_t). Both are combined by modeling occlusion of the background by the foreground using the alpha mask $\hat{\mathsf{A}}_t$:

$$\hat{\mathsf{F}}_t = \hat{\mathsf{A}}_t \cdot \hat{\mathsf{FG}}_t + (1 - \hat{\mathsf{A}}_t) \cdot \hat{\mathsf{BG}}_t. \tag{3}$$

In addition to these three images, the state also consists of the estimated common movement speed T_t of foreground and alpha mask. T_t is represented as

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2. Available from http://www.i6doc.com/en/.

phase differences (unit length complex numbers) between consecutive frames in the Fourier domain. It has the same size as the images. As in the Frequency Domain Transformer Networks [1], the next foreground frame $(\hat{\mathsf{FG}}_t, \hat{\mathsf{A}}_t)$ can easily be predicted by phase-adding T_t to the Fourier representations $FFT(.)$ of $(\mathsf{FG}_{t-1}, \mathsf{A}_{t-1})$ which is realized by element-wise multiplication of these complex matrices. After going back to the spatial domain by the inverse Fourier transformation $FFT^{-1}(.)$, the foreground and alpha mask are moved according to the estimated movement speed.

We calculate the difference between the predicted frame $\hat{\mathsf{F}}_t$ and observed frame F_t and update each part of the state to minimize the mean squared loss $\mathcal{L}(\hat{\mathsf{F}}_t, \mathsf{F}_t)$. As the predicted frame is computed by a simple differentiable function graph, we can easily perform gradient descent by a function graph for the backward pass that has the same structure. Instead of using automatic differentiation packages for updating each state, we hard-wired gradient computation in our computational graph. This results in a computation graph that realizes a Kalman filter-like prediction-correction cycle in its forward pass. For updating the state T_t, which is in Fourier space, we calculate the phase differences between $FFT(\mathsf{A}_t)$ and $FFT(\mathsf{A}_{t-1})$ as well as $FFT(\mathsf{FG}_t)$ and $FFT(\mathsf{FG}_{t-1})$. For computing T_t, we take the weighted average between T_{t-1} and calculated phase differences $\tilde{\mathsf{T}}_t$.

We also include two filtering mechanism for A and T after each update. With the assumption of blob-like response in A, we apply a Difference of Gaussian filter. We also filter the phase difference by an averaging filter, with the assumption that the phase difference between adjacent rows and adjacent columns are near-constant. The effect of removing phase filtering is illustrated in Fig. 2(c).

The proposed model hard-wires our assumptions that the foreground moves in front of a stationary background and that it occludes the background according to the alpha mask. Furthermore, we hard-wire motion estimation and prediction by Frequency Domain Transformer Networks [1].

3.3 Model Extension by Learnable Layers

So far, our model has zero learnable parameters. Hence, it hard-wires our assumptions, but cannot learn to exploit the statistical properties of data. Since our prediction-correction computation graph is differentiable, we can backpropagate a loss through the network that is unfolded in time. Hence, any parameter can be updated by gradient descent, and we can easily add parameters at suitable computation steps.

For initializing the spatial states FG, BG, and A we use three different convolutional networks. Each has four convolutional layers, with DenseNet-like connections, followed by ReLU activations. We also initialize T using the first two steps of A and FG. At each time step, each state is the weighted average between the updated state and the output of the convolutional network. We use a decaying gain for this weighted average so that in the initial step, we only use the convolutional network output, and later we rely more and more on the updated states. Note that the convolutional network also fills-in occluded parts

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2.
Available from http://www.i6doc.com/en/.

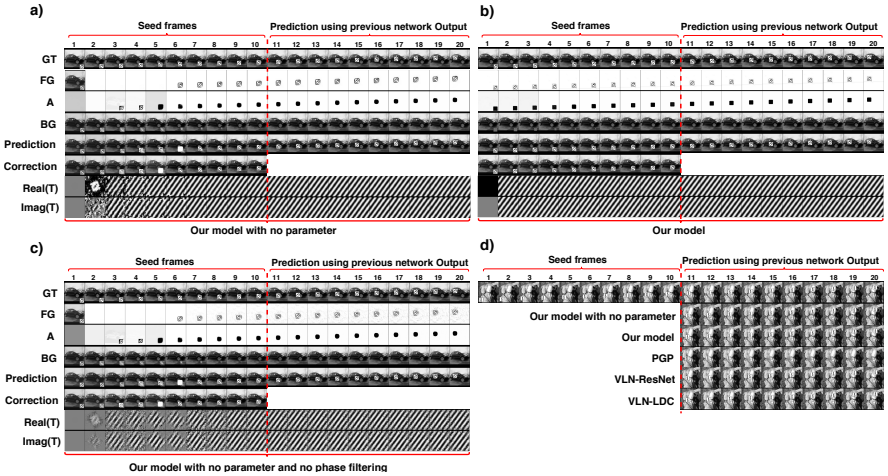

Fig. 2: a,b) Internal states' development for one sample Moving MNIST sequence (randomly selected). Note that albeit with varying success levels, both models can segment foreground and background and estimate foreground motion. c) Effect of removing phase filtering. d) Predictions for a randomly selected sample with different models.

of the background BG.

4 Experimental Results

4.1 Dataset and Training

We use a variant of the *Moving MNIST* data set to evaluate our proposed architecture. It contains twenty frames with one MNIST image, moving inside a 128×128 frame. Foreground moving objects were chosen randomly from training and test set and placed at a random position with a random velocity and random background image chosen from the STL-10 dataset. Note that the objects are moved with subpixel velocity.

The models and the update gains are trained end-to-end using backpropagation through time. We used Adam optimizer and MSE prediction loss as well as the cyclic learning rate.

4.2 Evaluation

We evaluate our architecture against Conv-PGP and two different VLN models. In these experiments, we predicted ten frames from ten seed inputs. Sample results of our models, as well as used baselines, are presented in Fig. 2. Fig. 2 also shows the development of the internal states of our two model variants for one moving MNIST sample in detail. The representations are easily interpretable. They segment foreground and background and estimate the foreground speed.

Table 1 reports the prediction losses, structural similarity, and the number of parameters for the evaluated models. It can be observed that our proposed model outperforms our baselines.

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2. Available from http://www.i6doc.com/en/.

Table 1: Prediction losses for Moving MNIST.

Model	L1	MSE	SSIM	# of params
Conv-PGP [8]	0.0323	0.0074	0.9025	32K
Our model	**0.0024**	**0.0002**	**0.9896**	2K
Our model without param	0.0059	0.0010	0.9737	**0**
VLN-ResNet [5]	0.0166	0.0009	0.9540	1.3M
VLN-LDC [9]	0.0126	0.0006	0.9686	1.4M

5 Conclusion

We proposed an end-to-end learnable neural network for motion segmentation that models foreground and background separately and predicts foreground motion by Frequency Domain Transformer Networks. The network estimates interpretable internal states using a hard-wired prediction-correction scheme. The basic method is highly computationally efficient and has zero parameters. We added a few trainable layers to optimize prediction for the specific dataset at hand. Experiments indicate that our method can solve the motion segmentation task in synthetic dataset. With far fewer parameters, our proposed architecture significantly outperforms the results of both VLN and Conv-PGP models on the tested dataset. The model with learnable parameters performs better than the model without parameters.

Acknowledgment This work was funded by grant BE 2556/16-1 (Research Unit FOR 2535 Anticipating Human Behavior) of the German Research Foundation (DFG).

References

[1] H. Farazi and S. Behnke. Frequency domain transformer networks for video prediction. In *ESANN*, 2019.

[2] K. Greff, A. Rasmus, M. Berglund, H. Valpola, and J. Schmidhuber. Tagger: Deep unsupervised perceptual grouping. In *NIPS*, 2016.

[3] J. Hsieh, B. Liu, D. Huang, L. Fei-Fei, and J. Niebles. Learning to decompose and disentangle representations for video prediction. In *ICLR*, 2018.

[4] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu. Spatial transformer networks. In *NIPS*, 2015.

[5] F. Cricri, X. Ni, M. Honkala, E. Aksu, and M. Gabbouj. Video ladder networks. *arXiv:1612.01756*, 2016.

[6] V. Michalski, R. Memisevic, and K. Konda. Modeling deep temporal dependencies with recurrent grammar cells. In *NIPS*, 2014.

[7] R. Memisevic. Learning to relate images. *IEEE Transactions on PAMI*, 2013.

[8] F. De Roos. Modeling spatiotemporal information with convolutional gated networks. Master's thesis, Chalmers University of Technology, 2016.

[9] N. Azizi, H. Farazi and S. Behnke. Location dependency in video prediction. In *ICANN*, 2018.

[10] A. Dave, O. Russakovsky, and D. Ramanan. Predictive-corrective networks for action detection. In *CVPR*, 2017.

[11] J. Hur and S. Roth. Iterative residual refinement for joint optical flow and occlusion estimation. In *CVPR*, 2019.