Dynamic Visual Sequence Prediction with Motion Flow Networks

Dinghuang Ji[1] Zheng Wei[1] Enrique Dunn[2] Jan Michael Frahm[1]

[1]The University of North Carolina at Chapel Hill
[2]Stevens Institute of Technology

Abstract

We target the problem of synthesizing future motion sequences from a temporally ordered set of input images. Previous methods tackled this problem in two manners: predicting the future image pixel values and predicting the dense time-space trajectory of pixels. Towards this end, generative encoder-decoder networks have been widely adopted in both kinds of methods. However, pixel prediction with these networks has been shown to suffer from blurry outputs, since images are generated from scratch and there is no explicit enforcement of visual coherency. Alternately, crisp details can be achieved by transferring pixels from the input image through dense trajectory predictions, but this process requires pre-computed motion fields for training, which limit the learning ability for the neural networks. To synthesize realistic movement of objects under weak supervision (without pre-computed dense motion fields), we propose two novel network structures. Our first network encodes the input images as feature maps, and uses a decoder network to predict the future pixel correspondences for a series of subsequent time steps. The attained correspondence fields are then used to synthesize future views. Our second network focuses on human-centered capture by augmenting our framework to include sparse pose estimates [30] to guide our dense correspondence prediction. Compared with state-of-the-art pixel generating and dense trajectories predicting networks, our model performs better on synthetic as well as on real-world human body movement sequences.

1. Introduction

Image-based motion prediction aims to generate plausible visualizations of the temporal evolution of an observed scene. In principle, a set of multiple images of the scene of interest may enable geometry-based view synthesis through direct prediction of the variation in scene content and/or viewing parameters (e.g. model-based rendering).

Figure 1. We predict multi-image motion flows to synthesize future image sequences given a partially observed motion.

However, the problem of direct appearance-based prediction of image motion is heavily ambiguous as the relationship between the scene and the observer is not uniquely defined. The problem becomes even more challenging when the scope of the desired visualization encompasses multiple time steps into the future. In this context, motion prediction can be seen as a pair of complementary problems: view synthesis and motion field estimation. View synthesis strives to render an image observation given partial specification of the scene contents and the observation parameters. Motion field estimation strives to determine dense pixel correspondences among a pair of image observations of a common scene. Given an input image and a motion field, it is straightforward to synthesize a novel image. Conversely, given an input image and a synthesized image, there is an abundance of methods to estimate the motion field. To the best of our knowledge, no supervised learning methods have been deployed to address the motion prediction problem by leveraging the complementary nature of these problems. In this paper, we attack the motion prediction problem within an image synthesis framework, so as to predict the motion flow and appearance simultaneously.

Predicting pixel values. View synthesis networks are naturally adopted to approach the visual prediction problem. To resolve motion ambiguity, Xue *et al*. [32] adopts a variational autoencoder framework to model the uncertainty of predicting the next state of a single input image. They propose a Cross Convolutional Network to encode image

and motion information as feature maps and convolutional kernels, respectively. The network directly outputs future image pixels, while a probabilistic model within the network makes it possible to sample and synthesize many possible future frames from a single input image. However, Zhou *et al*. [35] shows that this kind of model suffers from heavy blurriness when directly outputting pixels. Instead of predicting pixels, Walker *et al*. [26] adopt a variational autoencoder to generate a distribution of possible trajectories. They use the output of [29] as ground truth for dense pixel trajectories among the source and target images used to train their network. However, there is no evidence that the CNN network can improve upon the given ground truth dense trajectories, potentially imposing systematic biases into the prediction. In our proposed framework, we expect the network to learn the dense motion flows by minimizing the synthesis error through a weakly-supervised encoder-decoder architecture.

Increasing the predictive scope. Predicting images for more than one time step in the future has been previously addressed by Walker *et al*. [28] and Zhou *et al*. [36]. Walker *et al*. take an input image and predicts motion vectors with discretized directions and magnitudes. Recurrent networks are adopted to generate longer sequences. The method proposed in [36] generates future image sequences within a generative adversarial network (GAN), which has greatly improved the image generation quality compared to a baseline auto-encoder network. However, the GAN may suffer from systematic appearance artifacts correlated to the training set appearance distribution. We generate multiple output predictions through an iterative network that internally accumulates sequential pairwise pixel motion fields.

Modeling Scene Dynamics. Zhou *et al*. [35] propose "Appearance Flow" to learn dense pixel correspondences between different camera views under weak supervision, this method showed impressive success on static objects. However, predicting the motion of dynamic (and potentially non-rigid) objects is a heavily under-constrained problem. Directionally constrained correspondence prediction was recently addressed by Ji *et al*. [14] by learning the epipolar geometric constraints between two views and reducing the 2D flow search to a 1D search. Their experimental results outperform the traditional 2D appearance flow search [35]. However, for dynamic objects, no geometric clues have been adopted to assist the correspondences search. Along these lines, the convolutional pose machine(CPM) [30] is recently widely used to detect human body pose, this network is trained with large datasets of labeled human joint positions and achieves astonishing speed and accuracy on 2D human pose estimation. We develop a pair of image synthesis networks: one a general appearance-based predictor, the other a capture-specific pose-constrained predictor.

Our Contributions In this paper, we propose two motion flow-based view synthesis networks to tackle the visual prediction problem for dynamic scene content. The first network (MotionFlow) predicts 2D motion flows between multiple time steps, while the second network (PoseFlow) constrains the motion flows computation through domain-specific estimated directional priors. The novelty of our work can be summarized as:

- We propose the first weakly-supervised framework to model motion flow for the dynamic sequence synthesis problem.
- We incorporate sparse human body pose estimates to constrain dense motion flow prediction.

2. Related Works

Long range motion flow. Optical flow estimation among successive frames is mainly used to generate motion flows [2, 9, 22]. Brox *et al*. [4, 19] estimate optical flows simultaneously within multiple frames by adopting robust spatio-temporal regularization. Some long-range optical flow algorithms do not assume temporal smoothness. Wills and Belongie [31] estimate dense correspondences of image pairs using a layered representation initialized with sparse feature correspondences. Irani [13] describes linear subspace constraints for flow across multiple frames. Brand [3] applies a similar approach to non-rigid scenes. Sand and Teller [20] propose to represent video motion using a set of particles, which are optimized by measuring point-based matching along the particle trajectories and distortion between the particles.

Future prediction. Future prediction has been used in various tasks such as estimating the future trajectories of cars [27], pedestrians [16], or general objects [34] in images or videos. Given an observed image or a short video sequence, models have been proposed to predict a future motion field [18, 21, 28, 25]. Zhou *et al*. [37] frames the prediction problem as a binary selection task to determine the temporal sequence of two video clips. [24] trains a deep network to predict visual representations of future images with large amounts of unlabeled video data from the Internet. Different from our paper, this method predicts related future images instead of predicting object movements.

View synthesis with CNN. Recent methods for synthesizing novel views, objects, or scenes under diverse view variations have been boosted by the ability of Convolutional Neural Networks (CNNs) to function as image decoders. Hinton *et al*. [11] learned a hierarchy of capsules, computational units that locally transform their input, for generating small rotations to an input stereo pair. Dosovitskiy *et al*. [8] learned a generative CNN model to image of a chair with respected to given input graphics codes i.e. identity, pose, and lighting. Inspired by this paper, Tatarchenko *et al*. [23]

and Yang *et al*. [33] adopt a encoder-decoder network to implicitly learn graphics code from training image pairs or sequences. Tatarchenko *et al*. [23] proposed a approach to predict images and silhouettes without explicit decoupling of identity and pose. Yang *et al*. [33] applied input transformation to the learned pose units of source images to obtain desired target images, and apply recurrent network to enable synthesize sequences with large viewpoint difference.

Since the above methods generate new pixels from scratch and thus the synthesized results will tend to be blurry. Zhou *et al*. [35] propose to use the pixels of the input image as much as possible, by learning the pixel correspondences within given input images. This method can obtain synthesis with crisp texture and much less blurriness. However, since this method poses no constraints on the learned appearance flow, some of the generated synthesis has large texture distortions. Generative adversarial networks (GANs) have shown great promise for improving image generation quality [10]. GANs are composed of two parts, a generative model and a discriminative model, to be trained jointly. Some extensions have combined GAN structure with multi-scale laplacian pyramid to produce high-resolution generation results [7].

3. Our Approach

We address two main challenges in the learning-based prediction of extended motion from input images: 1) enhancing visual coherence, while simultaneously 2) reducing the supervision required for training. To this end, we generate future views with two motion flow networks (shown in Fig. 2 and 5) implemented with encoder-decoder networks. The core idea is to deploy an iterative predictive network to estimate dense correspondence fields across multiple time steps in the future. Since the direct output of the encoder-decoder network are motion fields, the synthesized views are comprised of pixels mapped from the input image instead of pixels directly synthesized by the decoder.

3.1. MotionFlowNet: Appearance Flow Estimation for Sequence Synthesis

The goal of an appearance flow network is to synthesize an output target image I_t by sampling pixels from an input source image I_s. The process of pixel sampling is guided by a dense 2D motion flow (e.g. pixel-wise displacement) field. The output of the network is a flow field $f = (f_x^{(i)}, f_y^{(i)})$, defined over the (i) pixels in the input image and yielding an image formation process of the form

$$g(I_s) = I_t(x^{(i)}, y^{(i)}) = I_s(x^{(i)} + f_x^{(i)}, y^{(i)} + f_y^{(i)}), \quad (1)$$

In general, learning pairwise correspondence fields requires a set of N source and target image pairs $\langle I_s, I_t \rangle^n \in \mathcal{D}$ are given during the training session. The learning is formalized as minimizing the pixel-wise reconstruction error (i.e. intensity difference): $\sum_{\langle I_s, I_t \rangle \in \mathcal{D}} \|I_t - g(I_s)\|_p$, where $\mathcal{D}$ is the set of training pairs, $g(.)$ refers to the motion-based image from the neural network whose weights we wish to estimate, $\|.\|_p$ denotes the L_p norm. Since the predicted motion fields are in sub-pixel coordinates, the synthesized view is obtained through bi-linear interpolation:

$$I_t^{(i)} = \sum_{q \in \mathcal{B}\left(x^{(i)}, y^{(i)}\right)} I_s^{(q)}(1 - |x^{(i)} - x^{(q)}|) \cdot (1 - |y^{(i)} - y^{(q)}|), \quad (2)$$

where $\mathcal{B}\left(x^{(i)}, y^{(i)}\right)$ denotes the set of four integer pixel positions bounding (i.e. top-left, top-right, bottom-left, bottom-right) the real-valued pixel coordinates of a given pixel $(x^{(i)}, y^{(i)})$, which is the corresponding positions for the ith pixel in I_t.

To generate multi-frame sequences, the decoder network outputs multiple 2D motion flows, and iteratively take pixels from the synthesized images to generate future images. Our training objective is based on pixel-wise prediction over all time steps for training sequences:

$$\sum_{k \in M, \cdots, N} \|I_k - g^{(k-M+1)}(I_{M-1})\|_2\} \quad (3)$$

In this formulation, for each motion sequence instance, we are given an ordered ground truth image set $\{I_n\}$, partitioned into input motion observations and target image predictions to be used within our supervised learning framework. More specifically, $I_{1 \geq j < M}$ are used as input images depicting the start of a motion sequence, and we aim to predict a sequence of images corresponding to $I_{M \geq k \leq N}$, which depicting the observation at immediately subsequent timesteps. In our notation, $g^{(n)}$ refers to the output image associated with the accumulated n-th motion flow defined over the last available image observation I_{M-1} of the input motion. Accordingly, the direct output of our encoder-decoder network is a set of $N - M$ total predicted pixel motion flows between successive timesteps and having the same pixel dimension as the input imagery.

3.2. PoseFlowNet: Appearance Flows with Constrained Directions

Motion flow estimation on dynamic objects is a challenging problem, as there are no geometric constraints (like epipolar constraints learned in [14]) that can be leveraged to reduce the motion flow search space. Hence, the correspondence search space for each pixel, into the next frame, spans the whole image. To ease the correspondence problem, we focus on human motion sequences and adopt an off-the-shelf pose estimator [5] to reliably determine subject landmarks across our input motion image sequence. We

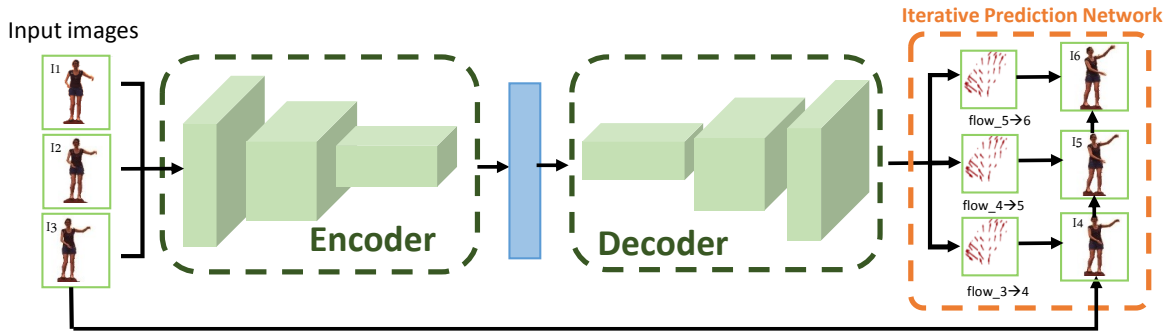

Figure 2. MoFlow Network. In this example network, three input images are concatenated as input for encoder network, the decoder network output three motion flows. Pixels of input image 3 are borrowed with learned motion flows to synthesize image in future timesteps so as to minimize the pixel reconstruction errors. The network iteratively borrows pixels from synthesized images to generate future images.

then leverage these detected sparse joint location estimates to 1) make predictions on future pose configurations, and 2) enforce consistency of the estimated dense motion field to these predicted poses. In practice, the geometry-based generalization of sparse local motion estimates is not robust to fine-grain appearance-based cues and leads to strong visual artifacts. Accordingly, the computation of motion flow prediction is decoupled into a directional component estimated from sparse pose predictions and a magnitude component that is estimated from input image observation

Feature Guided Correspondence Computations. The pose estimator outputs sparse joint positions (18 points) for each detected person in the image (shown in Fig. 3(a)(b)). If the subject shows up in profile view, some joint points will be missed. We fill these null values with symmetric joint positions. The human body movements are complex as each local part (left arm, right leg *etc*.) moves independently. Beier *et al*. [1] propose a method to compute how points around line segments move accordingly given line segment movements. With this method, given input human poses, we can obtain dense motion flow between consecutive frames.

Figure 3. (a)(b) Pose estimation results for images within a motion sequence. (c) Computed motion flow with method [1].

In a 2D image (Fig. 4 left), the coordinate mapping of a point X on a line segment MN are represented as (u, v), which are computed by Eq. (4),(5). If in the next time step (Fig. 4 right), position of MN changed to $M^{'}N^{'}$, then the new position of point X would be $X^{'}$ which is computed by Eq. (6).

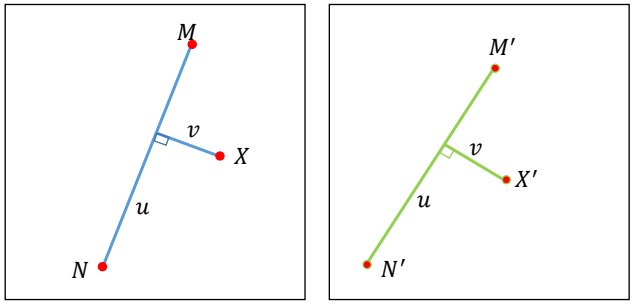

Figure 4. Between left and right image, endpoints of line segment MN are changed to $M^{'}N^{'}$.

$$u = \frac{(X - M) \cdot (N - M)}{\|N - M\|^2} \tag{4}$$

$$v = \frac{(X - M) \cdot Perpendicular(N - M)}{\|N - M\|} \tag{5}$$

$$X^{'} = M^{'} + u \cdot (N^{'} - M^{'}) + v \cdot \frac{v \cdot Perpendicular(N^{'} - M^{'})}{\|N^{'} - M^{'}\|} \tag{6}$$

Here function $Perpendicular(N - M)$ obtains perpendicular vector to $N - M$, which has the same length as $N - M$. In this coordinate system, value u defines the position along the line, and v is the distance from pixel X to the line MN. The value range of u is 0 to 1 as pixel moves from M to N, and is less than 0 and greater than 1 outside that range. The value for v is the perpendicular proportional distance from pixel X to the line MN. If there is just one line pair, the transformation of the whole image

proceeds as Eq. (4),(5),(6). Since the human body is composed of multiple line segments (we define 14 local parts on the human body.), pixels should naturally move in compliance to its nearest line segment. Since the assignment of pixels to local parts is unknown, a weighting strategy of the coordinate transformations for each line is performed, for each line segment a position $X_i' = (u_i, v_i)$ is computed for each pixel X. To calculate the weighted average of those displacements we follow

$$\begin{aligned} w_i =& \frac{1}{(a + dist)^b} \\ X^{'} =& X + \sum_i \frac{w_i}{\sum_i w_i} * (X_i^{'} - X) \end{aligned} \tag{7}$$

Here a is a constant to prevent illegal division, variable b decides the displacement of a pixel along with different line segments. If b is large, every pixel will be affected only by the line nearest to it. If b is zero, each pixel will be affected by all lines equally. We set $b = 1.5$ in all experiments. A sample motion flow field is visualized in Fig. 3(c) which highlights the motion vectors between Fig. 3(a) and Fig. 3(b). It can be observed that motion estimates make no distinction between pixel on a moving limb and nearby pixels not belonging to the limb (e.g. pixels on the torso). We address this challenge by estimating a per-pixel motion magnitude based on the appearance of the input sequence.

Sequence Synthesis with Constrained Correspondences Search. We propose the PoseFlow network (shown in Fig. 5), which takes images along with detected poses as input. Input poses are fed to a pose prediction network to predict future poses, and generate the dense motion flow fields (with Eq. (4),(5) and (6)) from the predictions. The pose prediction network is composed of four fully connected layers and outputs pose offsets compared to previous frame. Detailed network structure is listed in supplemental materials.

The encoder-decoder network has same configuration as MotionFlowNet. However, instead of predicting 2D motion flows, the output of our decoder is the magnitude of motion flows, the final output of the network is the multiplication of the predicted motion flows and the magnitude fields. By learning appropriate magnitude fields, some mistakenly computed motion flows can be mitigated. For example, in Fig. 3(c), we observe motion vectors on torso above the right arm, caused by the proximity to the moving right arm. However, between Fig. 3(a) and Fig. 3(b), pixels on the torso are actually not moved. We expect the network optimize magnitudes so as to mitigate this problem, i.e. magnitudes learned on torsos would be near zeros.

3.3. Implementation details

We trained the network parameters using the ADAM optimization method [15]. For different datasets, the input sequence may contain different number of images to reduce the motion prediction ambiguity. For our base implementation we use three stacked images as input motion observations and output three predicted images as a single stack.

4. Experiments

Datasets. We adopt two datasets to verify our method, the synthetic Sprites dataset and real image dataset human3.6M [12, 6].

Sprites Dataset. This dataset consists of 672 unique characters, and for each character there are 5 rigid-body movements from 4 different viewpoints. Each animation ranges from 6 to 13 frames. The image contains single character, with original pixel size of 60 $\times$ 60, we resize it to 224 $\times$ 224 to fit our network architecture. In our experiments, our training and testing sequences have length 6. For animations longer than 6 frames, we take sequences with 5 overlap frames. For example, 8 frames animation can generate 3 subsequences with length 6, with frame indices 1-6,2-7,and 3-8. We use 600 characters for training, 72 for testing and that collect 12,642 sequences for training, and 2000 sequences for testing.

Human3.6M Dataset. Human3.6M dataset [12, 6] is collected for tasks like 3D reconstruction of body movements, motion recognition and semantic segmentation. It's acquired by recording the performance of 5 female and 5 male subjects, under 4 different viewpoints. Overall, it has 3.6 million 3D human poses and corresponding images, consisting of 17 scenarios (discussion, smoking, posing, talking on the phone *etc*.). Since the subject number is very limited, we adopt 9 of them for training and 1 for testing. Since "Posing" sequences contains variety of motions, we generate the training and testing sequences from them. With each video, we take 6 consecutive frames as a sequence, the selected sequences have no overlaps, which gives us 10,125 training sequences and 1,600 testing sequences.

Baseline Methods. We compare our methods with a state-of-the-art pixel generating based sequence prediction method **ECCV16** [36], which adopt a generative adversarial network to improve the image qualities. The authors of **ECCV16** kindly trained the model for us with the same datasets as our experiments. To evaluate the effectiveness of our PoseFlow network, we synthesize the predicted images trough the method described in [1] (**SIG92**) using the pose parameters estimated on the ground truth imagery.

Qualitative Evaluations. To illustrate the effectiveness of our method, Figure 6 plots the synthesized images from the trained network and compare with baseline methods. The third row of Fig. 6 shows see some artifacts (high-

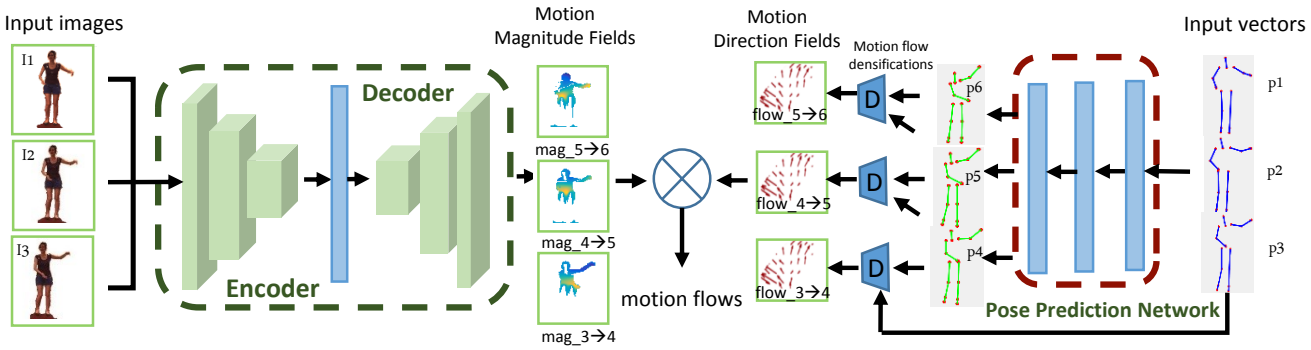

Figure 5. PoseFlow network. Left part of the network output pixel-wise predictions of motion flow magnitude, and the right part is a fully connected network predicting the future sparse poses that are densified into directional flow fields.

lighted in red) generated with [1], this is caused by inaccurate motion flows for torso pixels. Our network can learn appropriate magnitudes along the motion directions to mitigate this artifact. Compared with **MoFlowNet**, **PoseFlowNet** has less blurriness (highlighted in green boxes), and more accurate shape deformations (shown in Table 3.

In Fig. 6, we compare the synthesized images with baseline methods. **ECCV16** outputs a sequence of 64×64 images, we resize them to be 224×224. While poses can be reasonably predicted, the synthesized appearance can differ strongly from the input image. This can be attributed to the GAN network mimicking the test results by sampling from training samples, instead of borrowing pixels specifically from the input test images. Since Sprites dataset contains synthetic Emoji characters, pose detector cannot detect poses from them, so we only compare our **MoFlowNet** with **ECCV16** (shown in Fig. 7). Again, **ECCV16** can generate correct poses as the groundtruth, however the color is distorted, while our method generates more similar and crisp appearance, especially on the static regions.

Quantitative Evaluations. As an error metric, we use the mean squared error (MSE) between the synthesized output and ground truth summed over all pixels. In Tab. 1, we show the MSE for synthesized 3 frames tested on human3.6m and Sprites dataset. We can see for Human3.6M dataset, the MoFlowNet and PoseFlowNet achieve on par synthesis errors along the sequences, and outperform the baseline methods by big margins. **MoFlowNet** reduce the synthesis errors by half than **ECCV16** on Sprites dataset.

We adopt CPM [5] on synthesized images and their groundtruth to compare the estimated pose difference in terms of relative angle (RelAng) and lengths (RelLen). To measure the accuracy of our motion predictions, we compare against the baseline motion for points sampled along the straight-line segments detected on subsequent synthesized and ground truth images (shown in Table 2).

To highlight the effectiveness of **PoseFlowNet** decoupled motion flow estimation, we compare against the geometry-only flow estimate (**PosePred**) attained from densifying our sparse pose motion predictions. Table. 3 shows how **PoseFlowNet** consistently outperforms **ECCV16**, MoFlowNet and the geometry-based motion field method.

To verify how the length of input sequences affect the synthesis process, we adopt input length $1 - 4$ on Sprites dataset and show the first two prediction errors (in Table. 4)

To compare with the flow generating network, we take the public model trained for [28] and predict the next frame given input images ([28] test with one image, PoseFlowNet test with the same image and its two previous frames, since

Method	Frame 4	Frame 5	Frame 6
SIG92	235.6	561.2	932.5
ECCV16	4602.2	4737.9	4993.1
MoFlowNet	185.1	380.5	850.5
PoseFlowNet	197.6	365.1	796.1
ECCV16	53.9320	54.1431	54.8665
MoFlowNet	27.0103	27.7398	27.9549

Table 1. MSE testing error for different frames in human3.6m (top four rows) and Sprites (bottom two rows) dataset.

Method	Frame 4	Frame 5	Frame 6
PosePred	3.59 – 3.55	5.72 – 4.23	6.66 – 5.33
ECCV16	22.78 – 15.20	25.67 – 13.17	33.32 – 18.15
MoFlowNet	1.91 – 3.39	3.90 – 5.26	5.03 – 4.17
PoseFlowNet	**1.54 – 2.84**	**2.11 – 3.23**	**4.54 – 4.32**

Table 2. End positions – Motion flow direction prediction error for different frames in human3.6m dataset. The values are in the unit of pixels and degrees.

Method	Frame 4	Frame 5	Frame 6
PosePred	4.82 – 2.11	5.31 – 4.06	5.91 – 6.39
ECCV16	26.22 – 9.51	21.91 – 8.08	20.08 – 7.24
MoFlowNet	3.47 – 1.51	5.29 – 1.93	7.38 – 2.40
PoseFlowNet	**2.78 – 1.32**	**3.93 – 1.69**	**4.82 – 1.88**

Table 3. RelAng – RelLen testing error for different frames in human3.6m dataset. The values are in the unit of degrees and pixels.

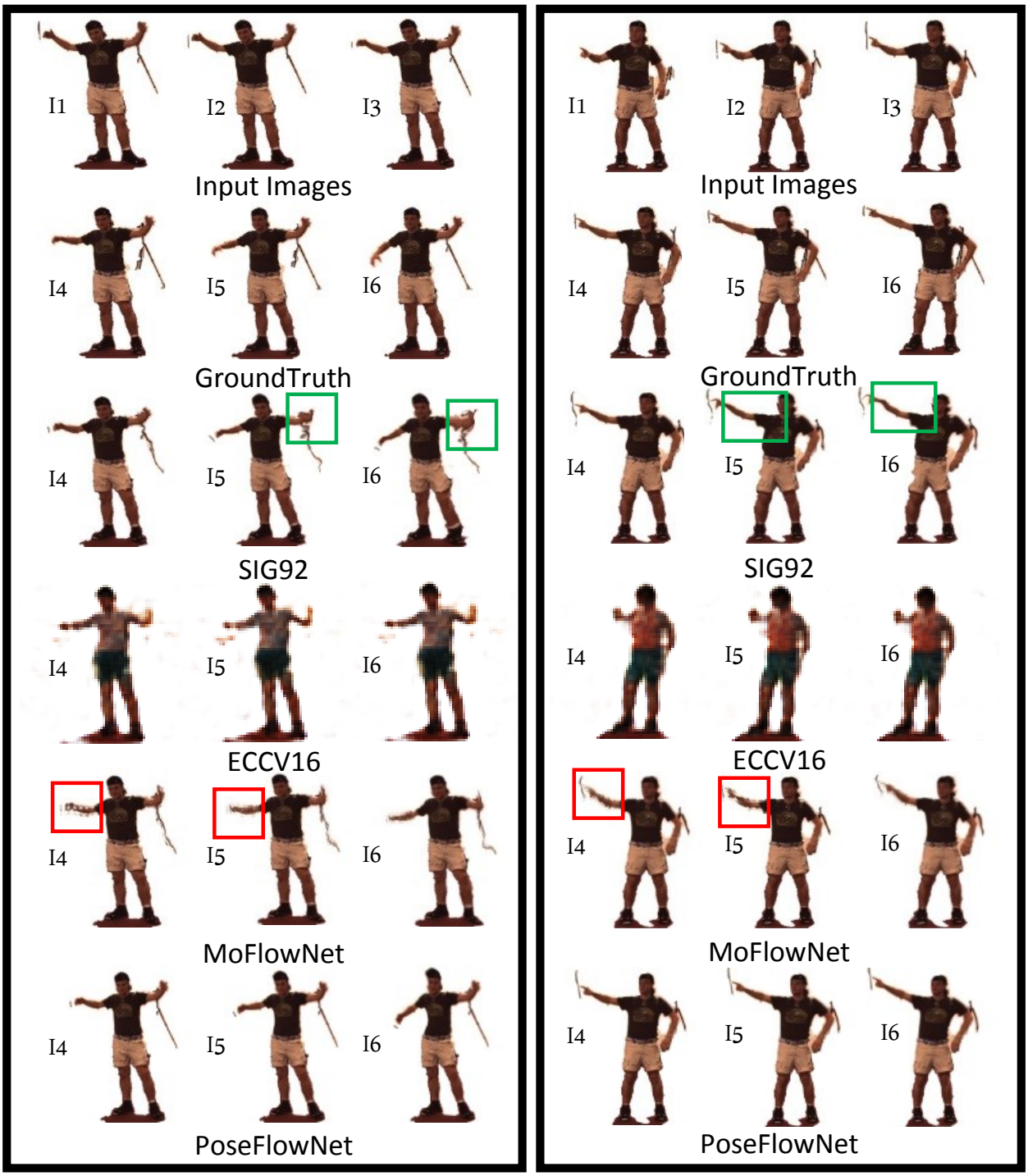

Figure 6. Two testing sequences for human3.6m dataset, compare results generated by **SIG92**, **ECCV16**, **MoFlowNet** and **PoseFlowNet**.

our method require three images as input). The public model only predicts the motion flow of the input image, we visualize the motion flows generated by [28] and PoseFlowNet. We adopt the optical flow method proposed by [17] as ground-truth. The red boxes in Fig. 8 show our flow direction is closer to the ground-truth. By measuring the direction error on non-white pixels, within the test set, PoseFlowNet and [28] achieve 6.3 and 26.8 degree errors.

PoseFlowNet learns magnitude field, which acts like masks. To verify the effectiveness of learned magnitude

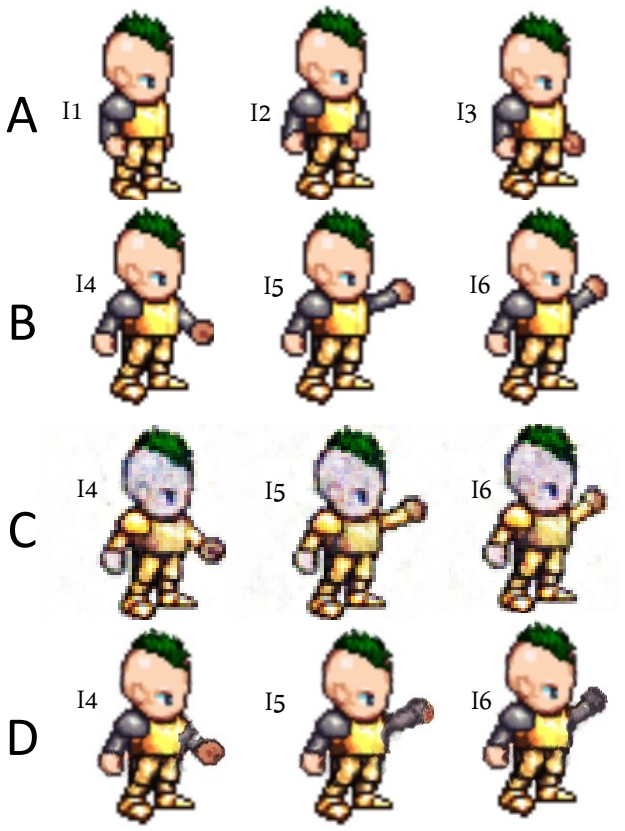

Figure 7. Testing sequences for Sprites dataset (Row A: input frames, row B: ground truth output frames), and compare results generated by ECCV16 (row C) and MoFlowNet (row D).

Input images #	First Prediction	Second Prediction
1	60.2	73.5
2	35.2	41.8
3	27.7	28.0
4	**23.5**	**25.4**

Table 4. MoFlowNet testing errors with different input images for frame 5 and 6 on Sprites dataset.

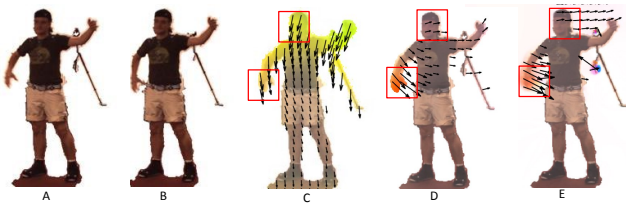

Figure 8. Motion flow prediction evaluation (A: input image, B: next frame, C: Flow by [28], D: Flow by PoseFlowNet, E: Groundtruth flow)

fields, we compare with the network that fills masks with all 1s. From Fig. 9 C, we can see that without learning magnitude fields, the synthesized images (highlighted in green boxes) will have severe distortions. PoseFlowNet prevents pixels from moving into the wrong direction with the help of the learned magnitude fields.

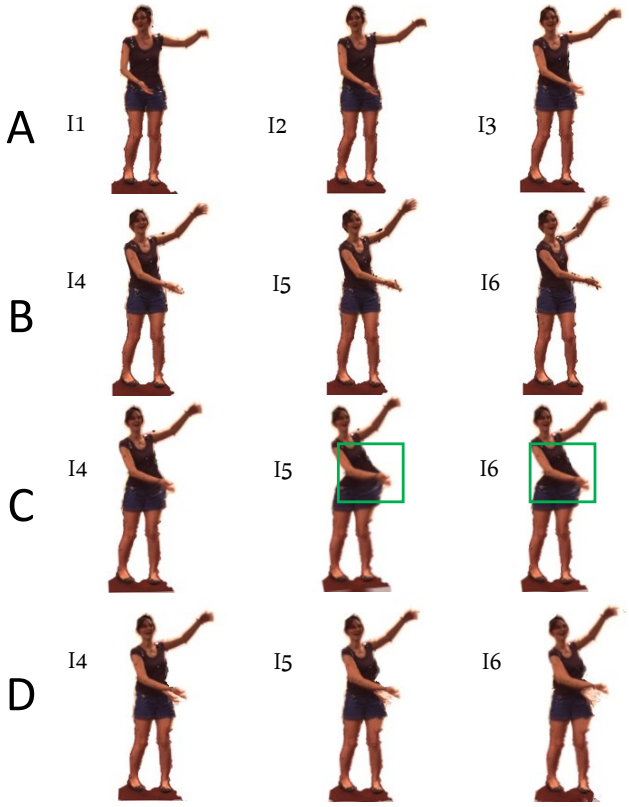

Figure 9. One sample test sequence for Human3.6M dataset (Row A: input frames, row B: ground truth output frames), and compare results generated PoseFlowNet without learning the magnitude field (row C) and PoseFlowNet (row D).

5. Conclusions

Our MoFlowNet introduces the auto-encoder framework to the dynamic-object motion prediction problem. In doing so we have reduced supervisory requirements of dense flow-based synthesis methods and augmented the scope of their prediction to encompass multiple frames into the future. Conversely, our PoseFlowNet focuses on human capture scenarios and introduces a framework that constraints the search space by enforcing spatio-temporal pose coherency and robustifying these estimates through a learned appearance-based preponderance. Moreover, by decoupling these complementary problem aspects into an hybrid neural network, we have outperformed the current state of the art in challenging synthetic and real-capture datasets. Future work includes, generalizing our 2D motion model (which does not explicitly model out-of-plane rotation) to include full 3D skeletal motion constraints and refine the model on images with dynamic background where current methods failed to generate satisfying results.

References

[1] T. Beier and S. Neely. Feature-based image metamorphosis. In *SigGraph*, 1992.

[2] M. Black and P. Anandan. Robust dynamic motion estimation over time. *CVPR*, 1991.

[3] M. Brand. Morphable 3d models from video. *CVPR*, 2001.

[4] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. *ECCV*, 2014.

[5] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. *arXiv preprint arXiv:1611.08050*, 2016.

[6] C. S. Catalin Ionescu, Fuxin Li. Latent structured models for human pose estimation. In *International Conference on Computer Vision*, 2011.

[7] Denton, E. L, C. Soumith, S. Arthur, and F. Rob. Deep generative image models using a laplacian pyramid of adversarial networks. In *Advances in Neural Information Processing Systems 28*, pages 1486–1494. Curran Associates, Inc., 2015.

[8] A. Dosovitskiy, J. Springenberg, and T. Brox. Learning to generate chairs with convolutional neural networks. *: IEEE International Conference on Computer Vision and Pattern Recognition.*

[9] M. Elad and A. Feuer. Recursive optical flow estimation adaptive filtering approach. *Visual Communication and Image Representation*, 1998.

[10] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. 2014.

[11] G. Hinton, A. Krizhevsky, and S. Wang. Transforming auto-encoders. *Artificial Neural Networks and Machine Learning-ICANN*.

[12] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1325–1339, jul 2014.

[13] M. Irani. Multi-frame optical flow estimation using subspace constraints. *ICCV*, 1999.

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

[15] D. Kingma, J. Ba, and G. Gamow. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

[16] K. Kitani, B. Ziebart, J. Bagnell, and M. Hebert. Activity forecasting. *ECCV*, 2012.

[17] C. Liu. Beyond pixels: Exploring new representations and applications for motion analysis. 2009.

[18] C. Liu, J. Yuen, and A. Torralba. Sift flow: Dense correspondence across scenes and its applications. 2011.

[19] N. Papenberg, A. Bruhn, T. Brox, S. Didas, and J. Weickert. Highly accurate optic flow computation with theoretically justified warping. *IJCV*, 2006.

[20] S. T. Peter Sand. Particle video: Long-range motion estimation using point trajectories. 2006.

[21] S. L. Pintea, J. C. van Gemert, and A. W. Smeulders. Dejavu: Motion prediction in static images. 2014.

[22] J. Shi and J. Malik. Motion segmentation and tracking using normalized cuts. *ICCV*, 1998.

[23] M. Tatarchenko, A. Dosovitskiy, and A. Brox. Single-view to multi-view: Reconstructing unseen views with a convolutional network. *arXiv preprint arXiv:1511.06702*.

[24] C. Vondrick, H. Pirsiavash, and A. Torralba. Anticipating the future by watching unlabeled video. *CoRR*, 2015.

[25] J. Walker, C. Doersch, A. Gupta, and M. Hebert. . an uncertain future: Forecasting from static images using variational autoencoders. *ECCV*, 2016.

[26] J. Walker, C. Doersch, A. Gupta, and M. Hebert. An uncertain future: Forecasting from static images using variational autoencoders. In *European Conference on Computer Vision*, 2016.

[27] J. Walker, A. Gupta, and M. Hebert. Patch to the future: Unsupervised visual prediction. *CVPR*, 2014.

[28] J. Walker, A. Gupta, and M. Hebert. Dense optical flow prediction from a static image. *ICCV*, 2015.

[29] H. Wang and C. Schmid. Action recognition with improved trajectories. In *IEEE International Conference on Computer Vision*, Sydney, Australia, 2013.

[30] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. Convolutional pose machines. In *CVPR*, 2016.

[31] J. Wills and S. Belongie. A feature-based approach for determining dense long range correspondences. *ECCV*, 2004.

[32] T. Xue, J. Wu, K. L. Bouman, and W. T. Freeman. Visual dynamics: Probabilistic future frame synthesis via cross convolutional networks. In *NIPS*, 2016.

[33] J. Yang, S. Reed, M. Yang, and H. Lee. Weakly-supervised disentangling with recurrent transformations for 3d view synthesis. *Advances in Neural Information Processing Systems.*

[34] J. Yuen and A. Torralba. A data-driven approach for event prediction. *ECCV*, 2010.

[35] T. Zhou, S. Tulsiani, W. Sun, J. Malik, and A. A. Efros. View synthesis by appearance flow. *ECCV*, 2016.

[36] Y. Zhou and T. Berg. Learning temporal transformations from time-lapse videos. *ECCV*, 2016.

[37] Y. Zhou and T. Berg. Temporal perception and prediction in ego-centric video. *ECCV*, 2016.