From Here to There:
Video Inbetweening Using Direct 3D Convolutions

2019/06/03 Mizuki Matsubara

はじめに

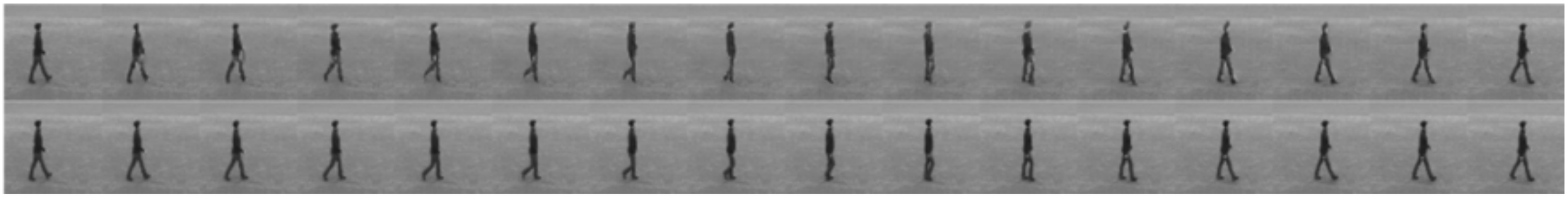
- 論文紹介：

  - 2019年5月27日にarxivに投稿

  - URL: https://arxiv.org/pdf/1905.10240.pdf

  - Author: Google Researchの人

  - 最初と最後の画像を入力として、その間の動画を作る (Youtube等みつからず..)



- 論文選択理由：

  先週のMirrorGAN (Text2Image)と組み合わせれば、ラノベ2アニメが作れる！？

  *ラノベが好きなわけではありません

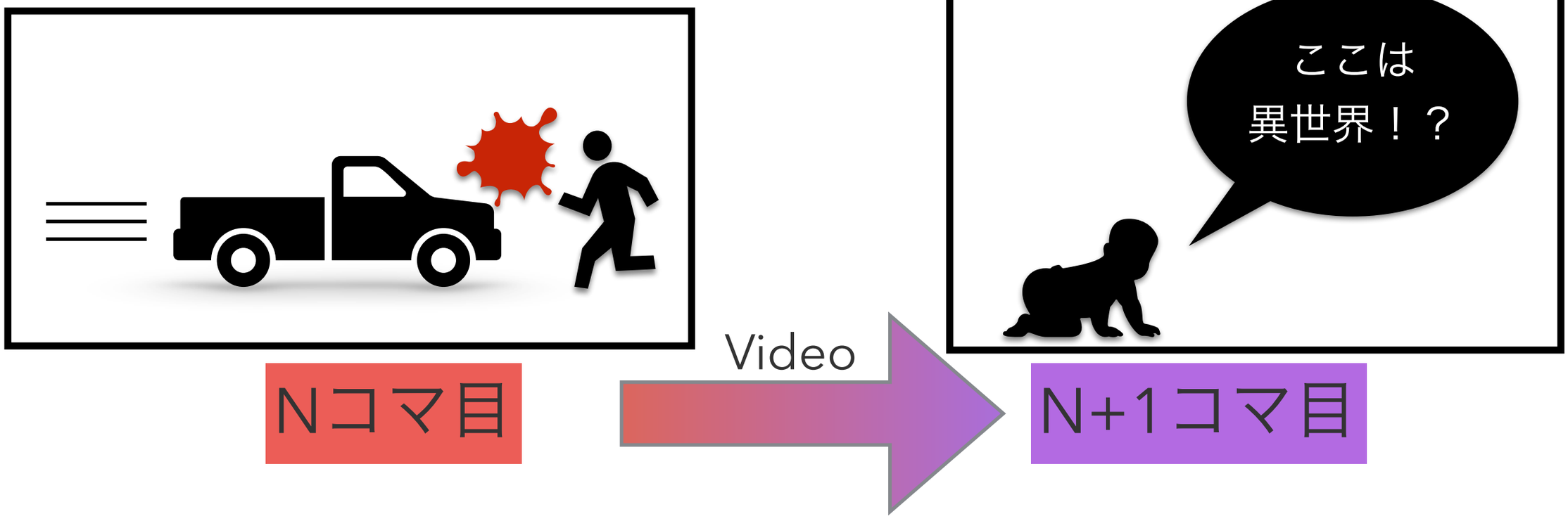
Li, Yunpeng, Dominik Roblek, and Marco Tagliasacchi. "From Here to There: Video Inbetweening Using Direct 3D Convolutions." *arXiv preprint arXiv:1905.10240* (2019).

背景

- Motivation: 漫画2アニメ (将来的に)

  連続するニコマを入力として、その間の動画を生成すればアニメができる

  不完全な動画でも、アニメ制作の補助になる可能性もある



連続するニコマを入力⇒Video Inbetweeningタスク

主要な動画生成技術

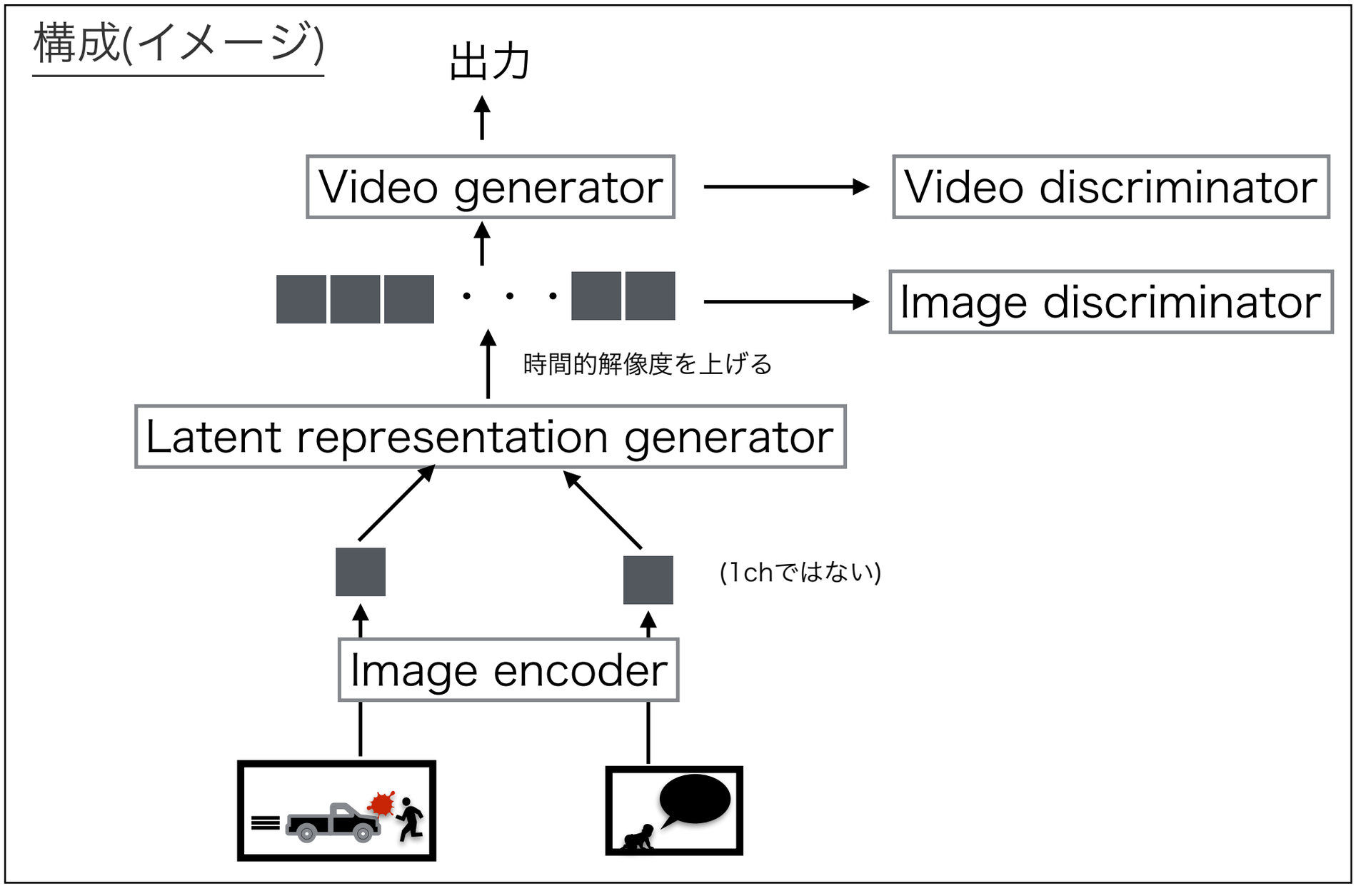
論文のアーキテクチャの概要

本論文のポイント：

latent representation (訳:潜在表現)を作ること

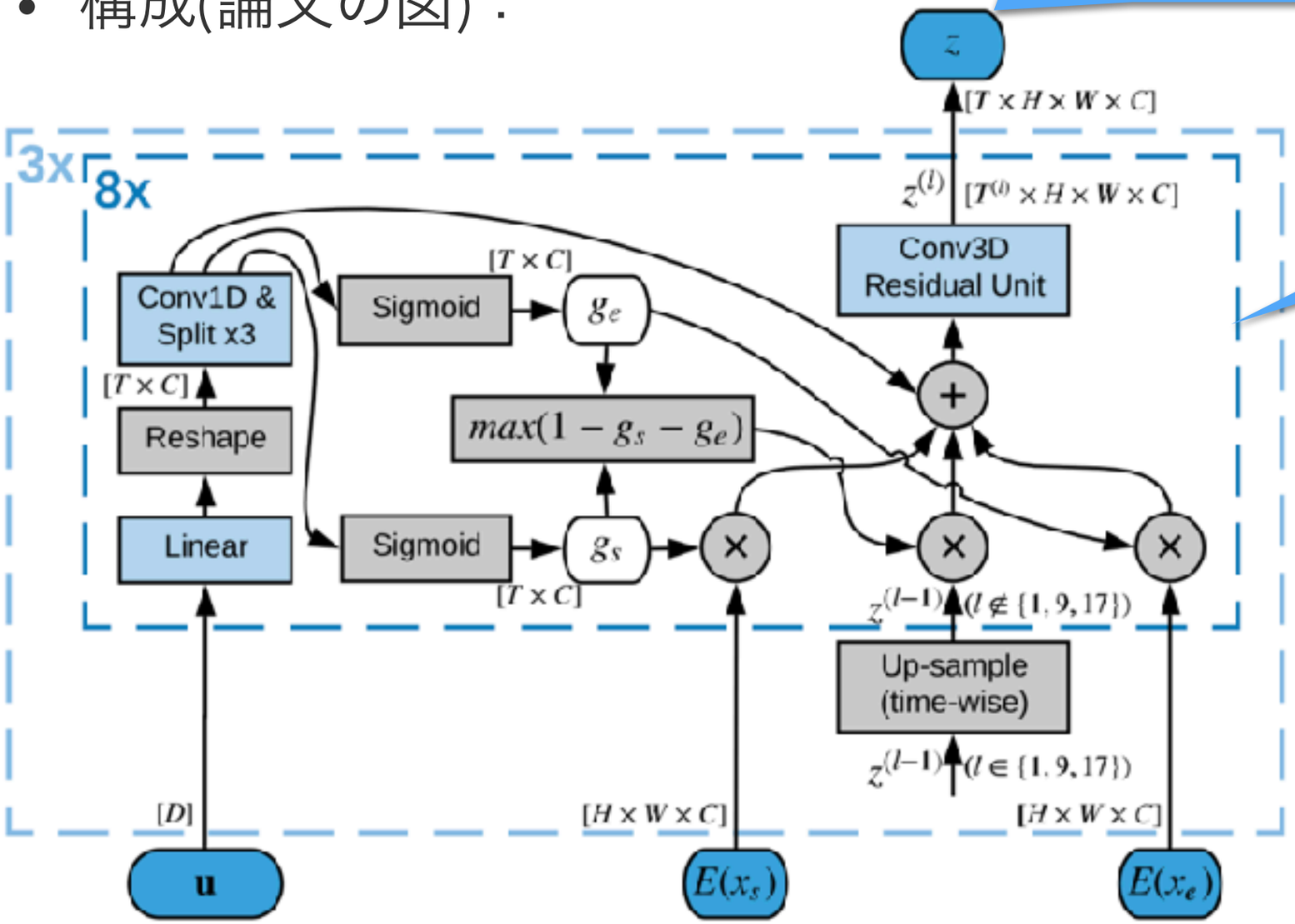
⇨ 入力2フレームの時間的解像度を上げるもの

構成(イメージ)



提案手法

- 構成(論文の図)：



Video generator (7層の3D Conv.)

LR generator (内枠)

外枠は演算の工夫
Coarse-to-fine generation

start frame xs

end frame xe

video (xs , x^1 , . . . , x^T −2 , xe)

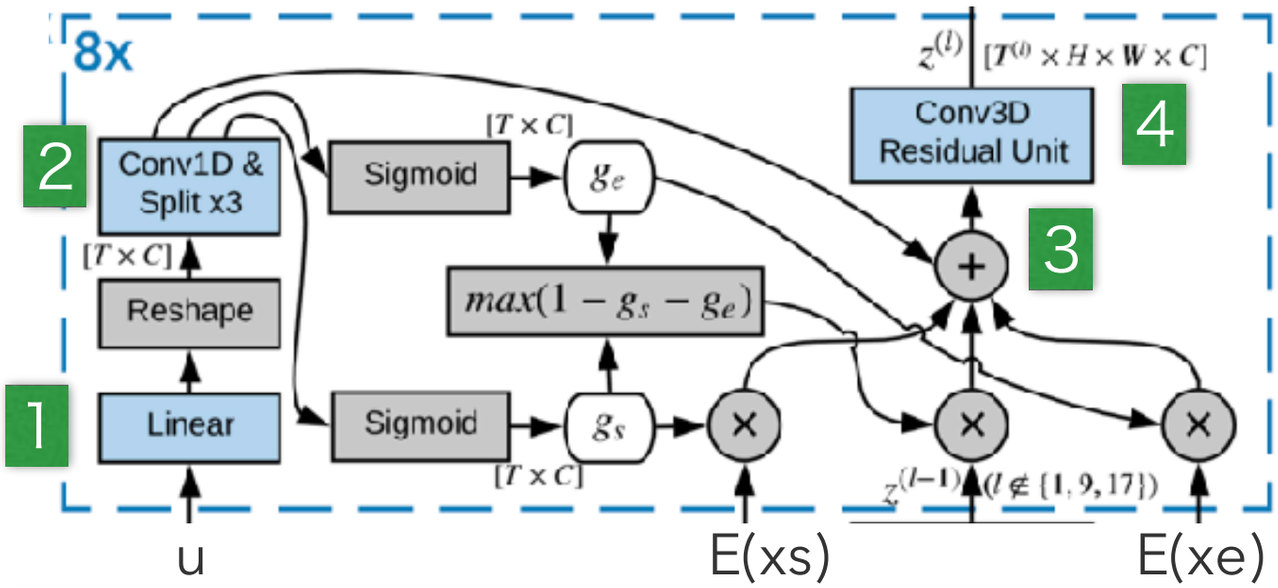
Gaussian noise vector u ∈ RD

T(出力のframe数) =16

D(ノイズのlength)=128

Image encoder (6層の2D Conv.)

Latent representation generator



Residual unit
3D convで構成

L個中のl番目の処理 (×8の方の四角)　　{A, k, b}はlearning parameter

1. ノイズを線形変換 u^(l) = Au + b
2. start, endそれぞれのゲートfuncであるgsとgeを作成

$$g_s^{(l)} = \sigma(u^{(l)} * k_s^{(l)} + b_s^{(l)}), \quad (2)$$
$$g_e^{(l)} = \sigma(u^{(l)} * k_e^{(l)} + b_e^{(l)}), \quad (3)$$

kは畳み込み層
σはシグモイド関数

3. l-1番目の中間出力とE(xs)とE(xe)を結合

$$z_{\text{in}}^{(l)} = g_s^{(l)} \cdot E(x_s) + g_s^{(l)} \cdot E(x_e) + \max(0, 1 - g_s^{(l)} - g_e^{(l)}) \cdot z^{(l-1)} + n^{(l)}, \quad (4)$$

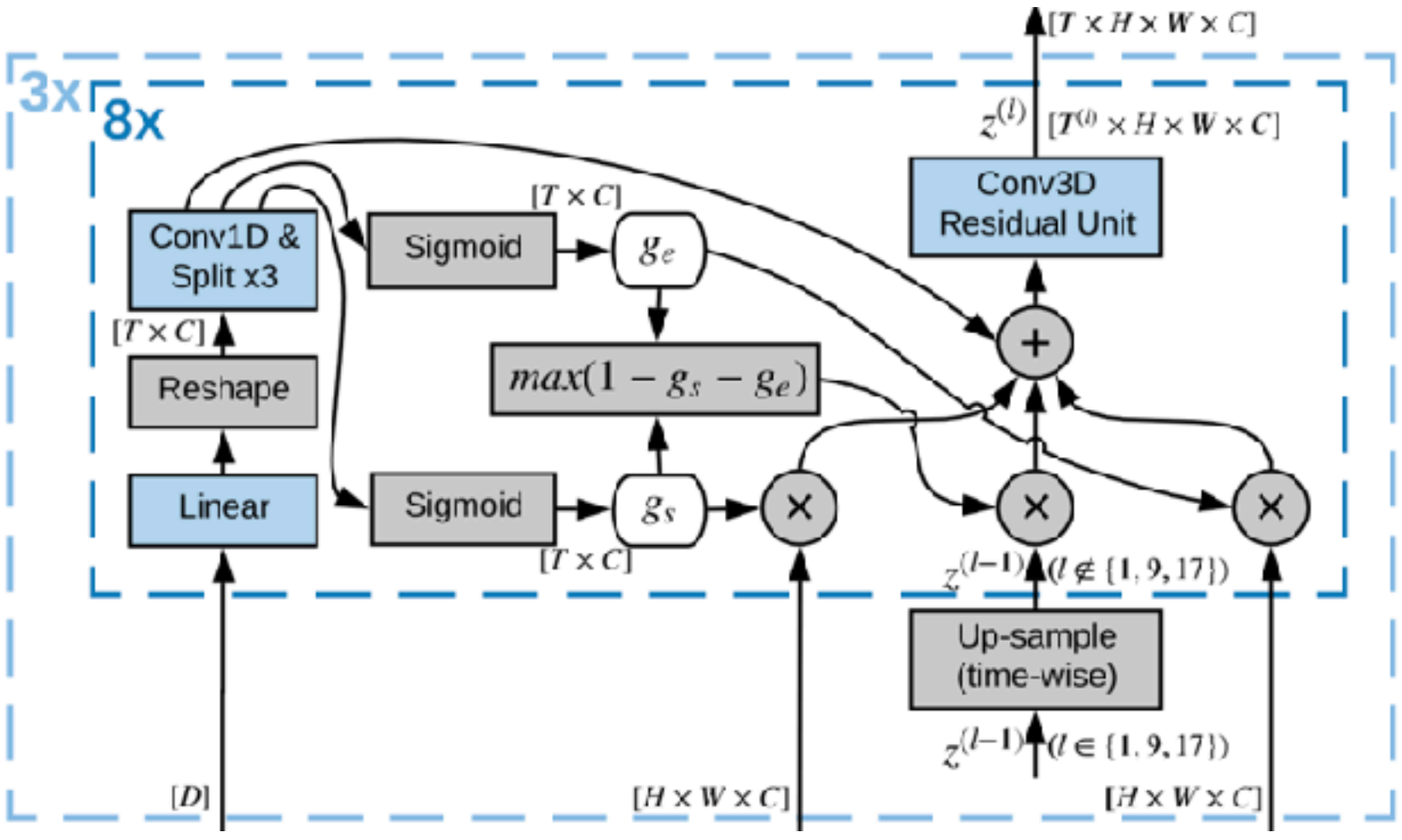
4. residual blockにより一つ前(l - 1番目)の中間出力を改良

$$z^{(l)} = h(z^{(l-1)} + h(z_{\text{in}}^{(l)} * k_1^{(l)} + b_1^{(l)}) * k_2^{(l)} + b_2^{(l)}), \quad (5)$$

hはReLU

⇒ LSTMのようなもの。入力と前の層の出力を見るネットワーク

Coarse-to-fine generation



外枠は演算の工夫
Coarse-to-fine generation

z^(l)の次元はT(フレーム数)にしとくのではない

計算コストを抑えるために

T/4⇒T/2⇒T/1と変化させている

Loss function

- end-to-endで学習

$$\min_{D_V} : \mathcal{L}(D_V) = \mathbb{E}_{(X,\hat{X})}\left[-\log D_V(X) - \log(1 - D_V(\hat{X}))\right] \tag{7}$$

$$\min_{D_I} : \mathcal{L}(D_I) = \mathbb{E}_{(X,\hat{X})}\left[\frac{1}{T-2}\sum_{i=1}^{T-2}\left[-\log D_I(x_i) - \log(1 - D_I(\hat{x}_i))\right]\right] \tag{8}$$

$$\min_{G=\{E,G_Z,G_V\}} : \mathcal{L}(G) = \mathbb{E}_{(X,\hat{X})}\left[-\log D_V(\hat{X}) - \frac{1}{T-2}\sum_{i=1}^{T-2}\log D_I(\hat{x}_i)\right] \tag{9}$$

Dv(X)・・・Video discriminator

DI(X)・・・Image discriminator

G・・・generator側全体

Experiment

- 使用したDataset

  BAIR robot pushing [10], KTH Action Database [34], and UCF101 Action Recognition Data Set [36]
- 比較方法

Fréchet video distance (FVD) ⇒画像の自然さを評価

SSIM ⇒元の動画との一致度 (既存手法との比較用)

実験結果

Table 1: We report the mean FVD for both the full model and two baselines, averaged over all 10 training runs with 100 stochastic generations each run, and the corresponding 95% confidence intervals. A lower value of the FVD corresponds to higher quality of the generated videos.

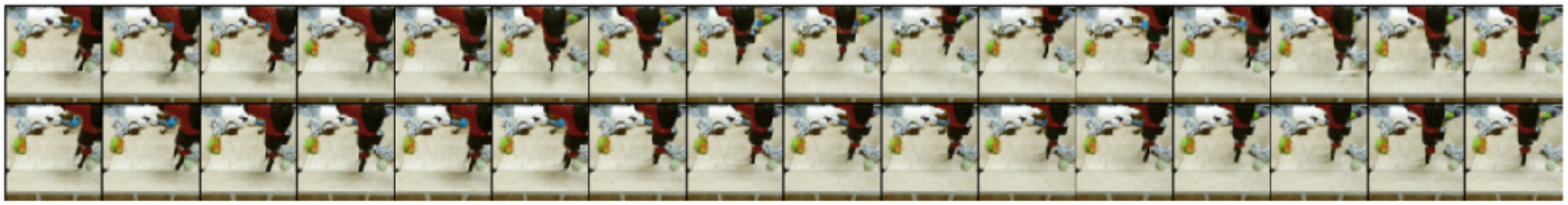
	BAIR	KTH	UCF101
Full model	152 [144, 160]	153 [148, 158]	424 [411, 438]
- w/o fusion	175 [166, 184]	171 [163, 180]	463 [453, 474]
- Naïve	702 [551, 895]	346.1 [328, 361]	1101 [1070, 1130]

提案処理
提案処理 - gate
提案処理 - LR

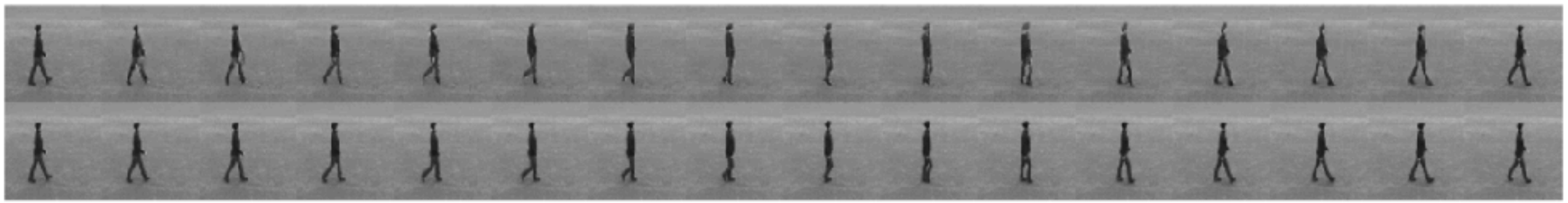
⇒ 提案処理のアーキテクチャが良いことがわかる

結果: 出力動画の例

BAIR:



KTH:



UCF101:

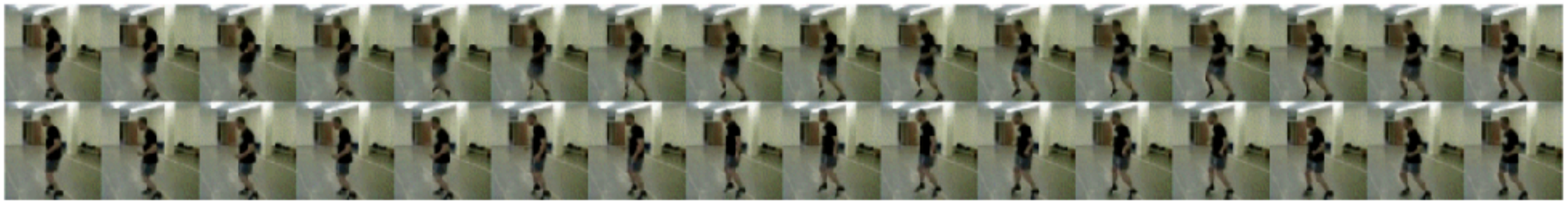


Figure 2: Examples of videos generated with the proposed model. For each of the three datasets, the top row represents the generated video sequences, the bottom row the original video from which the key frames are sampled.

実験結果 (既存手法との比較)

Table 3: Average SSIM of our model using direct 3D convolution and alternative methods based on RNN (SDVI) or optical flow (SepConv and SuperSloMo). Higher is better. Note the difference in setup: our model spans a time base twice as long as the others. The SSIM for each test example is computed on the best sequence out of 100 stochastic generations, as in [4, 9, 23, 49]. We report the mean and the 95%-confidence interval for our model over 10 training runs.

	BAIR	KTH	UCF101
14 in-between frames			
3D-Conv (ours)	0.836 [0.832, 0.839]	0.733 [0.729, 0.737]	0.686 [0.680, 0.693]
7 in-between frames			
SDVI, full [49]	0.880	0.901	0.598
SDVI, cond. 2 frames	0.852	0.831	—
SepConv [29]	0.877	0.904	0.443
SuperSloMo [19]	—	0.893	0.471

SSIMは高いほど良い。

UCF101はチャレンジングなデータセットらしく、それで性能が出ている

SSIMがあっていても嬉しくはない(同じ映像を作り出すタスクではない)。参考程度