Multi-View Video Synthesis Through Progressive Synthesis and Refinement

Mohamed Ilyes Lakhal[1][a], Oswald Lanz[2][b] and Andrea Cavallaro[1][c]
[1]*Queen Mary University of London, London, U.K.*
[2]*Free University of Bozen-Bolzano, Bolzano, Italy*

Keywords: Multi-View Video Synthesis, Generative Models, Temporal Consistency.

Abstract: Multi-view video synthesis aims to reproduce a video as seen from a targeted viewpoint. This paper proposes to tackle this problem using a multi-stage framework to progressively add more details on the synthesized frames and refine wrong pixels from previous predictions. First, we reconstruct the foreground and the background by using 3D mesh. To do so, we leverage the one-to-one correspondence between rendered mesh faces between the input and the target view. Then, the predicted frames are defined with a recurrence formula to correct wrong pixels and adding high-frequency details. Results on the NTU RGB+D dataset show the effectiveness of the proposed approach against frame-based and video-based state-of-the-art models.

1 INTRODUCTION

Multi-view video synthesis tries to synthesize a video of a person performing an action as seen from a target view given an input-view video (see Fig. 1).

The multi-view synthesis problem can be classified onto computer-graphics (Zhang et al., 2017) methods and learning-based methods (Ma et al., 2017). The computer-graphics methods synthesize detailed body from an arbitrary viewpoint, however, the computation time to get the body representation and the sensibility towards unseen clothing make them hard to be applied to in-the-wild data. Learning-based methods use generative models such as Generative Adversarial Network (GAN) (Goodfellow et al., 2014) to synthesize the person from a target-view. Recent advance in human-mesh recovery (Kanazawa et al., 2019) has allowed the combination of both expressive cues (3D mesh) and GANs (Liu et al., 2019) of a person seen from a target-view.

In this paper, we propose a pipeline that consists of iterative steps (or layers) of synthesis. Each layer is represented as a GAN network. The first (or *reconstruction*) layer, synthesizes the low-level frequencies (*i.e.* overall structure of the novel-view scene and human body). The network structure uses a mask containing information about the visibility of the input view onto the target view. The mask is obtained from the 3D body mesh to separate the information of the background, visible body part, and occluded body part. The mask is shown to be useful for the network to focus on each of the three parts individually

Figure 1: **Multi-view video synthesis.** Given a video of a scene with a person acting (*e.g.* throwing), captured from a fixed camera, the goal is to synthesize a video from a target view as it would be captured by another fixed camera from a different viewpoint.

[a] https://orcid.org/0000-0003-4432-9740
[b] https://orcid.org/0000-0003-4793-4276
[c] https://orcid.org/0000-0001-5086-7858

Lakhal, M., Lanz, O. and Cavallaro, A.
Multi-View Video Synthesis Through Progressive Synthesis and Refinement.
DOI: 10.5220/0011759400003417
In *Proceedings of the 18th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2023) - Volume 5: VISAPP*, pages 230-238
ISBN: 978-989-758-634-7; ISSN: 2184-4321

in the synthesis. Then, the *refinement* layer is an iterative module that takes the synthesis from the previous layer and recursively synthesizes and refines it.

2 BACKGROUND

Scene synthesis methods focus on reconstructing a novel view from neighboring regions of the captured scene. Methods use synchronized multi-view images to infer the 3D geometry in order to reconstruct the scene (Chaurasia et al., 2013; Niklaus et al., 2019; Wiles et al., 2020). The novel view is then obtained by re-projection onto the image plane of the desired camera view. However, most of these methods suppose that the scene contains static objects which makes the synthesis stable (*i.e.* less jitter effect), are adapted for indoor scenes (Shin et al., 2019).

Articulated object novel-view synthesis includes: human bodies (Kundu et al., 2020), human faces (Wu et al., 2020). Methods estimating the 3D human body shape from videos can be categorized into template-based (Vlasic et al., 2008), model-based (Bogo et al., 2015), free-form (Guo et al., 2017) and learning-based reconstruction (Saito et al., 2019). These methods require 3D mesh for each human body and clothing dress which makes them impractical in some scenarios (*e.g.* in-the-wild applications).

Synthesizing the human body motion attracted a lot of attention lately (Ma et al., 2017; Liu et al., 2019; Dong et al., 2018; Wang et al., 2019). Successful attempts to the problem are related to the pose-guided view synthesis problem (Ma et al., 2017). The synthesis generator combines the input image of the person from a given pose along with a target pose. The more descriptive (*i.e.* descriptive of the human body on other representation space than RGB) the target pose is the better the synthesis (Liu et al., 2019; Dong et al., 2018). In contrast, the multi-view synthesis problem focuses on synthesizing the same pose of the human body as well as keeping a temporal consistency across the frames (Lakhal et al., 2019; Lakhal et al., 2020).

The modeling of the generator can benefit from progressively synthesizing and refining. Methods in the literature can be divided into two categories, feature-block recurrence or model recurrence. Feature-block recurrence (Men et al., 2020; Zhu et al., 2019) progressively learns to transfer the pose in the feature space at each block of the generator model. On the other hand, model recurrence (Karras et al., 2018a; Shaham et al., 2019) progressively refines the prediction using a generator initialized at each step. By doing so, the network needs to learn the data distribution by starting from a good initialization at the previous step.

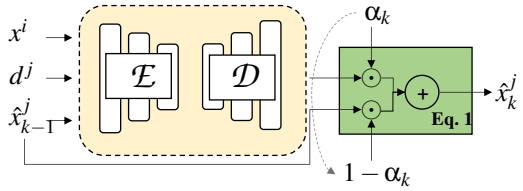


Figure 2: **Overview of the proposed PSR-Net.** The model is composed of K generators (called layers). At each layer $k \in \{2,\dots,K\}$ the generator $G^k_{i\to j}$ takes the input view video x^i, target depth d^j and the synthesis from the previous stage $k-1$ which will guide the synthesis to produce $\hat{x}^j_k$. The video is obtained with Eq. 1 from the synthesis of the layer $k-1$ and k. KEY – α_k: weighting factor.

3 MODEL

This section presents our proposed Progressive Synthesis and Refinement Network (PSR-Net) model. We provide insights on key choices and we detail the architecture of each stage of the proposed pipeline.

The idea of our PSR-Net is inspired by the success of progressive GANs (Shaham et al., 2019; Karras et al., 2018b) where the image is gradually synthesized and refined. The pipeline presented in Fig. 2 is composed of K generators (also called layers), not necessarily of the same architecture, that gradually synthesize the video using iterative recurrence formula as:

$$\hat{x}^j_k = \begin{cases} G^{k=1}_{i\to j}(x^i, \hat{x}^j_0, d^j) & \text{if } k=1 \\ \alpha_k G^k_{i\to j}(x^i, \hat{x}^j_{k-1}, d^j) + (1-\alpha_k)\hat{x}^j_{k-1} & \text{if } k \geq 2 \end{cases} \quad (1)$$

where $\hat{x}^j_0$ is the initial target-view estimate, d^j target-view depth, and x^i (resp. x^j) input (resp. target) view video. The weighting factor $\alpha_k \in [0,1]$ is a hyper-parameter and $k \in \{1,\dots,K\}$ is the layer index.

3.1 Reconstruction Layer $G^{k=1}_{i\to j}$

We aim to synthesize the foreground and background guided by the initial foreground estimation of $\hat{x}^j_0$. We first estimate the 3D mesh using an off the shelf estimator such as the method proposed in (Kanazawa et al., 2019). Then, we synthesize the foreground and the background with dedicated decoders using the mask obtained from the projection of the 3D mesh on the target view.

Visibility Map. Let $F^i \in \mathbb{R}^{T\times W\times H}$ be the projection of a 3D human body mesh onto the image plane of

Algorithm 1: Visibility map.

Input: F^i; F^j; k (k-NN) value;
k-NN_face (Step I & II (Lakhal et al., 2020))
Output: Visibility map $\mathbf{M^{i\to j}}$

$\mathbf{M^{i\to j}} \leftarrow 1_{T\times w\times h}$
for $(t,u_x,u_y) \in \{[1..T],[1..w],[1..h]\}$ **do**
  if k-NN_face$(F^j[t,u_x,u_y],k)$ **then**
    $\mathbf{M^{i\to j}}[t,u_x,u_y] \leftarrow 2$
  else
    $\mathbf{M^{i\to j}}[t,u_x,u_y] \leftarrow 3$
  end
end

view i capturing the scene of the action, where T, W, and H represent the number of frames, width, and height of the video stream. In practice, the 3D human body mesh can be represented with the SMPL (Loper et al., 2015) model and estimated using an off-the-shelf human-mesh recovery method (Kanazawa et al., 2019). Alg. 1 describes the process to obtain the visibility map. Using the one-to-one correspondence between the projected mesh F^i (resp. F^j) of the input (resp. target) view, we build the visibility map $\mathbf{M}^{i\to j} \in \{1,2,3\}^{T\times w\times h}$ where the labels (*1, 2, 3*) represent (background, visible and occluded). Instead of independently processing the mesh from the temporal axis, we leverage the information available about the faces F^i and F^j. We consider a mesh index (or face) $f \in F_t^j$ occluded if $f \notin F^i$ for all $t \in \{1,\dots,T\}$ (see Fig. 3).

Network Architecture. We adopt a context-based generator with separated encoders $\mathcal{E}_*$ for each of the inputs $x^i, \hat{x}_0^j$, and d^j. Then, each part that the visibility mask $\mathbf{M}^{i\to j}$ represents (*i.e.* background, visible and occluded) is decoded using a separated decoder $\mathcal{D}_*$ and $\mathbf{M}^{i\to j}$ (see Fig. 4(d)). Let $\mathbf{k}^{i\to j} = \{(u_x,u_y) \in \mathbb{N}^{w\times h} | \mathbf{M}^{i\to j}[t,u_x,u_y] = k; 1 \le t \le T\}$ refers to the map with the label $k \in \{1, 2, 3\}$. The novel-view video is obtained as:

$$\tilde{x}_1^j = \hat{x}_b^j \odot \mathbf{1}^{i\to j} + \hat{x}_v^j \odot \mathbf{2}^{i\to j} + \hat{x}_o^j \odot \mathbf{3}^{i\to j}, \quad (2)$$

where $\hat{x}_b^j$ (resp. $\hat{x}_v^j$, and $\hat{x}_o^j$) is the background (resp. visible, and occluded) part synthesis and is obtained using the decoder $\mathcal{D}_b$ (resp. $\mathcal{D}_v$, and $\mathcal{D}_o$). We also add an explicit temporal constraint using the estimated optical-flow obtained from the target view mesh O^j and a warping function $\mathcal{W}$ as defined in (Jaderberg et al., 2015) such as:

$$\hat{x}_1^j = \begin{cases} \tilde{x}_{1,t}^j & \text{if } t = 1 \\ \tilde{x}_{1,t}^j + \zeta.\mathcal{W}(\tilde{x}_{1,t-1}^j, O_{t+1\to t}^j) & \text{if } t \in [2..T], \end{cases} \quad (3)$$

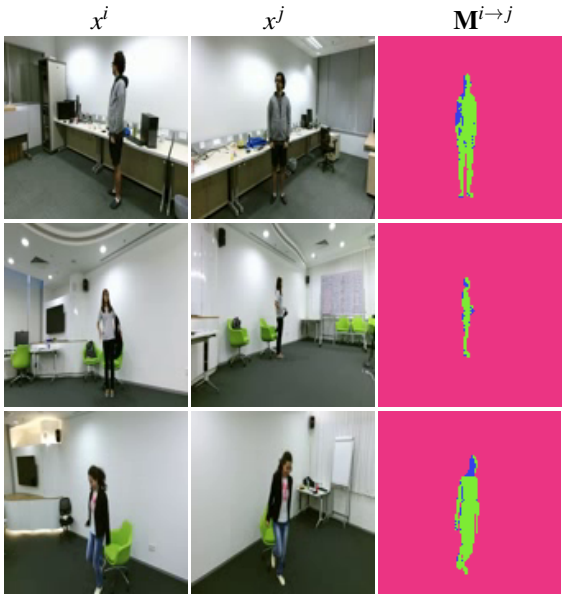


Figure 3: **Visibility map.** Using the correspondence between the projected mesh onto the input view and of the target view, we determine the visibility map $\mathbf{M}^{i\to j}$ (see Alg. 1). We build a dictionary using the projected mesh in the input view by keeping track whether a face index is present along with its neighboring visible faces. Then, using the resulted dictionary, we assign to each face index in the target-view mesh the visibility value. Legend – x^i: input-view; x^j: target-view; background; occluded; visible.

where ζ, controls the importance of the warping of previously synthesized frames with the synthesis from the current time frame produced by the generator. The factor ζ is set empirically to .1 as suggested by (Lakhal et al., 2020).

To train the generator we use a Huber loss L_r (Huber, 1964) as reconstruction loss. We also employ a temporal perceptual loss L_p (Lakhal et al., 2019) that penalises the prediction on the spatio-temporal feature space from a function Φ called perceptual network. We employ an adversarial loss (Goodfellow et al., 2014) (denoted as L_a), in order to enforce high frequency constraints. The training loss for the reconstruction layer is given as:

$$L_{k=1} = L_r + 0.01(L_p + L_a). \quad (4)$$

3.2 Recurrence Layer $G_{i\to j}^k$

The role of this layer is to recursively refine the synthesized video by adding high-frequency details. We give equal priority to all pixels and, therefore, choose to predict $\hat{x}_k^j$ using one decoder.

Network Architecture. The synthesized videos $\{\hat{x}_u^j\}_{u=1}^k$ with $k \in \{2,\dots,K\}$ progressively approximate better the novel-view video. Therefore, we use

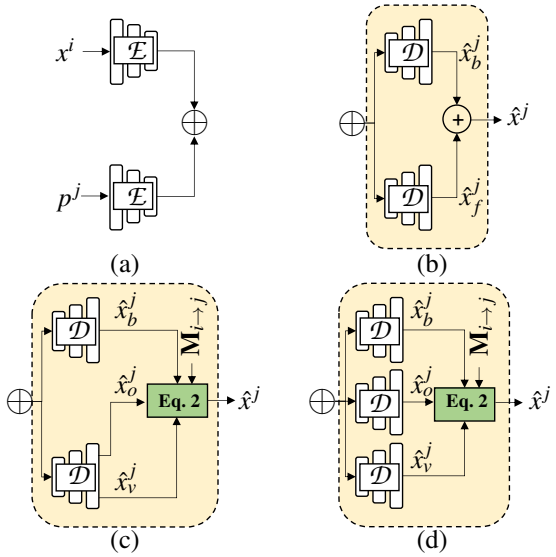


Figure 4: **Reconstruction layer variants.** (a) *Feature encoding*: the input view video x^i and the target view prior p^j are encoded using separate encoders and are merged together using a convolution operation; (b) *context-based approach*: we use separated decoders $\mathcal{D}$ for each of the foreground and the background; (c) *proposed (common head)*: the visible and occluded regions are synthesized using separated heads (*i.e.* final deconvolution operation) but they share the same decoder; (d) *proposed (separate heads)*: each information from the map $\mathbf{M}^{i\to j}$ is synthesized on a separated branch. KEY – p^j: target modality *e.g.* depth; $\hat{x}_f^j$: foreground; $\hat{x}_b^j$: background; $\hat{x}_o^j$: occluded; $\hat{x}_v^j$: visible; $\oplus$: concatenation.

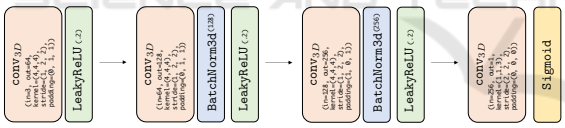


Figure 5: Detailed architecture of the temporal discriminator $\mathbf{D}_t$.

two encoders, one encoder $\mathcal{E}_v$ (parameterized by θ_v) for the videos x^i and $\hat{x}_{k-1}^j$, the other encoder is for the target depth $\mathcal{E}_d$. As motivated above, we use one decoder $\mathcal{D}_k$ in order to synthesize the video. To account for temporal consistency and encourage smooth video synthesis, we add a video-based discriminator in the adversarial training. We use a similar discriminator architecture as the one proposed in (Tulyakov et al., 2018) (see Fig. 5). The total loss to train the model is given by:

$$L_k = L_r + 0.01 L_a. \tag{5}$$

4 EXPERIMENTS

This section describes the evaluation setup along with quantitative and qualitative results against state-of-the-art methods to assess the effectiveness of the our proposed model. We also present a detailed ablation study to justify the architectural choices that we make in the model.

4.1 Experimental Setup

Dataset. We use NTU RGB+D (Shahroudy et al., 2016), a synchronised multi-view action recognition dataset where ach action is captured from 3 views. We use the following evaluation measures:

- **Frame-Base Evaluation.** We assess the per-frame synthesis quality of each model using Structural Similarity (SSIM) and by measuring the error-sensitivity using Peak Signal-to-Noise-Ratio (PSNR) (Zhou Wang et al., 2004).
- **Fréchet Video Distance.** (FVD) (Unterthiner et al., 2019) is an extension of Fréchet Inception Distance (FID) (Heusel et al., 2017) and it is specifically designed for videos. FVD quantifies the quality and the diversity of samples generated from a parametric model with respect to the ground truth.
- **Pose Estimation.** We use the pose estimator proposed in (Raaj et al., 2019) and report the L_2 error, Percentage of Correct Keypoints (PCK) (Yang and Ramanan, 2013) along with the precision, recall, and F1 of the estimations.

Implementation Details. We train PSR-Net with $K = 3$ layers (see Fig. 6). We use a 6 layered ResNet (Zhu et al., 2017) with 3D convolutions as a base for each of the three generators. We train all the model parameters with Adam optimizer (Kingma and Ba, 2015) with the hyperparameters $(\alpha_1, \alpha_2) = (0.5, 0.999)$. The learning rate is set to $2{\cdot}10^{-5}$.

4.2 Reconstruction Layer

The focus of the reconstruction layer is to accurately reconstruct the foreground (*i.e.* motion) in the target-view. We investigate different model architectures and highlight the effectiveness of the computed visibility mask $\mathbf{M}^{i\to j}$ in the foreground reconstruction (see Fig. 4). We present three models (Context-based, Common head, Separate heads) that take $x^i, \hat{x}_0^j$, and d^j and process them with separate encoders. Then, the feature presentation from the encoders are concatenated and fed to a convolution layer with the following configurations:

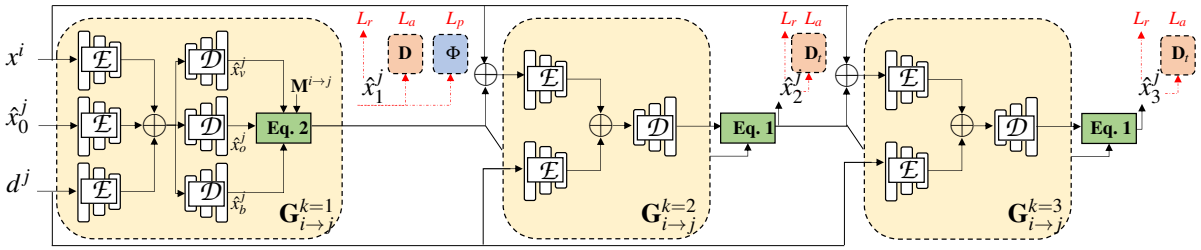


Figure 6: **Proposed PSR-Net model**. Detailed architecture of the proposed PSR-Net with $K = 3$ layers. KEY – Φ: spatiotemporal perceptual network (Lakhal et al., 2019); $\mathcal{E}$ (resp. $\mathcal{D}$): encoder (resp. decoder) network, we omit the weights θ for clarity; $\rightarrow$: information flow; $\oplus$: concatenation.

Table 1: Results of the model ablation of the first layer $G^{k=1}_{i\rightarrow j}$. We investigate the synthesis performance with the different decoding strategies. The focus of this network is about foreground synthesis and hence the background synthesis is not a primary model evaluation measure. KEY – M: mask, best, second best.

Decoders	SSIM	M-SSIM	PSNR	M-PSNR
Context-based	.811	.981	23.32	32.31
Common head	.795	.980	23.16	32.21
Separate heads	.801	.981	23.13	32.38

Table 2: Model ablation of the second stage $G^{k=2}_{i\rightarrow j}$. We investigate different encoding strategies. KEY – M: mask, best, second best.

Encoders	SSIM	M-SSIM	PSNR	M-PSNR
Separate encoders	.801	.978	23.12	31.57
Concat all	.791	.978	22.97	31.64
Concat videos	.811	.978	23.25	31.41

- **Context-Based:** we only synthesize the foreground and the background with the dedicated decoders (see Fig. 4(a)).
- **Common Head:** in the decoder of the foreground branch, we output two convolution heads that correspond to the visible and occluded part from the input view (see Fig. 4(b)).
- **Separate Heads:** we exploit the information available in the visibility mask $\mathbf{M}^{i\rightarrow j}$ and output three branches with separate weights to synthesize the background, visible, and occluded parts respectively (see Fig. 4(c)).

Results show that the proposed model using separate decoders (heads) with the map $\mathbf{M}^{i\rightarrow j}$ produces better foreground synthesis compared to the two other models (see Tab. 1). Therefore, we keep this model as a default for $G^{k=1}_{i\rightarrow j}$.

4.3 Recurrence Layer

The aim of the following ablations is to set a base model for the recurrence layers, *i.e.* $G^{k}_{i\rightarrow j}, k \geq 2$. We conduct the following ablation to show the importance of inputs encoding at this stage.

- **Separate Encoders.** Each input $x^i, \hat{x}^j_{k-1}$, and d^j have separate encoder $\mathcal{E}_x$, $\mathcal{E}_{k-1}$, and $\mathcal{E}_d$ respectively.
- **Concat All.** We concatenate all the inputs $x^i, \hat{x}^j_{k-1}, d^j$ and then apply one encoder, $\mathcal{E}_v$.
- **Concat Videos.** The videos $x^i, \hat{x}^j_{k-1}$ are concatenated and encoded with a single encoder, $\mathcal{E}_v$, and the depth is encoded with $\mathcal{E}_d$.

Tab. 2 reports the scores using the three strategies above. Results show that combining the videos x^i and $\hat{x}^j_{k-1}$ with a single encoder provide better refinement with .811 (resp. 23.25) SSIM (resp. PSNR) score. The model using three encoders could not generalize well compared to the one that uses $\mathcal{E}_v$ and $\mathcal{E}_d$. This is because at this stage, the model uses prediction from the reconstruction layer (does not learn from scratch).

To avoid extensive hyper-parameter search, we fix the weighting factor α_k at each stage $k \in \{1, \ldots, K\}$ for all layers. Fig. 7 reports the sensitivity of the model $G^{k=1}_{i\rightarrow j}$ with different $\alpha \in \{.1, .3, .5, .7, .9\}$ val-

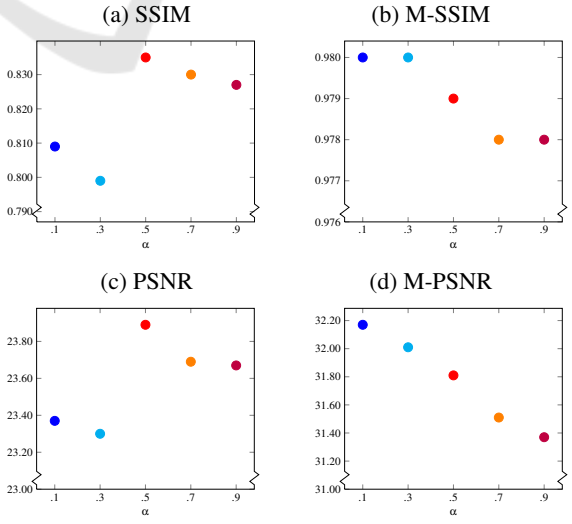


Figure 7: Sensitivity analysis with different α values using the recurrence layer $G^{k=2}_{i\rightarrow j}$ (higher score better). KEY – M: mask.

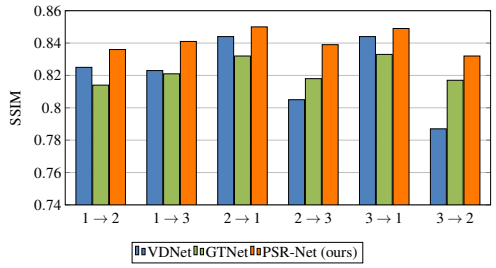
ues. We can see that $\alpha = .5$ obtains the the best value. This suggests that the synthesis of the previous layer is equally important to the synthesis in the k-th layer. We also note that the effectiveness of the proposed weighting scheme compared to conventional synthesis (without weighting). The *concat videos* model produces an overall SSIM of .811, however, by adding the weighting factor as defined in Eq. 1 with $\alpha_k = .5$ produces an overall SSIM of .835. For the rest of the paper, we fix *concat videos* using $\alpha_k = .5$ as default model with $G^{l=k}_{i\to j}$, for $k \geq 2$.

4.4 State-of-the-Art Comparison

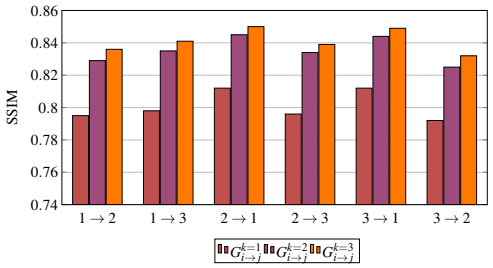
Model Performance. After each layer, the model enhances the synthesis reconstruction (see Tab. 3). This is because the network $G^k_{i\to j}; k \geq 2$ focuses on correcting mis-classified pixels and refining the videos. However, we notice a slight drop in the foreground synthesis (from .981 M-SSIM to .979) as we are no longer using the visibility mask $\mathbf{M}^{i\to j}$ with the reconstruction loss. The reason is we are assuming that the recurrence layer is to refine the synthesis and not to produce the temporal motion in the novel-view. Fig. 10 depicts two examples showing the idea of progressively refining the results and adding high-frequency details, *e.g.* in the first row after synthesizing the novel-view video the recurrence networks ($G^{k=2}_{i\to j}$ and $G^{k=3}_{i\to j}$) focus on correcting the lighting of the room from a wrong prediction in $G^{k=1}_{i\to j}$.

We further investigate challenging synthesis cases to better understand the model behaviour. Fig. 9 shows that, in some cases, the synthesis quality decreases from its initial prediction. We can see that in these cases, the network fails to rectify the artifacts.

Model Comparison. Tab. 3 compares the proposed PSR-Net with $K = 3$ against VDNet (Lakhal et al., 2019) and GTNet (Lakhal et al., 2020) and frame-base method PG^2 (Ma et al., 2017) and PATN (Zhu et al., 2019). We clearly see the advantage of the video based models compared to the frame-based. This is because the video based models use explicit temporal consistency across the synthesis. Also, the proposed model outperforms the other video-based models with an SSIM score of .841 and PSNR score of 24.01. Fig. 8 highlights the per-view SSIM scores of each model. We can see that the performance gradually enhances after each iteration. We also note that the overall scores obtained are consistent across views and the proposed method performs well in the challenging case of view $3 \to 2$. Since we explicitly avoid using context-based approach after the first stage, it is expected that GTNet produces better foreground synthesis than $G^{k=3}_{i\to j}$ with an M-SSIM (resp. M-PSNR)



(a) state-of-the-art comparison



(b) PSR-Net performance with $K = 3$ layers

Figure 8: Per-view SSIM comparison between the proposed PSR-Net against VDNet (Lakhal et al., 2019) and GTNet (Lakhal et al., 2020). KEY – $i \to j$: synthesis from view i to j.

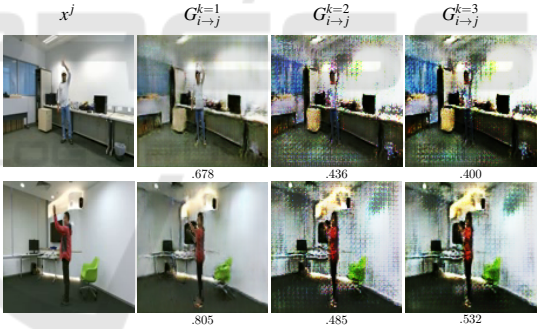


Figure 9: Failure cases using PSR-Net with $K = 3$ layers. We show two examples where the model fails to enhance the synthesized video and adds visible artifacts that degrade the visual quality. The bottom scores represent the SSIM score.

score of .981 vs. .979 (resp. 32.50 vs. 31.81). We also note that we only use the texture transfer modality as opposed to GTNet which uses extra modality (semantic segmentation). Tab. 4 reports the pose estimation scores, we note that the reconstruction model $G^{k=1}_{i\to j}$ performs well, however, the models $G^{k=2}_{i\to j}$ and $G^{k=3}_{i\to j}$ have higher L_2 error which is reflected by a lower PCK and F1 scores. To understand better this behaviour, we visualise the skeleton estimation over the synthesized frames in Fig. 11, we notice that because of the artifacts described earlier the pose-estimator confuses and outputs wrong keypoint locations (*e.g*

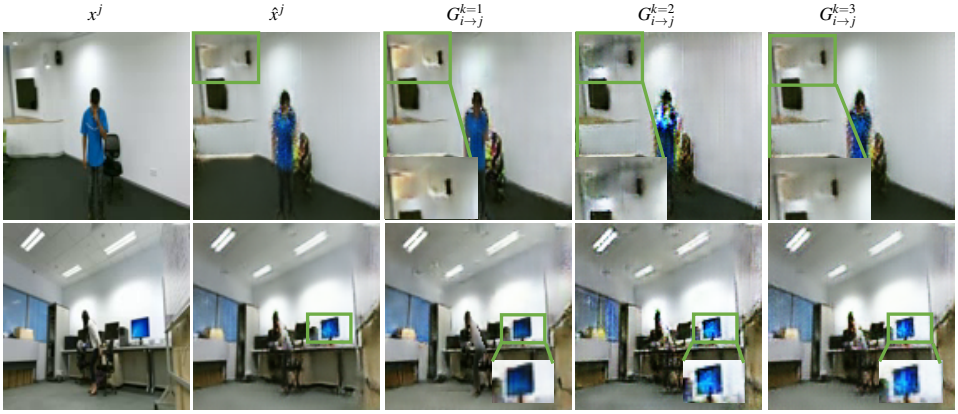


Figure 10: Progressive synthesis refinement with PSR-Net model using $K = 3$ layers. The first layer focuses on low-level frequencies (overall structure). Then, the refinement model focuses on the high-level details *e.g.* screen in the second row.

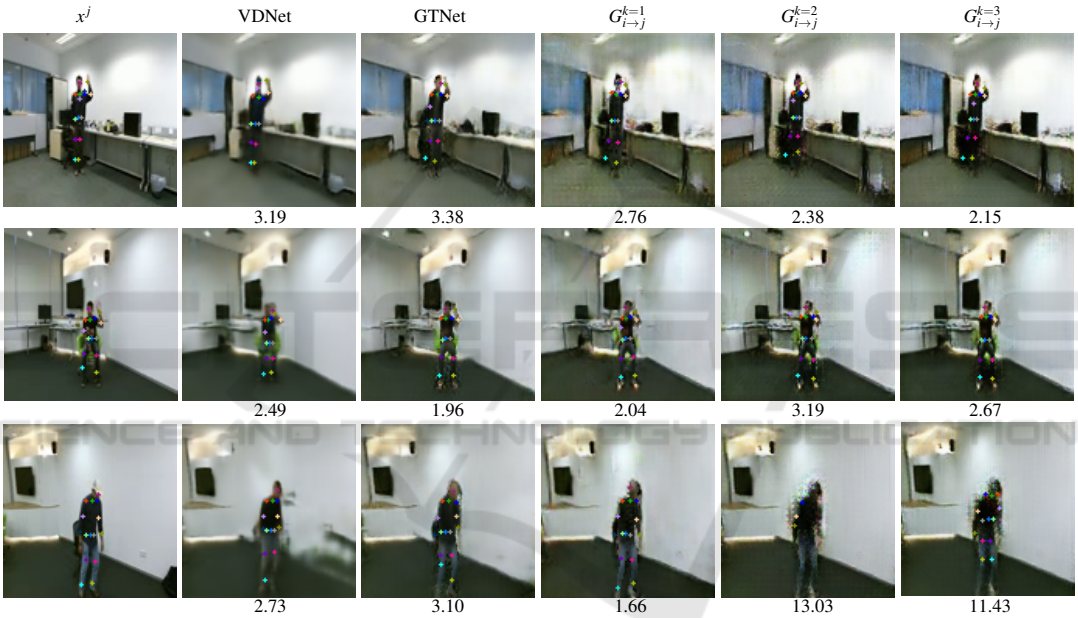


Figure 11: Qualitative results of the pose-estimation evaluation of the proposed PSR-Net against and video-based methods: VDNet (Lakhal et al., 2019) and GTNet (Lakhal et al., 2020). KEY – bottom score: L_2 error of the pose estimation.

third row). This explains the lower pose estimation performance and also causes higher FVD scores (see Tab. 3). Fig. 12 compares the methods described above. We can see that using the recurrence layer focuses more on the background synthesis (*e.g* fifth column).

5 CONCLUSIONS

We present a pipeline that consists of iteratively synthesizing and refining the target-view video using a recurrence formula. First, using a computed visibility map obtained from the one-to-one correspondence

Table 3: Comparison of the proposed PSR-Net against frame-based methods (PG^2 (Ma et al., 2017) and PATN (Zhu et al., 2019)) and video-based methods (VDNet (Lakhal et al., 2019) and GTNet (Lakhal et al., 2020)). KEY: M: mask; s^j: 2D skeleton; $\mathcal{T}_{i\to j}^s$: texture transfer (Lakhal et al., 2020) ; $\mathcal{S}^j$: semantic segmentation; d^j: depth; best, second best.

Model	Modality	SSIM	M-SSIM	PSNR	M-PSNR	FVD
PG^2	s^j	.582	95.4	16.90	25.87	11.84
PATN	s^i, s^j	.534	.948	16.24	24.55	13.11
VDNet	d^j, s^j	.821	.972	23.18	29.70	**5.78**
GTNet	$\mathcal{T}_{i\to j}^s, \mathcal{S}^j, d^j$	.823	**.981**	23.81	**32.50**	**4.96**
$G_{i\to j}^{k=1}$	$\mathcal{T}_{i\to j}^s, d^j$	.801	**.981**	23.13	**32.38**	5.90
$G_{i\to j}^{k=2}$	$\mathcal{T}_{i\to j}^s, d^j$	**.835**	**.979**	**23.89**	31.81	8.14
$G_{i\to j}^{k=3}$	$\mathcal{T}_{i\to j}^s, d^j$	**.841**	**.979**	**24.01**	31.81	7.49

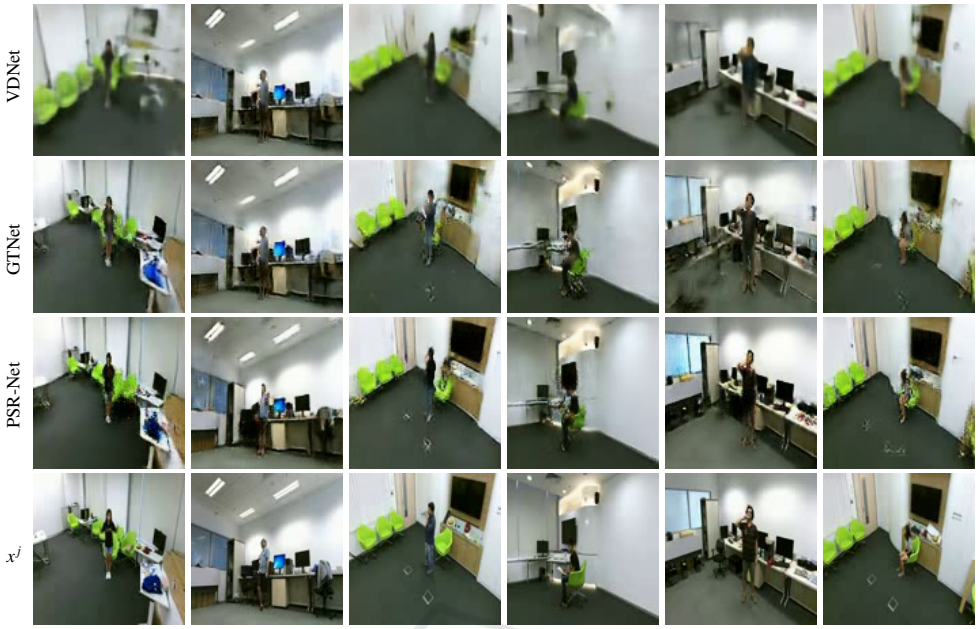


Figure 12: Qualitative comparison of the proposed PSR-Net against VDNet (Lakhal et al., 2019) and GTNet (Lakhal et al., 2020).

Table 4: Human pose estimation performance over synthesized novel-view videos of the proposed PSR-Net against frame-based methods (PG^2 (Ma et al., 2017) and PATN (Zhu et al., 2019)) and video-based methods (VDNet (Lakhal et al., 2019) and GTNet (Lakhal et al., 2020)). KEY – best; second best.

Model	**L_2**	**PCK** 0.20	0.05	0.01	**Prec.**	**Rec.**	**F1**
PG^2	10.91	97.8	74.7	14.4	88.1	12.8	22.4
PATN	11.68	98.0	69.7	10.22	88.4	.1	17.8
VDNet	4.37	99.3	92.4	51.2	91.0	**55.3**	**68.7**
GTNet	3.95	**99.5**	93.0	57.6	92.3	52.7	67.1
$G_{i\to j}^{k=1}$	**3.21**	**99.5**	**94.8**	**63.6**	91.3	**62.6**	**74.3**
$G_{i\to j}^{k=2}$	**3.85**	**99.6**	**94.3**	**58.3**	**92.9**	46.3	61.8
$G_{i\to j}^{k=3}$	4.04	**99.6**	93.7	56.9	**93.5**	42.5	58.5

between the input and target 3D human body mesh the novel-view is synthesized with a first generator. Then, we iteratively refine the synthesis using the results from the previous step via a weighting scheme. Results on the NTU RGB+D dataset demonstrate that the proposed approach is effective and produces high-quality background and foreground frames.

REFERENCES

Bogo, F., Black, M. J., Loper, M., and Romero, J. (2015). Detailed full-body reconstructions of moving people from monocular RGB-D sequences. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 2300–2308.

Chaurasia, G., Duchene, S., Sorkine-Hornung, O., and Drettakis, G. (2013). Depth synthesis and local warps for plausible image-based navigation. *ACM Transactions on Graphics (TOG)*, 32(3).

Dong, H., Liang, X., Gong, K., Lai, H., Zhu, J., and Yin, J. (2018). Soft-Gated Warping-GAN for Pose-Guided Person Image Synthesis. In *Neural Information Processing Systems (NeurIPS)*, pages 474–484.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative Adversarial Nets. In *Neural Information Processing Systems (NeurIPS)*, pages 2672–2680.

Guo, K., Xu, F., Yu, T., Liu, X., Dai, Q., and Liu, Y. (2017). Real-time geometry, albedo, and motion reconstruction using a single rgb-d camera. *ACM Transactions on Graphics (TOG)*, 36(4).

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. (2017). GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. In *Neural Information Processing Systems (NeurIPS)*, pages 6626–6637.

Huber, P. J. (1964). Robust Estimation of a Location Parameter. *Annals of Mathematical Statistics*, 35(1):73–101.

Jaderberg, M., Simonyan, K., Zisserman, A., and kavukcuoglu, k. (2015). Spatial transformer networks. In *Neural Information Processing Systems (NeurIPS)*, pages 2017–2025.

Kanazawa, A., Zhang, J. Y., Felsen, P., and Malik, J. (2019). Learning 3D Human Dynamics From Video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5607–5616.

Karras, T., Aila, T., Laine, S., and Lehtinen, J. (2018a). Progressive Growing of GANs for Improved Quality, Stability, and Variation. In *Proceedings of the In-*

ternational Conference on Learning Representations (ICLR).

Karras, T., Aila, T., Laine, S., and Lehtinen, J. (2018b). Progressive Growing of GANs for Improved Quality, Stability, and Variation. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Kingma, D. P. and Ba, J. (2015). Adam: A Method for Stochastic Optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Kundu, J. N., Seth, S., Jampani, V., Rakesh, M., Babu, R. V., and Chakraborty, A. (2020). Self-Supervised 3D Human Pose Estimation via Part Guided Novel Image Synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6151–6161.

Lakhal, M. I., Boscaini, D., Poiesi, F., Lanz, O., and Cavallaro, A. (2020). Novel-view human action synthesis. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*.

Lakhal, M. I., Lanz, O., and Cavallaro, A. (2019). View-LSTM: Novel-view video synthesis through view decomposition. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 7576–7586.

Liu, W., Piao, Z., Jie, M., Luo, W., Ma, L., and Gao, S. (2019). Liquid Warping GAN: A Unified Framework for Human Motion Imitation, Appearance Transfer and Novel View Synthesis. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 5903–5912.

Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., and Black, M. J. (2015). SMPL: A Skinned Multi-Person Linear Model. *ACM Transactions on Graphics (TOG)*, 34(6):248:1–248:16.

Ma, L., Jia, X., Sun, Q., Schiele, B., Tuytelaars, T., and Van Gool, L. (2017). Pose Guided Person Image Generation. In *Neural Information Processing Systems (NeurIPS)*, pages 406–416.

Men, Y., Mao, Y., Jiang, Y., Ma, W.-Y., and Lian, Z. (2020). Controllable Person Image Synthesis with Attribute-Decomposed GAN. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5083–5092.

Niklaus, S., Mai, L., Yang, J., and Liu, F. (2019). 3D Ken Burns Effect from a Single Image. *ACM Transactions on Graphics (TOG)*, 38(6).

Raaj, Y., Idrees, H., Hidalgo, G., and Sheikh, Y. (2019). Efficient Online Multi-Person 2D Pose Tracking With Recurrent Spatio-Temporal Affinity Fields. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4615–4623.

Saito, S., Huang, Z., Natsume, R., Morishima, S., Kanazawa, A., and Li, H. (2019). PIFu: Pixel-Aligned Implicit Function for High-Resolution Clothed Human Digitization. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 2304–2314.

Shaham, T. R., Dekel, T., and Michaeli, T. (2019). SinGAN: Learning a Generative Model From a Single Natural Image. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 4569–4579.

Shahroudy, A., Liu, J., Ng, T.-T., and Wang, G. (2016). NTU RGB+D: A large scale dataset for 3D human activity analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1010–1019.

Shin, D., Ren, Z., Sudderth, E. B., and Fowlkes, C. C. (2019). 3D Scene Reconstruction with Multi-layer Depth and Epipolar Transformers. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 2172–2182.

Tulyakov, S., Liu, M.-Y., Yang, X., and Kautz, J. (2018). MoCoGAN: Decomposing motion and content for video generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1526–1535.

Unterthiner, T., van Steenkiste, S., Kurach, K., Marinier, R., Michalski, M., and Gelly, S. (2019). FVD: A new Metric for Video Generation. In *Proceedings of the International Conference on Learning Representations (ICLR) Workshops*.

Vlasic, D., Baran, I., Matusik, W., and Popović, J. (2008). Articulated Mesh Animation from Multi-View Silhouettes. In *Proceedings of SIGGRAPH*, page 1–9.

Wang, T.-C., Liu, M.-Y., Tao, A., Liu, G., Kautz, J., and Catanzaro, B. (2019). Few-shot Video-to-Video Synthesis. In *Neural Information Processing Systems (NeurIPS)*, pages 5013–5024.

Wiles, O., Gkioxari, G., Szeliski, R., and Johnson, J. (2020). SynSin: End-to-End View Synthesis From a Single Image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7465–7475.

Wu, S., Rupprecht, C., and Vedaldi, A. (2020). Unsupervised Learning of Probably Symmetric Deformable 3D Objects From Images in the Wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–10.

Yang, Y. and Ramanan, D. (2013). Articulated Human Detection with Flexible Mixtures of Parts. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 35(12):2878–2890.

Zhang, C., Pujades, S., Black, M. J., and Pons-Moll, G. (2017). Detailed, accurate, human shape estimation from clothed 3D scan sequences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5484–5493.

Zhou Wang, Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. (2004). Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing*, 13(4):600–612.

Zhu, J.-Y., Park, T., Isola, P., and Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 2242–2251.

Zhu, Z., Huang, T., Shi, B., Yu, M., Wang, B., and Bai, X. (2019). Progressive pose attention transfer for person image generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2342–2351.