Appendix A. Architecture Details of CVMN

A.1. Encoder and Decoder Network

Table 1 shows the detailed configuration for each module of our CVMN. Specifically, we list the parameters we used the hourglass module and the detailed configurations for motion field decoder and visibility mask decoder.

We feed the three reference images separately into the hourglass encoder and then concatenate the three 48-channel feature tensor to form a 144-channel feature tensor. We then feed the feature tensor to four motion field decoders and two visibility mask decoders. The spatial resolution of the input is 256 × 256 with three color channels.

Table 1: Configuration details for the encoder and decoder network. $k/s/c$ stand for kernel/stride/channel. The convolution layers are always followed by a ReLU layer except for the last layer of motion decoder, and the last ReLU layer for visibility decoder is explicitly listed for emphasis

Encoder		Motion Field Decoder				Visibility Mask Decoder			
hourglass		Type	k	s	c	Type	k	s	c
stack	1	Input	-	-	144	Input	-	-	144
block	2	Conv	7	1	288	Conv	7	1	288
feature	104	MaxPool	3	2	288	Conv	3	1	576
inplanes	18	Conv	3	1	576	Conv	1	1	576
out channel	48	Conv	1	1	576	DeConv	3	1	288
		DeConv	4	2	288	DeConv	3	1	144
		DeConv	3	1	144	Conv	3	1	144
		Conv	1	1	144	Deconv	3	1	24
		DeConv	3	1	24	Conv	1	1	24
		Conv	1	1	24	ReLU	-	-	24
		Conv	1	1	24				

A.2. Blending Network

The blending network contains no learnable parameters and is implemented as a grid sampler module. It takes a reference image and a motion field tensor as input and output the warped view. The motion fields are normalized to $[-1, 1]$, which is invariant w.r.t. the actual resolution of input views. We use the grid sampler provided by PyTorch to calculate the derivative for back-propagation.

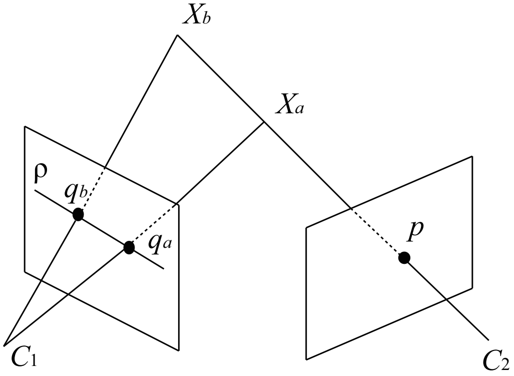

Figure 1: Epipolar geometry between two views

Appendix B. Derivation of the Epipolar Constraint

We use the epipolar constraint $\Phi(\rho, p')$ in our loss function. Here we show how to derive this constraint under our cyclic rectification.

Given the rigid transformation R, t between the two views C_1, C_2 and their intrinsics K_1, K_2, we first derive the epipolar line ρ of a pixel p. As shown in Fig. 1, we first find p's projection line in C_2 and select two 3D points $X_a, X_b \sim K_2^{-1}\tilde{p}$ on the projection line, where $\tilde{p}$ is the homogeneous coordinate of p. We then project X_a, X_b into C_1 as

$$[q_a, q_b] = \pi(K[R, t][\tilde{X_a}, \tilde{X_b}]) \tag{1}$$

where $\pi(\cdot)$ is the projection function that maps a 3D point to 2D pixel. The epipolar line ρ is thus defined by q_a, q_b. Specifically, we parameterize the epipolar line with the 2D pixel q and a normalized direction vector k. Assume p' is the correspondence of p mapped by the motion field $\mathcal{F}$. The distance between p' and the epipolar line ρ is computed as

$$\Phi(\rho, p') = \|(p' - q) - \langle p' - q, k\rangle \cdot k\| \tag{2}$$

where $\langle \cdot, \cdot \rangle$ is the inner product. Φ is differentiable w.r.t. the motion field $\mathcal{F}$. We implement Φ using basic tensor operations in PyTorch, which automatically calculate the derivative for back propagation.

Appendix C. Additional Results

C.1. Results by CVMN-I2 and CVMN-O3

Fig.2 shows the qualitative comparison results for the ablation studies. We compare synthesized sample images by our CVMN with its two variants CVMN-

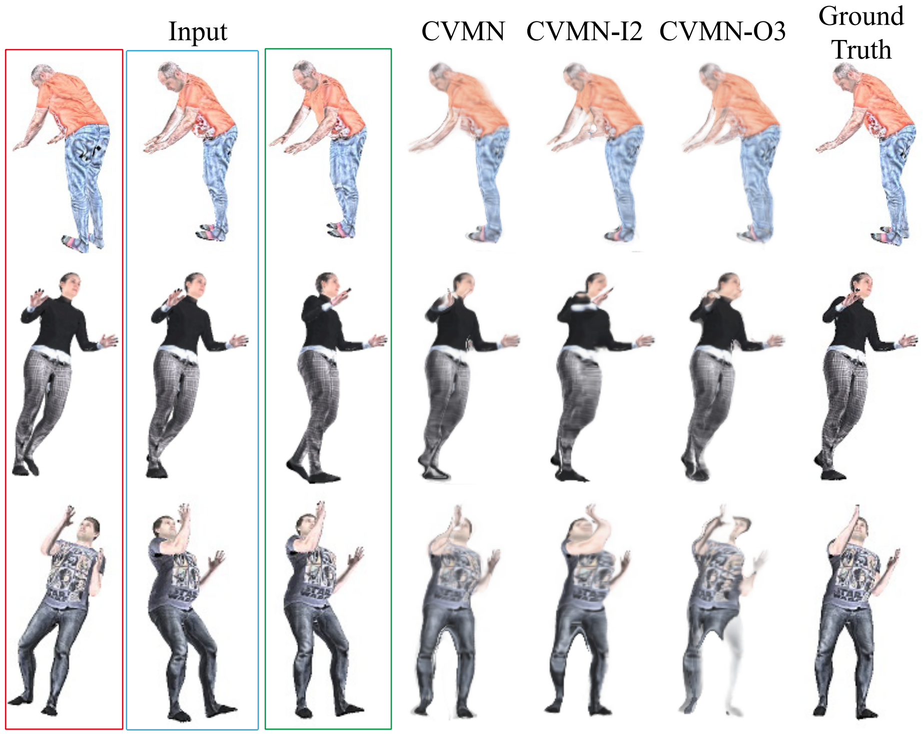

Figure 2: Qualitative comparison results for the ablation study. From left to right: we show the three input reference images, sample synthesized image by CVMN, CVMN-I2, and CVMN-O3, and the ground truth view.

I2 and CVMN-O3. Our results obviously have less artifacts and are closer to the ground truth. This indicates that our network design is optimal.

C.2. Results on SURREAL Dataset

Fig.3 shows several morphing sequences on the SURREAL dataset by our approach in comparison with the ground truth sequences. Our approach well preserves the shape and texture of the object along a circular view path. We can properly handle challenging cases with severe occlusions (*e.g.*, arms and legs).

C.3. Results on ShapeNet Dataset

Fig.4 shows several morphing sequences on the ShapeNet dataset by our approach in comparison with the ground truth sequences. It demonstrates the effectiveness of our approach on complex 3D object.

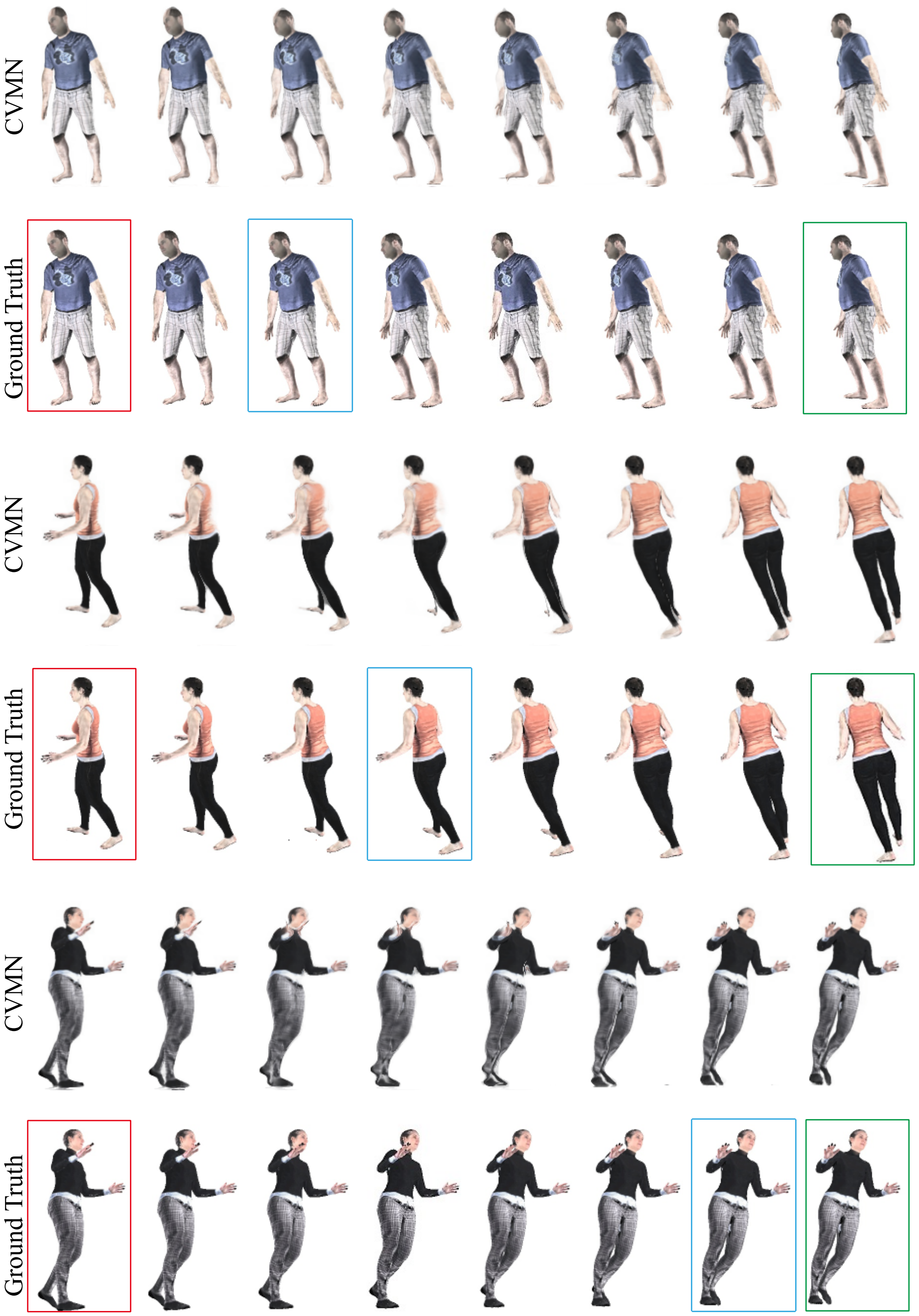

Figure 3: Additional results on SURREAL Dataset. We show 8 samples out of 24 images in our entire synthesized sequence in comparison with the ground truth images. The boxed images are used as references for our approach.

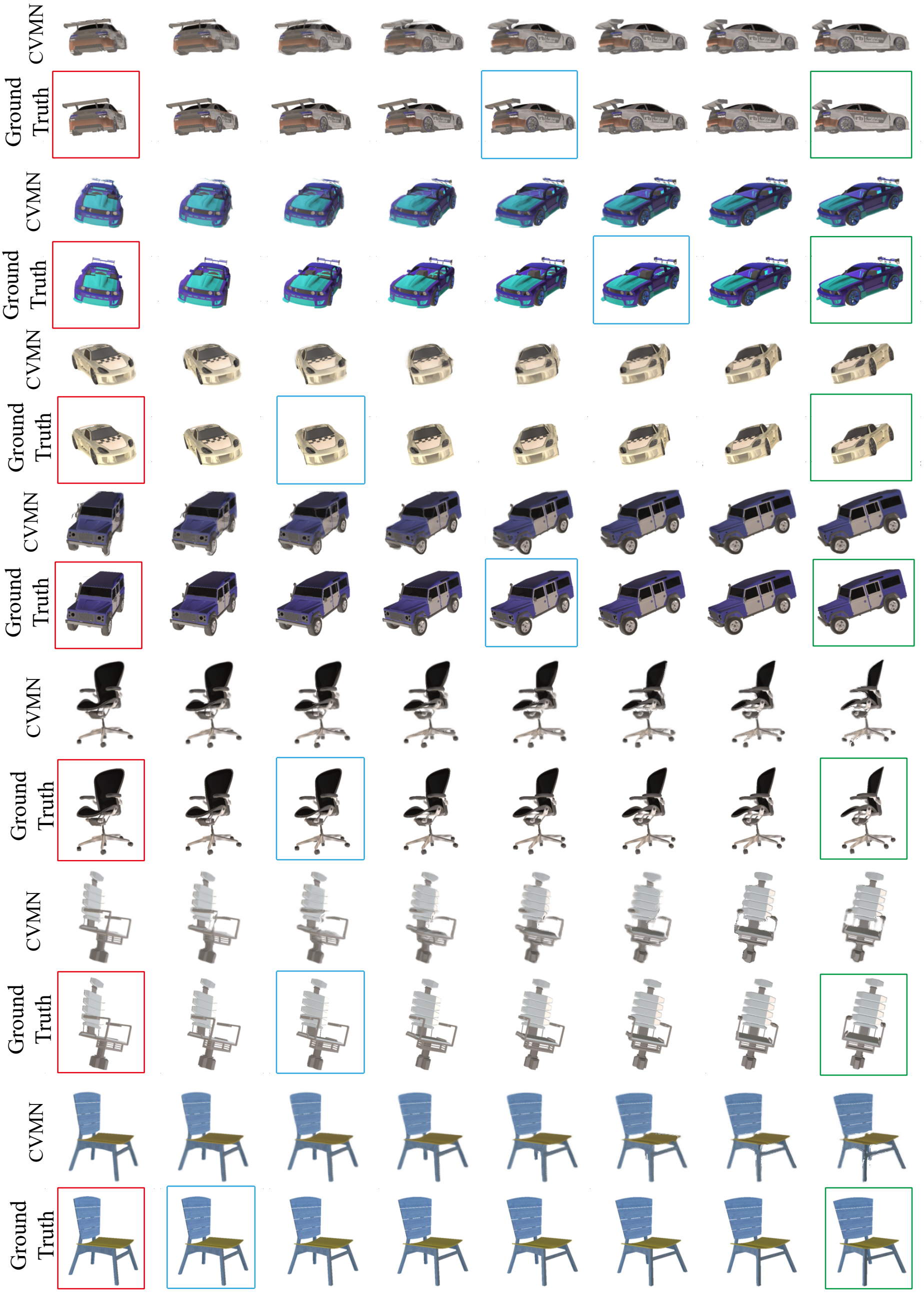

Figure 4: Additional results on ShapeNet Dataset (“car” and “chair”). We show 8 samples out of 24 images in our entire synthesized sequence in comparison with the ground truth images. The boxed images are used as references for our approach.