

Occlusion-robust Stylization for Drawing-based 3D Animation

Sunjae Yoon Gwanhyeong Koo Younghwan Lee Ji Woo Hong Chang D. Yoo
 School of Electrical Engineering, KAIST

{sunjae.yoon, kookie, youngh2, jiwoohong93, cd.yoo}@kaist.ac.kr

Abstract

3D animation aims to generate a 3D animated video from an input image and a target 3D motion sequence. Recent advances in image-to-3D models enable the creation of animations directly from user-hand drawings. Distinguished from conventional 3D animation, drawing-based 3D animation is crucial to preserve artist's unique style properties, such as rough contours and distinct stroke patterns. However, recent methods still exhibit quality deterioration in style properties, especially under occlusions caused by overlapping body parts, leading to contour flickering and stroke blurring. This occurs due to a 'stylization pose gap' between training and inference in stylization networks designed to preserve drawing styles in drawing-based 3D animation systems. The stylization pose gap denotes that input target poses used to train the stylization network are always in occlusion-free poses, while target poses encountered in an inference include diverse occlusions under dynamic motions. To this end, we propose Occlusion-robust Stylization Framework (OSF) for drawing-based 3D animation. We found that while employing object's edge can be effective input prior for guiding stylization, it becomes notably inaccurate when occlusions occur at inference. Thus, our proposed OSF provides occlusion-robust edge guidance for stylization network using optical flow, ensuring a consistent stylization even under occlusions. Furthermore, OSF operates in a single run instead of the previous two-stage method, achieving $2.4\times$ faster inference and $2.1\times$ less memory. Project is available at: github.io/Drawing-based-3D-Animation-page.

1. Introduction

Denoising diffusion models [10, 12, 47, 48] have reshaped the landscape of generative AI, leading to remarkable advancements [14, 23, 24, 41, 53, 64] in image, speech, and video generation. Image-to-3D diffusion models [28–30] have showcased the potential for creating 3D animations [32, 40, 68] from a single image. In this work, we focus here on 3D animation techniques for user-drawn images, referred

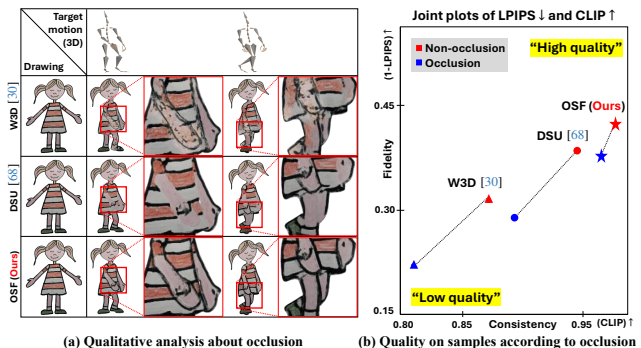


Figure 1. Quality deterioration in occlusion areas: (a) current systems exhibit stylistic inaccuracies in strokes and contours within overlapping body parts, and (b) animations with occlusion reduced consistency (CLIP [42]) and fidelity (LPIPS [67]) compared to ones without occlusion across diverse drawings in Amateur Drawings [46]. The CLIP measures consecutive frame similarity, and LPIPS measures perceptual differences between input and output.

to as drawing-based 3D animation. As shown in Figure 1 (a), drawing-based 3D animation systems take a reference hand-drawn image and a target 3D motion as inputs, and generate 3D animated drawings aligned with the target motion. In these systems, it is crucial to preserve artists' unique styles, such as contours and stroke patterns in output frames. However, synthesizing 3D structures inevitably causes undesirable distortions¹ on them. Thus, drawing-based 3D animation systems employ a stylization process using a neural network (referred to as a stylization network) fine-tuned on the input drawing to restore the distortions.

Recent advancements [30, 51, 68] of drawing-based 3D animation systems have demonstrated notable precision in preserving a drawing's identity within 3D animations. Nonetheless, these systems still suffer from quality deterioration in terms of drawing properties (*i.e.* contour and stroke) when dealing with occlusions in target poses. To be specific, in target poses where the object's body parts overlap, the outputs exhibit flickering contours and blurred internal stroke patterns. Figure 1 (a) presents examples of

¹The contour lines depicting the subject's shape become excessively thick, or the strokes conveying the artist's unique style become blurred.

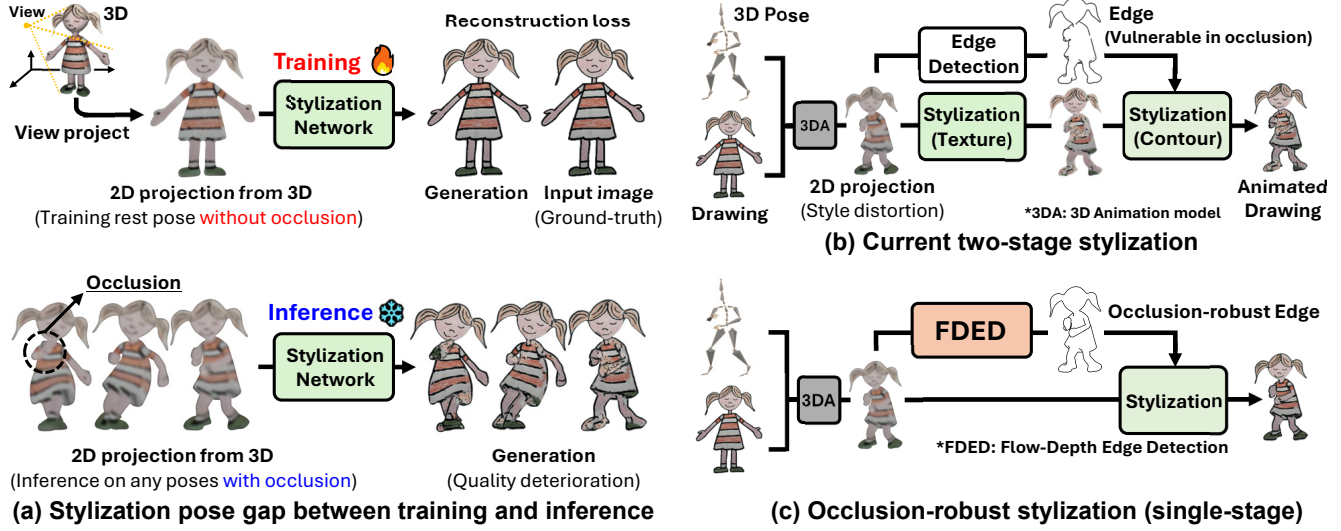


Figure 2. (a) Illustration of stylization pose gap between training and inference. The stylization network is fine-tuned on a single pose (simple occlusion-free pose), due to the unavailability of drawings about the same object’s other poses. During inference, the network encounters diverse occluded poses as unseen scenario, reducing stylization robustness. (b) Current two-stage stylization with edge guidance (vulnerable in occlusion) [30, 51, 68]. (c) Our proposed single-stage occlusion-robust stylization with flow-depth edge detection (FDED).

this issue: when the arm overlaps with the body, the arm’s texture and the surrounding clothing of the body are blurred (first row), and the contour line between overlapping parts intermittently appears and disappears (second row) as flickers. Furthermore, Figure 1 (b) presents quantitative evaluations of the generated animations in terms of consistency and fidelity for target motions with and without occlusion. Notably, motions with occlusion yield significantly lower quality compared to those without occlusion.

In fact, this quality deterioration stems from a ‘stylization pose gap’ between training and inference in the stylization network. The stylization pose gap refers to differences of the input target poses between training and inference of the stylization network. To be specific, shown in Figure 2 (a), the stylization network is fine-tuned (trained) to restore the original input drawing style onto a 2D projection of 3D animation². Here, the input drawing serves as ground-truth for the training, where its pose is relatively simple and free from occlusion to show the object’s entire body (*e.g.* rest pose in Figure 2 (a)). During inference, the stylization is applied to diverse poses, including those with occlusions. However, the network has not been trained to handle these occluded scenarios because ground-truth drawings for those conditions are unavailable³, consequently making it vulnerable in maintaining robust stylization under occlusions.

To this end, we propose Occlusion-robust Stylization Framework (OSF) to bridge the stylization pose gap between

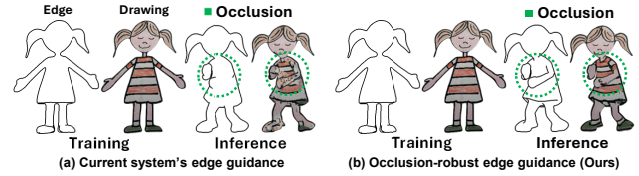


Figure 3. Resulting animated drawings and edges in training and inference: (a) current system’s edge and (b) occlusion-robust edge.

training and inference. Figure 2 (b) shows the current stylization framework for drawing-based 3D animation system. Given an input drawing and target 3D pose, 3D animation model (*i.e.* image-to-3D diffusion [30] and 3D rigging [3]) generates 3D structure following target pose, which is then projected into 2D for animation. Since the 2D projection exhibits style distortion, the stylization is performed on it. Here, edges extracted from the 2D projection are also used as auxiliary input, serving as an effective guideline for the stylization. As shown in Figure 3, we observed that the occlusion-free pose (*i.e.* rest pose) used in a training provides clear edges. However, at inference, occlusions across diverse poses obscure these edges, causing flickering and blurring within the output. To counter this, our proposed OSF in Figure 2 (c) incorporates flow-depth edge detection, which produces occlusion-robust edges to ensure consistent clarity and robust stylization under occlusion by recovering undetected edges using optical flow. Furthermore, current methods rely on a two-stage stylization process (texture to contour), resulting in unnecessary computational overhead and resource usage. In contrast, our framework operates in a single pass through edge-guided contrastive learning, achieving 2.4× faster inference and 2.1× less memory.

²For details of 3D animation, 3D diffusion model [30] builds 3D structure [37] and apply rigging [3] it on a skeleton to follow 3D target motion.

³We assume a case that only a single ground-truth drawing is available, as obtaining multiple drawings of the same object is not always feasible.

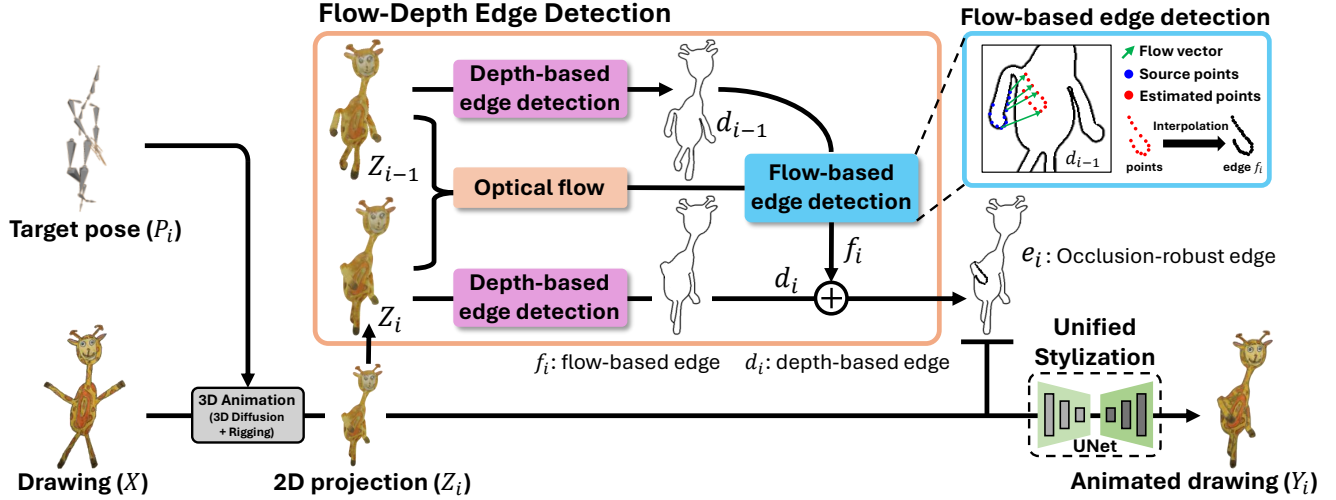


Figure 4. Occlusion-robust Stylization Framework (OSF) for drawing-based 3D animation. A 3D animation model processes a drawing X with the i -th target pose P_i to generate 3D structure following the target pose, which is rendered as a 2D projection Z_i for the animation. The proposed OSF stylizes the projection Z_i to final drawing animation Y_i , incorporating Flow-Depth Edge Detection (FDED), which produces an occlusion-robust edge e_i to guide robust stylization under unseen occluded poses. The edge e_i combines a depth-based edge d_i for unoccluded region and a flow-based edge f_i for occluded areas by reconstructing missing edges. Our method operates in a single stage, using a simple UNet-based [43] unified stylization network with our designed edge-guided contrastive learning.

2. Related Works

Image to 3D Animation. Image animation is a temporal sequence of shape deformations designed to ensure seamless transitions. Early works [1, 2, 15] investigated image-to-image transformations across poses and viewpoints, aided by geometric deformation [26, 49] and synthesis [7, 38]. Recently, 2D methods [45, 54, 57, 65, 66] have animated various objects but struggled to capture depth and perspective, prompting a shift to 3D animation [19, 30, 51]. Classic methods constructed 3D meshes from silhouettes and skeletons [4, 39, 44], while newer diffusion-based techniques [29, 30] generate multi-view images for optimized 3D reconstructions [17, 21, 37]. Rigging⁴ process [3, 20] then enabled animation of these meshes according to target motions. The 3D animations have been applied to diverse images, including human and animal images. However, hand-drawn images remain challenging: 3D augmentation distorts contours, and 2D projections (*i.e.* rendering) often cause blurry textures, especially under occlusion. To address this, we propose a robust stylization framework to maintain consistent quality under diverse occluded motions.

Edge Detection in 2D and 3D. Edge detection has been a fundamental building block for various computational 2D and 3D visions. In 2D, conventional methods (*e.g.* Sobel, Canny, and Laplacian) detect boundaries by analyzing gradient changes [5], while recent deep-learning approaches (*e.g.* Holistically-Nested Edge Detection [56]) leverage hierarchical feature extraction for more accurate edge maps.

Recent works have extended upto 3D data and predict geometric boundary using 3D contours [9], ridges [18] and neural network [27]. However, these edge detections rely on single frames, making them vulnerable to occlusions. By incorporating optical flow in consecutive frames, we provide more robust edge guidance for stylization in occlusion.

3. Method

Occlusion-robust Stylization Framework (OSF) in Figure 4 is proposed for drawing-based 3D animation. The 3D animation model (*i.e.* image-to-3D diffusion [30] with 3D rigging [3]) takes a drawing X and the i -th target 3D pose P_i to generate 3D animation, rendered as a 2D projection Z_i for visualizing it from user’s view point. 3D synthesis and animation often cause style distortions, and stylization addresses these issues in a lower-dimensional space (2D projection) as an intuitive solution. Formally, stylization takes the Z_i as input and predicts the animated drawing Y_i , where OSF provides occlusion-robust edge guidance e_i to enhance stylization robustness under occlusions as below:

$$Y_i = F(Z_i, e_i), \quad (1)$$

where F is a simple encoder-decoder network (*e.g.* U-Net [43]). The edge e_i can be utilized through various operations (*e.g.* concatenation, attention) within the network.⁵ We focus here on constructing the edge e (For simplicity, we omit subscript i). To be specific, OSF incorporates our designed Flow-Depth Edge Detection (FDED) to generate

⁴Creating a skeleton for 3D mesh to control its movements.

⁵We adopt a simple early fusion by concatenating it with Z_i , but it can be further enhanced with transformer-based interactions [22, 35, 60, 62].

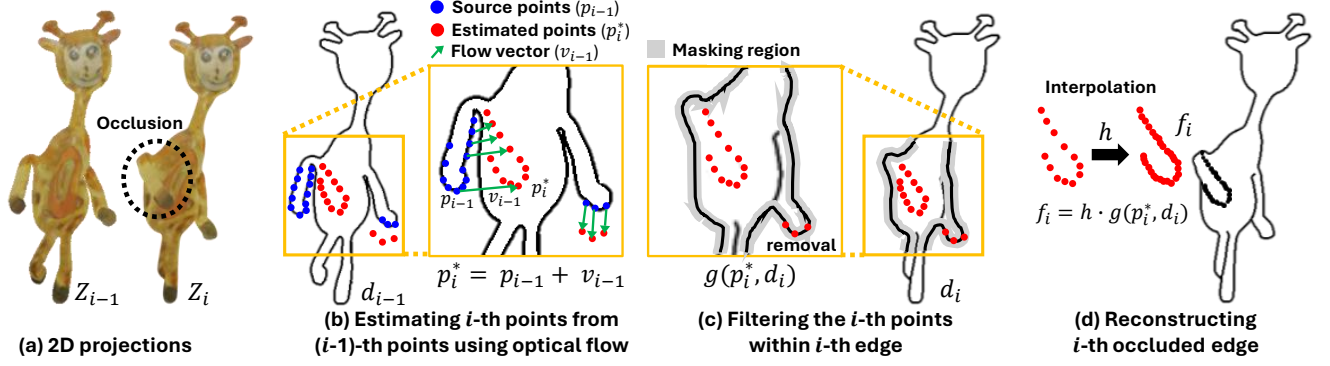


Figure 5. Illustration of flow-based edge detection. (a) shows sequential 2D projections Z_{i-1} and Z_i to compute optical flow v_{i-1} . (b) shows that the v_{i-1} is added to source points p_{i-1} to estimate their positions p_i^* in the next i -th frame. (c) shows that the points p_i^* are filtered into internal region of i -th edge. (d) shows that the remained points are interpolated to build edges. The g is the filtering operation to filter out estimated points p_i^* within the region of edge d_i and h is interpolation that builds flow-based edge f_i from the filtered points.

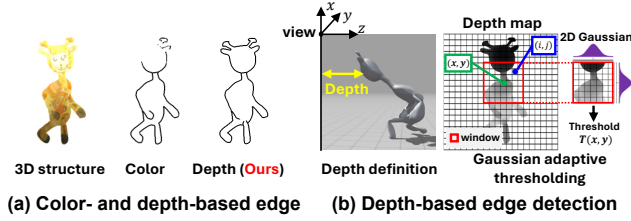


Figure 6. (a) shows the results of color- and depth-based edges with a light source placed on the head of a 3D object. (b) shows depth-based edge detection using Gaussian adaptive thresholding.

the occlusion-robust edge e . This edge can be separated into two parts: edge in occluded regions (e^o) and the other edge in unoccluded regions (e^u) as given below:

$$e = e^o \cup e^u. \quad (2)$$

To estimate the two edges, we propose depth-based edge detection for e^u and flow-based edge detection for e^o .

3.1. Depth-based edge detection

Depth-based edge detection aims to capture the unoccluded edge e^u in the 2D projection Z . While edges can be estimated by visual information (e.g. color) in Figure 6 (a), these edges become susceptible to brightness changes by the 3D object’s motions. To address this, we use the depth map (Figure 6(b)), measured by vertical distance between the viewpoint and the 3D mesh. To decide edges based on the depth map, we employed Gaussian adaptive thresholding [50], which is more responsive to local depth variations than global thresholding (e.g. Canny [5]). For each pixel $[x, y]$, an adaptive threshold $T[x, y]$ classifies edges as:

$$d[x, y] = \begin{cases} 1 & \text{if } D[x, y] > T[x, y] \\ 0 & \text{otherwise,} \end{cases} \quad (3)$$

producing a depth-based edge map d (1 for edges, 0 otherwise). Thus, we define the unoccluded edge as $e^u = d$.

$D[x, y]$ is the depth at position $[x, y]$. The threshold $T[x, y]$ is adaptively adjusted based on surrounding depth values within a specified window, assigning greater weight to regions closer to $[x, y]$ using Gaussian kernel weighting as:

$$T[x, y] = \frac{1}{R} \sum_{[i, j] \in \text{window}} G[i, j] \cdot D[x + i, y + j], \quad (4)$$

where $G[i, j] = \frac{1}{2\pi\sigma^2} \exp(-(i^2 + j^2)/(2\sigma^2))$ and constant $R = \sum_{[i, j] \in \text{window}} G[i, j]$. Window in Figure 6 (b) defines position $[i, j]$ in a square of width w centered at $[x, y]$.

3.2. Flow-based edge detection

Flow-based edge detection aims to capture occluded edges e^o in the projection Z . As shown in Figure 6 (a), depth-based edge detection alone can fail when body parts overlap (e.g. an arm over a torso) because their depths are too similar, causing flickering. To this end, as shown in Figure 4, we propose flow-based edge detection that leverages optical flow between the current frame Z_i and the previous frame $Z_{<i}$ (hereafter Z_{i-1} for convenience) to recover missing edges in occluded regions. To be specific, we obtain optical flow vector v_{i-1} using optical flow estimators [31, 52] between the Z_{i-1} and Z_i in Figure 5 (a). This vector v_{i-1} is a 2-dimensional vector connecting corresponding points between Z_{i-1} and Z_i to estimate optical flow. As shown in Figure 5 (b), we define these corresponding points of Z_{i-1} that lie on the edge⁶ d_{i-1} as source points p_{i-1} (i.e. blue points). To estimate the positions of the source points in the i -th frame, we shift them by v_{i-1} , computed as $p_i^* = p_{i-1} + v_{i-1}$. As we focus on finding undetected edges by occlusion, as shown in Figure 5 (c), we retain only p_i^* that

⁶To focus on the problem of predicting new edges based on existing ones, we assume that the chosen edges d_{i-1} are well-constructed without occlusion. Occlusion-free frames can be identified by checking the overlaps between joint points of 3D coordinate space or can be automatically detected using deterministic retrievals [33, 36, 58, 59, 61, 63].

lie within the interior of d_i . Finally, the remaining points are interpolated to construct flow-based edge f in Figure 5 (d). This can be formulated on i -th frame as:

$$f_i = h \cdot g(p_i^*, d_i), \quad (5)$$

where the g is a filter selecting valid points inside the depth-based edge d_i , and h is an interpolator constructing edge from points (we use dilation for efficiency). Therefore, we finally define the occluded edge as $e^o = f$.

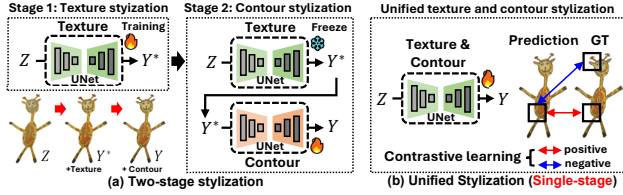


Figure 7. (a) two-stage stylization framework and (b) our single-stage unified stylization with edge-guided contrastive learning.

3.3. Unified Stylization Network

Since multiple drawing images for various poses of an object are not always available, we assume stylization operates in the most general scenario, where each object provides only a single drawing image for training. Figure 7 (a) shows the existing two-stage method [68], where texture stylization produces an intermediate output Y^* from Z , followed by contour stylization to produce final Y . Although this method improves contour training despite the limited contour (compared to texture) in the drawing, it uses extra labels (ground-truth Y^*) and increases both memory and time costs. To address these, Figure 7 (b) shows our unified (single-stage) stylization network (USNet), which learns all drawing properties at once. As contours are relatively sparse compared to textures, we introduce edge-guided contrastive learning for more discriminative contour stylization. To be specific, we first apply patch-wise reconstruction loss for stylization by predicting original drawing-style input frame $X \in \mathbb{R}^{N \times 3}$ from the 2D projection frame $Z \in \mathbb{R}^{N \times 3}$ as $\mathcal{L}_{recon} = \sum_{i=1}^N (Y_i - X_i)^2$, where Y is the stylization output in Equation (1), N is the number of patches, and each patch has 3 channels (RGB). We then extend this to contrastive learning, enforcing an inequality as:

$$\cos(Y_j, X_j) > \cos(Y_j, X_k), \quad \forall k \neq j, \quad (6)$$

so that the cosine similarity between Y_j and its ground-truth X_j for contour patches exceeds that of Y_j with other patches X_k as hard negatives. The X_k are randomly sampled from patches containing the occlusion-robust edge e . This inequality is ensured by contrastive ranking loss as below:

$$\mathcal{L}_c = \sum_j \max(0, \cos(Y_j, X_k) - \cos(Y_j, X_j) + \delta), \quad (7)$$

where $\delta = 0.1$ maintains a margin. Our final loss for USNet combines the two terms as $\mathcal{L} = \mathcal{L}_{recon} + \mathcal{L}_c$.

4. Experiment

4.1. Experimental Settings

Implementation Details. We use RAFT [52] for optical flow. Hyperparameters were chosen by study in Table 2, confirming $w = 9$ and dilation-based interpolation for h . For image-to-3D diffusion, we use Wonder3D [30] and Adobe Mixamo⁷ skeleton of 65 joints for automatic rigging.

Data and Baselines. We selected 120 test characters from the Amateur Drawings [46], following the same set in [68]. Training (fine-tuning) uses a single pose given by the characters, while inference applies stylization to various poses. For inference, each character is given 20 non-occluded motions (non-occlusion set) and 20 occluded motions (occlusion set)⁸ for a total of 4800 samples. We also prepare 80 separate characters for a validation set to investigate the effectiveness of occlusion. The 3D animation system includes non-stylization (DreamGaussian [51], Wonder3D [30]) and stylization methods (DrawingSpinUp [68], OSF).

4.2. Evaluation Metrics

We evaluate each drawing animation in terms of overall texture and contour by separating the two. For texture quality, we measure temporal consistency (via CLIP [42] similarity, SSIM [55] for smoothness, and FID [11] for naturalness) and fidelity (via LPIPS [67] to assess perceptual preservation of object identity). We construct ground-truth distributions using input drawings, and then measure FID by comparing distributions of the generated drawings. For contour quality, we extract contours using the method in [68] and evaluate their temporal consistency with CLIP, SSIM, and FID. All metrics are averaged over 10 runs with different seeds and Human evaluation of preferences is performed.

4.3. Experimental Results

Qualitative Comparisons. Figure 8 presents the 3D animation results of drawings across different 3D animation systems. Our comparison evaluates dynamic motions, such as dancing and sitting, applied to a variety of character drawings, ranging from humans to animals. Wonder3D [30] and DreamGaussian [51] are image-to-3D models, applied to drawing-based 3D animation task without including a stylization process. Without a stylization process, these models exhibit distortions in the drawing style, such as thickened contours and blurred stroke patterns. DrawingSpinUp (DSU) [68] and our proposed Occlusion-robust Stylization Framework (OSF) are stylization-based models designed for 3D animation of drawings, performing stylization to preserve the original style of the input image. Both models

⁷we use online platform (<https://www.mixamo.com>), but other automatic joint estimation methods [6, 13] can also be applicable.

⁸Our supplemental material provides all Mixamo motion names

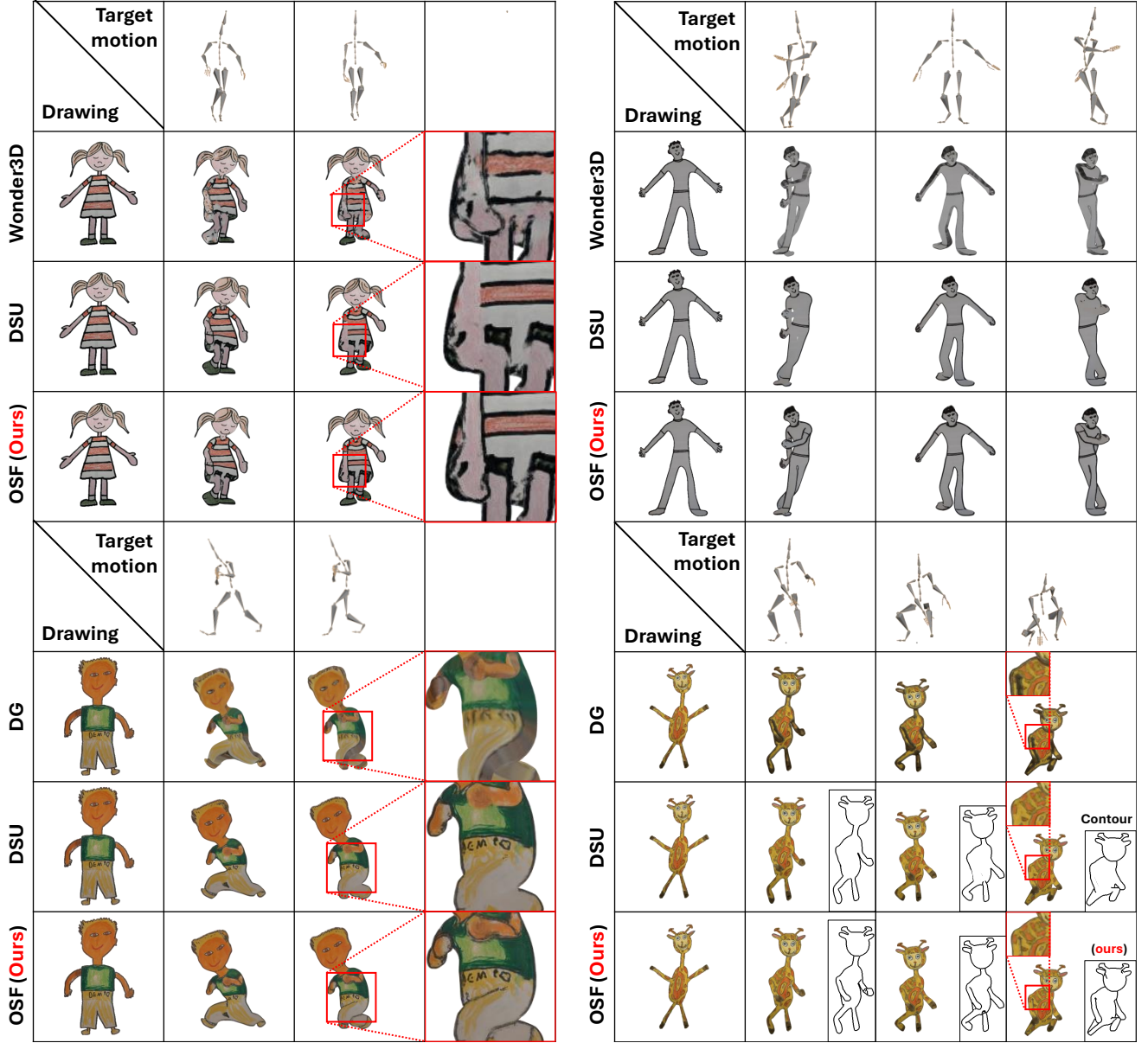


Figure 8. Qualitative comparisons across different drawing-based 3D animation models. The red box highlights a zoomed view of the stylization applied to animated drawing objects. The contour in the bottom right is extracted by the contour estimator from [68]. Appendix also provides more qualitative results. DSU: DrawingSpinUp, DG: DreamGaussian.

retain the style of the input image well. However, DSU tends to lose contour details in occluded body parts (*i.e.* red box), leading to flickering. This issue is particularly evident in the contour, as shown in the box visualizing extracted giraffe’s contours in bottom-right of Figure 8. OSF demonstrates robust contour preservation, even in the presence of occlusions occurring during various motions.

Quantitative Results. Table 1 presents quantitative results on different animation models. We evaluated the animation quality on motions without occlusion (*i.e.* non-

occlusion set) and with occlusion (*i.e.* occlusion set). As contours are a key attribute defining an object’s shape, we extract them for more detailed evaluation. The proposed FDED can serve as an additional condition for any stylization model, such that we also integrate it into DSU, observing steady gains in both stylization baselines (DSU and USNet). With the application of FDED, both consistency and fidelity have improved, with consistency enhancements largely attributed to improved contour stability. Our proposed USNet resolves the complex two-stage inference structure of DSU, while its performance is also comparable to or better than

Table 1. Quantitative evaluations on recent drawing-based 3D animation models, reported in a format of (non-occlusion set / occlusion set). DSU: DrawingSpinUp, USNet: Unified Stylization Network, FDED: Flow-Depth Edge Detection. (OSF = USNet + FDED).

Method	Texture				Contour			Human
	CLIP \uparrow	SSIM \uparrow	FID \downarrow	LPIPS \downarrow	CLIP \uparrow	SSIM \uparrow	FID \downarrow	
DreamGaussian [51]	0.903/0.898	0.842/0.813	554/592	0.725/0.798	0.894/0.875	0.828/0.786	441/513	0.01
Wonder3D [30]	0.914/0.891	0.844/0.816	531/586	0.714/0.782	0.906/0.886	0.832/0.801	432/476	0.03
DSU [68]	0.958/0.927	0.889/0.842	313/368	0.629/0.692	0.941/0.919	0.879/0.830	215/240	0.35
DSU + FDED (Ours)	0.971/0.962	0.903/0.881	297/331	0.601/0.637	0.978/0.974	0.936/0.910	191/199	-
USNet (Ours)	0.963/0.936	0.896/0.851	308/361	0.612/0.685	0.946/0.925	0.882/0.832	212/236	-
USNet + FDED (Ours)	0.974/0.963	0.910/0.888	293/325	0.586/0.622	0.982/0.977	0.938/0.914	188/195	0.61

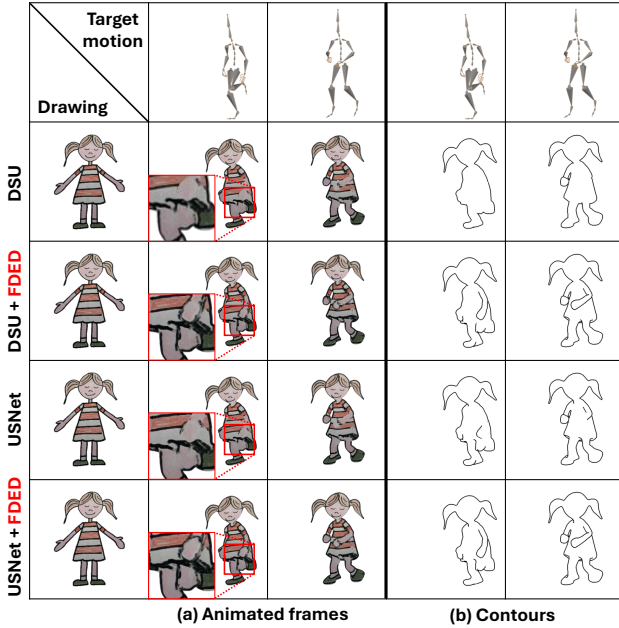


Figure 9. Effectiveness of FDED on stylization models, showing (a) resulting frames and (b) their contours with and without FDED.

DSU due to the edge-guided contrastive learning. The effectiveness of the contrastive learning is also validated in Figure 10 about learning optimization and qualitative comparisons.

4.4. Ablation Study

Effectiveness of FDED. Figure 9 demonstrates the effectiveness by applying FDED to stylization-based animation models (DSU and USNet). Without FDED (Figure 9 (b)), both struggle to render contours in occluded regions accurately, whereas FDED-integrated versions preserve contours more clearly. Table 2 shows an ablation study on depth-based edge (for unoccluded edge) and flow-based edge (for occluded edge), along with hyperparameter variations. From the first section, providing guidance edges significantly boosts performance, especially with flow-based edges, revealing the vulnerability of existing methods to occlusion. This improvement mainly stems from enhanced

Table 2. Ablation study on the main modules of FDED and their hyperparameters under validation occlusion set (D: depth-based edge, F: flow-based edge, w : window size of adaptive thresholding, h : edge reconstruction, and spline: B-spline interpolation).

Method	Texture		Contour	
	CLIP \uparrow	SSIM \uparrow	CLIP \uparrow	SSIM \uparrow
w/o FDED	0.946	0.873	0.929	0.849
w/ FDED (D)	0.956	0.887	0.941	0.876
w/ FDED (D + F)	0.971	0.906	0.980	0.934
D + F (w : 7)	0.968	0.905	0.977	0.931
D + F (w : 9)	0.971	0.906	0.980	0.934
D + F (w : 11)	0.970	0.904	0.978	0.932
D + F (w : 13)	0.968	0.903	0.977	0.930
D + F (h : spline)	0.969	0.906	0.977	0.930
D + F (h : dilation)	0.971	0.906	0.980	0.934

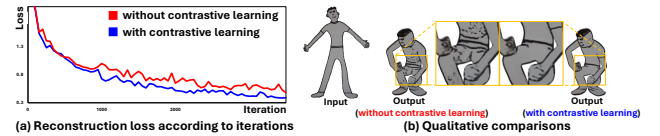


Figure 10. Ablation study on edge-guided contrastive learning. (a) demonstrates enhanced reconstruction loss, integrated with our contrastive learning and (b) shows qualitative comparisons of this.

contour quality. We also present qualitative results of this according to different guidance edges in Figure 11. In the second section, we also examine changes in the window size for Gaussian adaptive thresholding used in depth-based edge detection and find it to be relatively insensitive, presumably due to the continuous nature of the drawing. The third section covers two approaches for flow-based edges: B-spline interpolation [8] and dilation. Dilation produces better results, as it mitigates the scarcity of source points.

Effectiveness of contrastive learning. Figure 10 shows the effectiveness of edge-guided contrastive learning in terms of reconstruction loss and outcomes. Applying contrastive learning leads to a faster drop in reconstruction loss

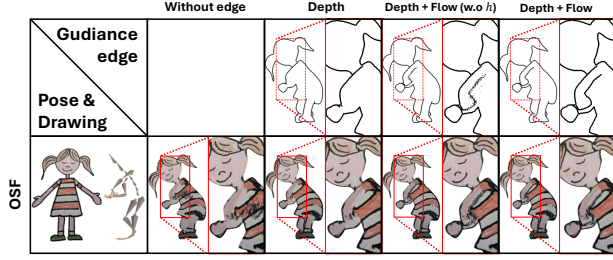


Figure 11. Ablation study about different edges in FDED. The red box shows the zoomed results in the occluded areas. The h denotes edge interpolation in flow-based edge detections.

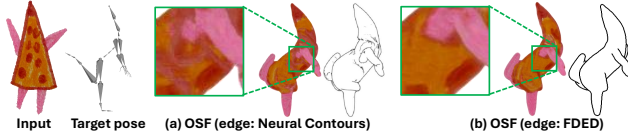


Figure 12. Comparison results about OSF with (a) 3D edge detection [27] and (b) Flow-Depth Edge Detection (FDED, Ours).

with fewer iterations (Figure 10 (a)), ultimately converging to a lower value. Consequently, the stylization provides sharper textures and contours, shown in Figure 10 (b).

Ablation study on edge guidance. Figure 11 shows qualitative animation results using various edges obtained from FDED. With USNet as the base stylization network, we test four edge configurations: (1) none, (2) depth-only, (3) depth + flow (no edge interpolation), and (4) depth + flow (with interpolation). Without edge guidance, stylization is highly unstable, producing blurred contours and strokes. We highlight occluded regions in a zoomed view (red box). Depth-only edges capture the overall silhouette but miss contours in occluded regions, causing incomplete animations. Incorporating flow-based edges addresses occluded contours but appears scattered when interpolation is absent. By adding the edge interpolation, flow-based edges produce sharper contours, improving the final result. Figure 12 presents the comparison results of 3D edge detection [27]. Although the drawn 3D edges convey finer detail and three-dimensional depth, they were unsuitable for stylization guidance. Because stylization misinterprets them as contours, the resulting texture merges with them, causing a blurry appearance.

Robustness analysis on occlusion. Figure 13 presents the occlusion robustness evaluation of the stylization-based 3D animation system, comparing results with and without the application of FDED. We define occlusion rate by calculating a reduction in the object’s visible area⁹ compared to the original input drawing pose. The giraffe character in Figure 13 illustrates example poses at varying occlusion levels. As occlusion increases, the output animation quality (*i.e.* blue)

⁹Samples only with visible occlusion are considered

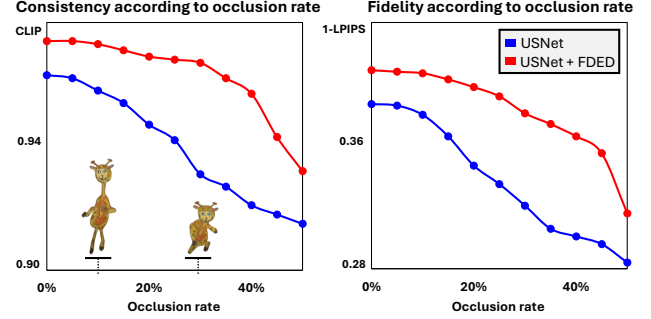


Figure 13. Robustness analysis of FDED on stylization-based 3D animation model (USNet). The characters illustrate examples of varying occlusion levels based on occlusion rate.

Table 3. Inference time and memory usage across stylization networks (excluding 3D diffusion model for isolated analysis).

Method	seconds/frame	memory
DSU [68]	0.276	11.62 GB
DSU [68] + FDED (Ours)	0.282	11.91 GB
USNet (Ours)	0.108	4.81 GB
USNet + FDED (Ours)	0.115	5.05 GB

of the stylization system significantly deteriorates. Applying FDED to the system enhances robustness, extending the range that ensures high-quality output in both consistency and fidelity. However, with occlusion above 40%, quality deteriorates, due to the complexity of multiple occluded areas, obscuring contours and strokes. For this, quantization [34] into patch-wise encoding may enhance robustness.

Computational Complexity Analysis. Table 3 presents a computational analysis of stylization processing in terms of inference time and resource usage for the DSU and USNet baselines. The proposed USNet resolves the DSU’s two-stage complex inference process into a single-stage process, achieving over twice the efficiency in inference time and memory usage. Furthermore, the integration of FDED introduces minimal overhead in inference time. This can be improved with diffusion acceleration techniques [16, 25].

5. Conclusion

This paper addresses the challenge of generating 3D animations from hand-drawn images while preserving stylistic details such as rough contours and strokes. Existing models suffer from quality deterioration, especially under occlusion, due to a stylization pose gap between the unoccluded poses used in a training and the dynamic poses encountered in an inference. To this end, we introduce Occlusion-robust Stylization Framework (OSF), which improves robustness of stylization under occlusion via our designed flow-depth edge detection. Furthermore, OSF operates in a single pass, achieving faster inference and reducing memory usage.

Acknowledgements

This work was partly supported by Institute for Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.RS-2021-II211381, Development of Causal AI through Video Understanding and Reinforcement Learning, and Its Applications to Real Environments) and partly supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No. RS-2022-II0951, Development of Uncertainty-Aware Agents Learning by Asking Questions)

References

- [1] Kfir Aberman, Rundi Wu, Dani Lischinski, Baoquan Chen, and Daniel Cohen-Or. Learning character-agnostic motion for motion retargeting in 2d. *arXiv preprint arXiv:1905.01680*, 2019. 3
- [2] Badour Albahar, Jingwan Lu, Jimei Yang, Zhixin Shu, Eli Shechtman, and Jia-Bin Huang. Pose with style: Detail-preserving pose-guided image synthesis with conditional stylegan. *ACM Transactions on Graphics (TOG)*, 40(6):1–11, 2021. 3
- [3] Ilya Baran and Jovan Popović. Automatic rigging and animation of 3d characters. *ACM Transactions on graphics (TOG)*, 26(3):72–es, 2007. 2, 3
- [4] Philip Buchanan, Ramakrishnan Mukundan, and Michael Doggett. Automatic single-view character model reconstruction. In *Proceedings of the international symposium on sketch-based interfaces and modeling*, pages 5–14, 2013. 3
- [5] John Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6):679–698, 1986. 3, 4
- [6] Hongsuk Choi, Gyeongsik Moon, Ju Yong Chang, and Kyoung Mu Lee. Beyond static features for temporally consistent 3d human pose and shape from a video. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1964–1973, 2021. 5
- [7] David Cohen-Steiner, Pierre Alliez, and Mathieu Desbrun. Variational shape approximation. In *ACM SIGGRAPH 2004 Papers*, pages 905–914. 2004. 3
- [8] C De Boor. A practical guide to splines. *Springer-Verlag google schola*, 2:4135–4195, 1978. 7
- [9] Doug DeCarlo, Adam Finkelstein, Szymon Rusinkiewicz, and Anthony Santella. Suggestive contours for conveying shape. *ACM Trans. Graph.*, 22(3):848–855, 2003. 3
- [10] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021. 1
- [11] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017. 5
- [12] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 1
- [13] Ji Woo Hong, Sunjae Yoon, Junyeong Kim, and Chang D Yoo. Joint path alignment framework for 3d human pose and shape estimation from video. *IEEE Access*, 11:43267–43275, 2023. 5
- [14] Ji Woo Hong, Tri Ton, Trung X Pham, Gwanhyeong Koo, Sunjae Yoon, and Chang D Yoo. Ita-mdt: Image-timestep-adaptive masked diffusion transformer framework for image-based virtual try-on. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 28284–28294, 2025. 1
- [15] Alexander Hornung, Ellen Dekkers, and Leif Kobbelt. Character animation from 2d pictures and 3d motion data. *ACM Transactions on Graphics (ToG)*, 26(1):1–es, 2007. 3
- [16] Rongjie Huang, Max WY Lam, Jun Wang, Dan Su, Dong Yu, Yi Ren, and Zhou Zhao. Fastdiff: A fast conditional diffusion model for high-quality speech synthesis. *arXiv preprint arXiv:2204.09934*, 2022. 8
- [17] Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, et al. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 559–568, 2011. 3
- [18] Tilke Judd, Frédo Durand, and Edward Adelson. Apparent ridges for line drawing. *ACM Trans. Graph.*, 26(3):19–es, 2007. 3
- [19] Akash Karthikeyan, Robert Ren, Yash Kant, and Igor Gilitschenski. Avatarone: Monocular 3d human animation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3647–3657, 2024. 3
- [20] Ladislav Kavan, Steven Collins, Jiří Žára, and Carol O’Sullivan. Skinning with dual quaternions. In *Proceedings of the 2007 symposium on Interactive 3D graphics and games*, pages 39–46, 2007. 3
- [21] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023. 3
- [22] Junyeong Kim, Sunjae Yoon, Dahyun Kim, and Chang D Yoo. Structured co-reference graph attention for video-grounded dialogue. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1789–1797, 2021. 3
- [23] Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. *arXiv preprint arXiv:2009.09761*, 2020. 1
- [24] Gwanhyeong Koo, Sunjae Yoon, Ji Woo Hong, and Chang D Yoo. Flexiedit: Frequency-aware latent refinement for enhanced non-rigid editing. In *European Conference on Computer Vision*, pages 363–379. Springer, 2024. 1
- [25] Gwanhyeong Koo, Sunjae Yoon, and Chang D Yoo. Wavelet-guided acceleration of text inversion in diffusion-based image editing. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4380–4384. IEEE, 2024. 8
- [26] Gwanhyeong Koo, Sunjae Yoon, Younghwan Lee, Ji Woo Hong, and Chang D Yoo. Flowdrag: 3d-aware drag-based image editing with mesh-guided deformation vector flow fields. *arXiv preprint arXiv:2507.08285*, 2025. 3

- [27] Difan Liu, Mohamed Nabail, Aaron Hertzmann, and Evangelos Kalogerakis. Neural contours: Learning to draw lines from 3d shapes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5428–5436, 2020. 3, 8
- [28] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9298–9309, 2023. 1
- [29] Yuan Liu, Cheng Lin, Zijiao Zeng, Xiaoxiao Long, Lingjie Liu, Taku Komura, and Wenping Wang. Syncdreamer: Generating multiview-consistent images from a single-view image. *arXiv preprint arXiv:2309.03453*, 2023. 3
- [30] Xiaoxiao Long, Yuan-Chen Guo, Cheng Lin, Yuan Liu, Zhiyang Dou, Lingjie Liu, Yuexin Ma, Song-Hai Zhang, Marc Habermann, Christian Theobalt, et al. Wonder3d: Single image to 3d using cross-domain diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9970–9980, 2024. 1, 2, 3, 5, 7
- [31] Bruce D Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *IJCAI’81: 7th international joint conference on Artificial intelligence*, pages 674–679, 1981. 4
- [32] Zhongjin Luo, Shengcai Cai, Jinguo Dong, Ruibo Ming, Liangdong Qiu, Xiaohang Zhan, and Xiaoguang Han. Rabbit: Parametric modeling of 3d biped cartoon characters with a topological-consistent dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12825–12835, 2023. 1
- [33] Tung M Luu, Donghoon Lee, and Chang D Yoo. Predictive coding for decision transformer. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7469–7476. IEEE, 2024. 4
- [34] Tung M Luu, Thanh Nguyen, Tee Joshua Tian Jin, Sungwoon Kim, and Chang D Yoo. Mitigating adversarial perturbations for deep reinforcement learning via vector quantization. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 595–602. IEEE, 2024. 8
- [35] Tung Minh Luu, Younghwan Lee, Donghoon Lee, Sunho Kim, Min Jun Kim, and Chang D Yoo. Enhancing rating-based reinforcement learning to effectively leverage feedback from large vision-language models. *arXiv preprint arXiv:2506.12822*, 2025. 3
- [36] Minuk Ma, Sunjae Yoon, Junyeong Kim, Youngjoon Lee, Sunghun Kang, and Chang D Yoo. Vlanet: Video-language alignment network for weakly-supervised video moment retrieval. In *European conference on computer vision*, pages 156–171. Springer, 2020. 4
- [37] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 2, 3
- [38] Pascal Müller, Peter Wonka, Simon Haegler, Andreas Ulmer, and Luc Van Gool. Procedural modeling of buildings. In *ACM SIGGRAPH 2006 Papers*, pages 614–623. 2006. 3
- [39] Andrew Nealen, Takeo Igarashi, Olga Sorkine, and Marc Alexa. Fibermesh: designing freeform surfaces with 3d curves. In *ACM SIGGRAPH 2007 papers*, pages 41–es. 2007. 3
- [40] Hao-Yang Peng, Jia-Peng Zhang, Meng-Hao Guo, Yan-Pei Cao, and Shi-Min Hu. Charactergen: Efficient 3d character generation from single images with multi-view pose canonicalization. *ACM Transactions on Graphics (TOG)*, 43(4): 1–13, 2024. 1
- [41] Trung X Pham, Tri Ton, and Chang D Yoo. Mdsngen: Fast and efficient masked diffusion temporal-aware transformers for open-domain sound generation. *arXiv preprint arXiv:2410.02130*, 2024. 1
- [42] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 1, 5
- [43] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention—MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015. 3
- [44] Ryan Schmidt, Brian Wyvill, Mario Costa Sousa, and Joaquim A Jorge. Shapeshop: Sketch-based solid modeling with blobtrees. In *ACM SIGGRAPH 2007 courses*, pages 43–es. 2007. 3
- [45] Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. First order motion model for image animation. *Advances in neural information processing systems*, 32, 2019. 3
- [46] Harrison Jesse Smith, Qingyuan Zheng, Yifei Li, Somya Jain, and Jessica K. Hodgins. A method for animating children’s drawings of the human figure. *ACM Trans. Graph.*, 42(3), 2023. 1, 5
- [47] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 1
- [48] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020. 1
- [49] Olga Sorkine, Daniel Cohen-Or, Yaron Lipman, Marc Alexa, Christian Rössl, and H-P Seidel. Laplacian surface editing. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 175–184, 2004. 3
- [50] Richard Szeliski. *Computer vision: algorithms and applications*. Springer Nature, 2022. 4
- [51] Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. *arXiv preprint arXiv:2309.16653*, 2023. 1, 2, 3, 5, 7
- [52] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Computer Vision—ECCV 2020*:

- 16th European Conference, Glasgow, UK, August 23–28, 2020, *Proceedings, Part II 16*, pages 402–419. Springer, 2020. 4, 5
- [53] Tri Ton, Ji Woo Hong, and Chang D Yoo. Taro: Timestep-adaptive representation alignment with onset-aware conditioning for synchronized video-to-audio synthesis. *arXiv preprint arXiv:2504.05684*, 2025. 1
- [54] Tan Wang, Linjie Li, Kevin Lin, Yuanhao Zhai, Chung-Ching Lin, Zhengyuan Yang, Hanwang Zhang, Zicheng Liu, and Lijuan Wang. Disco: Disentangled control for realistic human dance generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9326–9336, 2024. 3
- [55] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 5
- [56] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. In *Proceedings of the IEEE international conference on computer vision*, pages 1395–1403, 2015. 3
- [57] Zhongcong Xu, Jianfeng Zhang, Jun Hao Liew, Hanshu Yan, Jia-Wei Liu, Chenxu Zhang, Jiashi Feng, and Mike Zheng Shou. Magicanimate: Temporally consistent human image animation using diffusion model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1481–1490, 2024. 3
- [58] Sunjae Yoon, Dahyun Kim, Ji Woo Hong, Junyeong Kim, Kookhoi Kim, and Chang D Yoo. Weakly-supervised moment retrieval network for video corpus moment retrieval. In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 534–538. IEEE, 2021. 4
- [59] Sunjae Yoon, Ji Woo Hong, Eunseop Yoon, Dahyun Kim, Junyeong Kim, Hee Suk Yoon, and Chang D Yoo. Selective query-guided debiasing for video corpus moment retrieval. In *European Conference on Computer Vision*, pages 185–200. Springer, 2022. 4
- [60] Sunjae Yoon, Eunseop Yoon, Hee Suk Yoon, Junyeong Kim, and Chang D Yoo. Information-theoretic text hallucination reduction for video-grounded dialogue. *arXiv preprint arXiv:2212.05765*, 2022. 3
- [61] Sunjae Yoon, Ji Woo Hong, Soohwan Eom, Hee Suk Yoon, Eunseop Yoon, Daehyeok Kim, Junyeong Kim, Chanwoo Kim, and Chang D Yoo. Counterfactual two-stage debiasing for video corpus moment retrieval. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023. 4
- [62] Sunjae Yoon, Dahyun Kim, Eunseop Yoon, Hee Suk Yoon, Junyeong Kim, and Chnag D Yoo. Hear: Hearing enhanced audio response for video-grounded dialogue. *arXiv preprint arXiv:2312.09736*, 2023. 3
- [63] Sunjae Yoon, Gwanhyeong Koo, Dahyun Kim, and Chang D Yoo. Scanet: Scene complexity aware network for weakly-supervised video moment retrieval. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 13576–13586, 2023. 4
- [64] Sunjae Yoon, Gwanhyeong Koo, Ji Woo Hong, and Chang D Yoo. Dni: Dilutional noise initialization for diffusion video editing. In *European Conference on Computer Vision*, pages 180–195. Springer, 2024. 1
- [65] Sunjae Yoon, Gwanhyeong Koo, Geonwoo Kim, and Chang D Yoo. Frag: Frequency adapting group for diffusion video editing. *arXiv preprint arXiv:2406.06044*, 2024. 3
- [66] Sunjae Yoon, Gwanhyeong Koo, Younghwan Lee, and Chang Yoo. Tpc: Test-time procrustes calibration for diffusion-based human image animation. *Advances in Neural Information Processing Systems*, 37:118654–118677, 2024. 3
- [67] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 1, 5
- [68] Jie Zhou, Chufeng Xiao, Miu-Ling Lam, and Hongbo Fu. Drawingspinup: 3d animation from single character drawings. *arXiv preprint arXiv:2409.08615*, 2024. 1, 2, 5, 6, 7, 8