

Recovering Dynamic 3D Sketches from Videos

Supplementary Material

A. Methodological Details

A.1. Point Cloud Rasterization

To rasterize the point cloud to an image plane in Sec. 3.2, we first project the 3D points $P_{3D} = \{P_i\}$ onto the 2D space $P_{2D} = \{\tilde{P}_i\}$. For each pair of i -th normalized grid point and j -th point of the normalized point cloud \tilde{P}_{2D} to image dimensions, we use the Gaussian function to compute a rendered intensity J_{ij} :

$$J_{ij} = \exp\left(-\frac{D_{ij}^2}{2\sigma_j^2}\right), \quad (13)$$

where D_{ij} is the Euclidean distance between two points and σ_j indicates the point size factor that controls the contribution area of each point.

We dynamically adjust σ_j based on the depth of the point $\{d_j\}$ to account for perspective projection effects. Points farther from the camera are rendered with smaller sizes, following standard 3D rendering principles. Using the normalized depth $\{\hat{d}_j\} = \{\frac{d_j - d_{min}}{d_{min} - d_{max}}\}$, each point size σ_j is computed as:

$$\sigma_j = \frac{\mu}{0.5\mu \min(W, H)} \times \hat{d}_j \times \beta, \quad (14)$$

where μ and β denote the scaling and deblurring factor, and W, H are the image width and height. We set $\mu = 10$ and $\beta = 0.5$.

We aggregate the Gaussian contributions from all point cloud points to each grid point to generate the final rendered image. The intensity value for each pixel is computed by summing these contributions. We then normalize the intensities by dividing by the maximum value, ensuring the final image $\mathcal{J} \in \mathbb{R}^{H \times W}$ values fall within an appropriate range for visualization or processing. This process can be expressed as the following equation:

$$\mathcal{J}_i = \frac{\sum_{j=1}^M R_{ij}}{\max(\sum_{j=1}^M R_{ij})}. \quad (15)$$

The results can be shown in Figs. C and F. Note that each guidance view in Fig. C is rasterized into a 100×100 resolution image, which represents the actual resolution used for generating motion guidance in synthetic scenes.

A.2. Effect of the Suppression Function $\xi(\cdot)$

As described in Sec. 3.3, we adjust the suppression function $\xi(\cdot)$ to prevent unintended stroke movements in sketch synthesis. Figure A shows the effect of this function. We

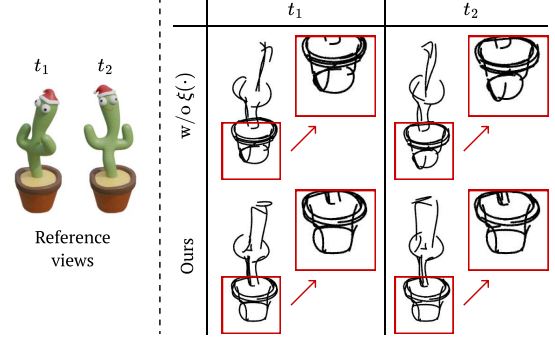


Figure A. Effectiveness of the function for suppression $\xi(\cdot)$. Without motion suppression, we observe noisy stroke movement at different time steps (t_1 and t_2) even if there are no motions.

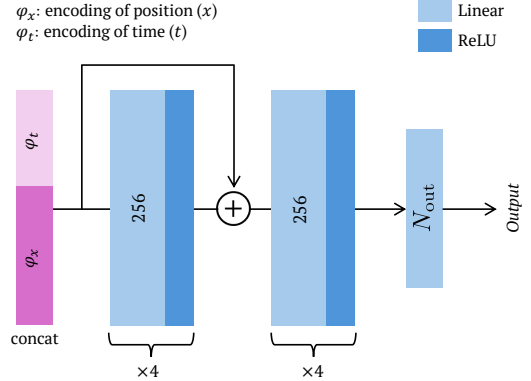


Figure B. Network architecture. All networks in the framework share the same architecture. N_{out} indicates the dimension of the output, which is 4 in \mathcal{M}_R , and 3 in the others.

observe that the model struggles to suppress undesired stroke movements even when no motion occurs. This demonstrates that our full approach achieves higher performance in extracting core motions.

B. Implementation Details

B.1. Network Architecture

Our network architecture, illustrated in Fig. B follows a consistent MLP structure across all components in Sec. 3.2 and 3.3, adopting a similar design to that proposed by [34]. Input is the concatenation of positional encoding of time φ_t and positions φ_x , and each linear layer, except for the final layer, outputs a 256-dimensional feature vector. The network \mathcal{M}_R yields outputs in $\mathbb{R}^{N \times 4}$, while other networks produce output vectors in $\mathbb{R}^{N \times 3}$. \mathcal{M}_R outputs quaternions for each stroke's rotation, which are subsequently converted to rotation matrices for stroke deformation.

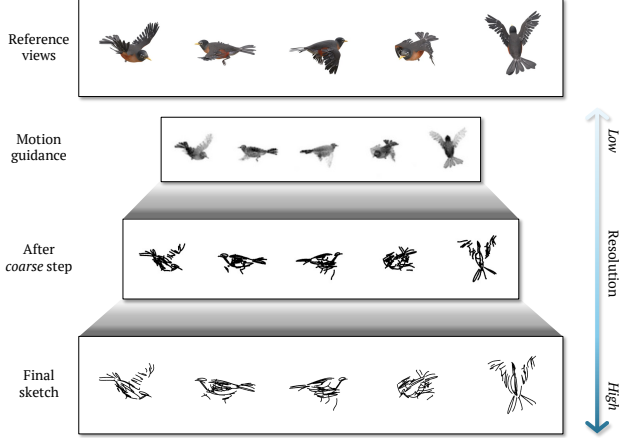


Figure C. Different resolution through processing stages. Our method gradually increases resolution according to stages to compute the location and deformation of strokes.

B.2. Optimization Details

The learning parameters differ between reconstructing synthetic datasets and real-world scenes. For synthetic scenes, we set the frequency value $L = 10$ for both temporal and spatial positional encoding. For real scenes presented in Fig. 9, we use $L = 8$ for temporal and $L = 10$ for spatial encoding. Learning rate values are also slightly different in synthetic and real scenes. For the former, in drawing process as described in Sec. 3.3, we apply a learning rate of 5.0×10^{-4} to \mathcal{M}_T and \mathcal{M}_R , and 1.0×10^{-3} to all other parameters. For the latter, during sketch reconstruction, we apply a learning rate of 5.0×10^{-4} to the canonical stroke positions, 1.0×10^{-4} to \mathcal{M}_T and \mathcal{M}_R , and 2.5×10^{-4} to \mathcal{M}_L . We set the learning rates $lr_{pcd} = 1.0 \times 10^{-3}$ and $lr_{mlp} = 5.0 \times 10^{-4}$ to optimize the canonical point cloud (*i.e.*, the point cloud before network-based shifting) and the motion guidance function detailed in Sec. 3.2 for all scenes. In addition, during learning motion guidance, to embed core motion information into the network, we initialize a canonical point cloud at $t = 0$ and reset the network parameters in the middle of the process.

Meanwhile, our framework is structured to gradually increase resolution as the optimization process progresses. As shown in Fig. C, We initially obtain motion guidance at quarter resolution of the target image size. Then, in the coarse stage of sketch synthesis, we render strokes into a 50% of the full resolution. We finally get moving sketches by optimizing the full resolution of the target frame size. For instance, for synthetic scenes, we first learn guidance at 100×100 resolution and then optimize the per-stroke transformation using 200×200 frames. The final output produces sketch frames at 400×400 resolution.

	Per-frame Chamfer (\downarrow)	Motion velocity distance ($\times 10^{-3}$) (\downarrow)
4DGS [†]	0.286 ± 0.057	4.24 ± 2.71
Deformable 3DGS [†]	0.269 ± 0.071	4.01 ± 2.67
SC-GS [†]	0.289 ± 0.053	3.99 ± 2.65
Liv3Stroke (Ours)	0.252 ± 0.049	4.16 ± 2.34

Table A. Quantitative results of 3D motion guidance accuracy of [†]GS-based works with filtering based on the opacity value.

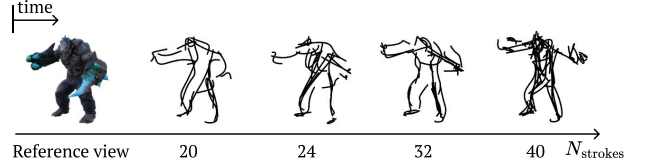


Figure D. The effects of using different numbers of strokes. When reconstructing a sketch video, we allow users to set N_{strokes} . More strokes produce detailed sketches, while fewer strokes yield abstract ones.

C. Additional Results

We provide results of all rendered synthetic scenes in Fig. F. Compared to other existing works, our framework can represent diverse movements and key features of the view-consistent structure directly from RGB video frames. We visualize guidance views at the full target image resolution for better clarity. We highly recommend finding videos in the supplementary material to see the whole movement of each scene.

C.1. Quantitative Results of the Motion Guidance

We present additional quantitative results of the motion guidance that we obtained from Sec. 3.2 and filtered results of GS-based dynamic reconstruction works [15, 44, 47] according to the opacity value with a threshold of $\alpha = 0.5$. Table A and Tab. 1 (b) of the main paper shows our method’s capability to capture meaningful 3D motion information, although it does not pursue realistic reconstruction.

C.2. Results of Different Number of Strokes

We provide results to show the effect of the number of strokes as in Fig. D. Like [5] and [41], we can control abstraction levels of sketches by adjusting the number of curves. With a higher number of strokes, we can capture more detailed features, while fewer strokes result in more abstract representations.

C.3. Limited Multi-View Information

We provide results under limited multi-view information in varied conditions. From a frontal view, we captured frames along a circular trajectory around the object, collecting 100 frames over an angle $\theta(^{\circ})$ while maintaining a constant dis-

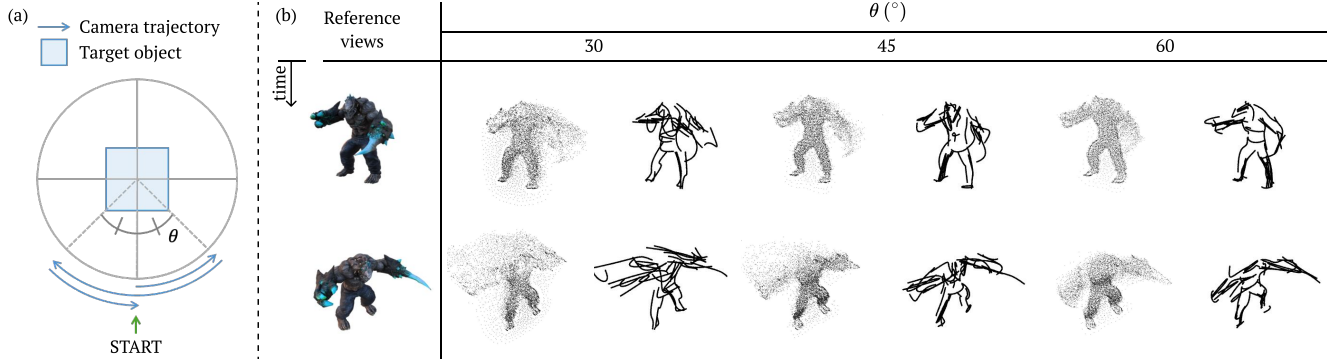


Figure E. Results of limited multi-view information. **(a)** Experimental setup for data collection. We followed a circular path around the object to capture frames, beginning from the frontal view (marked with a green arrow). **(b)** Results of the motion guidance and sketch under different angles. Our approach can achieve 3D motion sketch representation with limited yet sufficient viewpoint information ($\theta \geq 45$), even when the motion guidance exhibits noise, such as at $\theta = 45$.

Method	Novel views		Fixed views	
	Motion	Structure	Motion	Structure
CLIPasso	3.32	3.08	2.87	2.70
Sketch Video Syn.	2.66	2.64	3.93	3.77
Sugg. Contours	4.26	4.29	3.82	3.90
Liv3Stroke (Ours)	3.32	3.08	3.21	2.94

Table B. User study results. Note that “Motion” denotes the evaluation of how well the result describe the desired movement, and “Structure” is the score of how well it contains key features of the 3D structure.

tance from the center. The detailed experimental setup is visualized in Fig. E (a).

Our approach renders 3D sketches of the object in motion with sparse yet adequate viewpoint information, as illustrated in Fig. E (b). Additionally, we find that our method can roughly capture the 3D key structure of the object even when the motion guidance exhibits noise, such as at $\theta = 45$.

C.4. User study

We provide a questionnaire to evaluate the perceptual implication of generated sketches. Participants rated the sketches on a five-point scale (1-5), evaluating them from both novel camera viewpoints and the fixed perspective. The rating criteria were: (1) how effectively the sketch captures the motion and (2) how well it conveys the 3D structure of the target object.

Table B summarizes the answers of 44 participants. Overall, Suggestive Contours [6] achieves the highest scores across all metrics, which can be attributed to its direct contour extraction from 3D meshes, as illustrated in Fig. 6. Unlike other methods that rely on image-based processing, this approach results in higher evaluation scores. For novel views, LiveStroke performs comparably with CLIPasso [41]. While Sketch Video Synthesis [50] has a higher score in

the fixed views, it struggles to effectively capture 3D geometric features and motion characteristics when evaluated from moving camera perspectives. LiveStroke exhibits only minimal performance decrease when transitioning from the novel perspectives to the fixed view, demonstrating consistent performance regardless of the viewing perspective. This stability distinguishes our approach from others, which shows significant performance variations between different viewpoints.



Figure F. Results of synthetic scenes. Our approach can represent diverse motions by using view-consistent deformable 3D strokes.