LayerInbetween: Occlusion-Aware Stroke Correspondence and Inbetweening with Automatic Layering
— Supplemental Material —

HAORAN MO, The Hong Kong University of Science and Technology (Guangzhou), China
ZHONGYUE GUAN, The Hong Kong University of Science and Technology (Guangzhou), China and Tencent, China
YIXIN HU, Tencent America, United States of America
ZEYU WANG*, The Hong Kong University of Science and Technology (Guangzhou), China and The Hong Kong University of Science and Technology, China

1 Results of Reference Stroke Grouping

We show the stroke chains and automatic reference stroke grouping of the testing cases in Fig. 1 and Fig. 2.

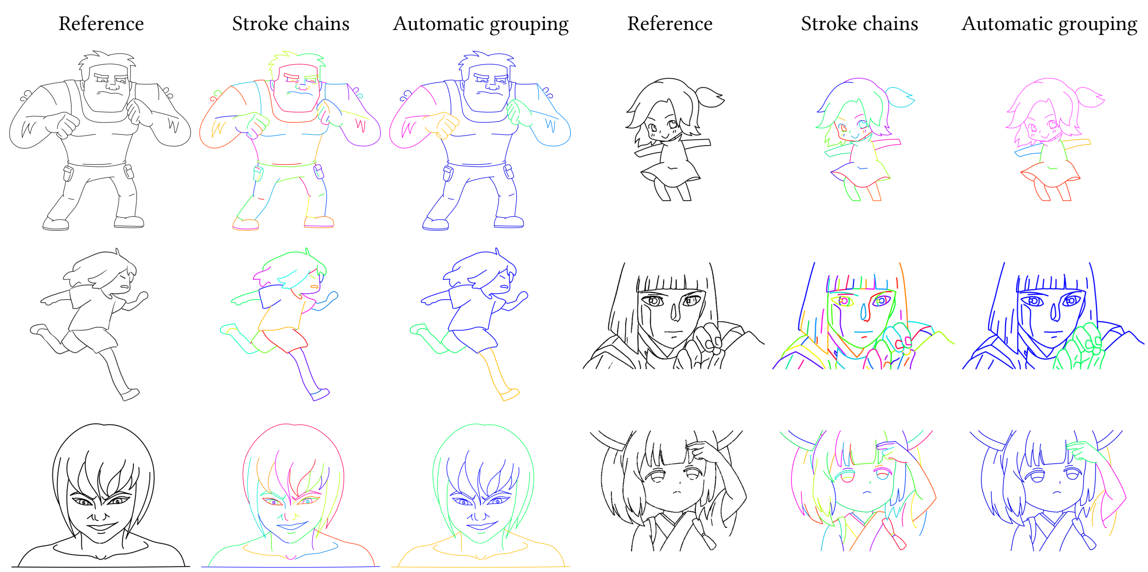

Fig. 1. Visualization of stroke chains and reference stroke grouping.

*Corresponding author

Authors' Contact Information: Haoran Mo, haoranmo@hkust-gz.edu.cn, The Hong Kong University of Science and Technology (Guangzhou), Guangzhou, China; Zhongyue Guan, zguanad@connect.ust.hk, The Hong Kong University of Science and Technology (Guangzhou), Guangzhou, China and Tencent, Shenzhen, China; Yixin Hu, yixinhu.yh@gmail.com, Tencent America, New York, United States of America; Zeyu Wang, zeyuwang@ust.hk, The Hong Kong University of Science and Technology (Guangzhou), Guangzhou, China and The Hong Kong University of Science and Technology, Hong Kong, China.

Fig. 2. Visualization of stroke chains and reference stroke grouping.

2 Visualization of Bidirectional Prediction Scheme

The bidirectional prediction scheme handles disappearing and appearing strokes resulting from occlusion in a unified manner. We show examples with the bidirectional prediction in Figures 3, 4, 5, and 6. The crossing-leg case in Fig. 3 is a little convincing during reverse prediction due to the sparse contexts on top of the legs, leading to a non-precise global layer transformation for the top curves of the right leg. But in the other cases with dense contexts, the quality of disoccluded strokes is much better.

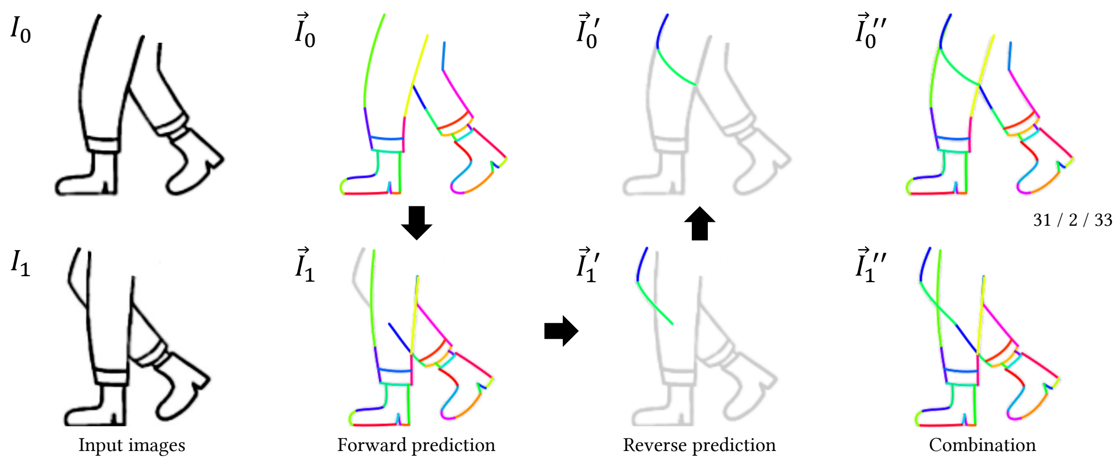

Fig. 3. Bidirectional prediction for stroke correspondence. Numbers on the right of each case denote forward stroke number / backward stroke number / total stroke number.

..

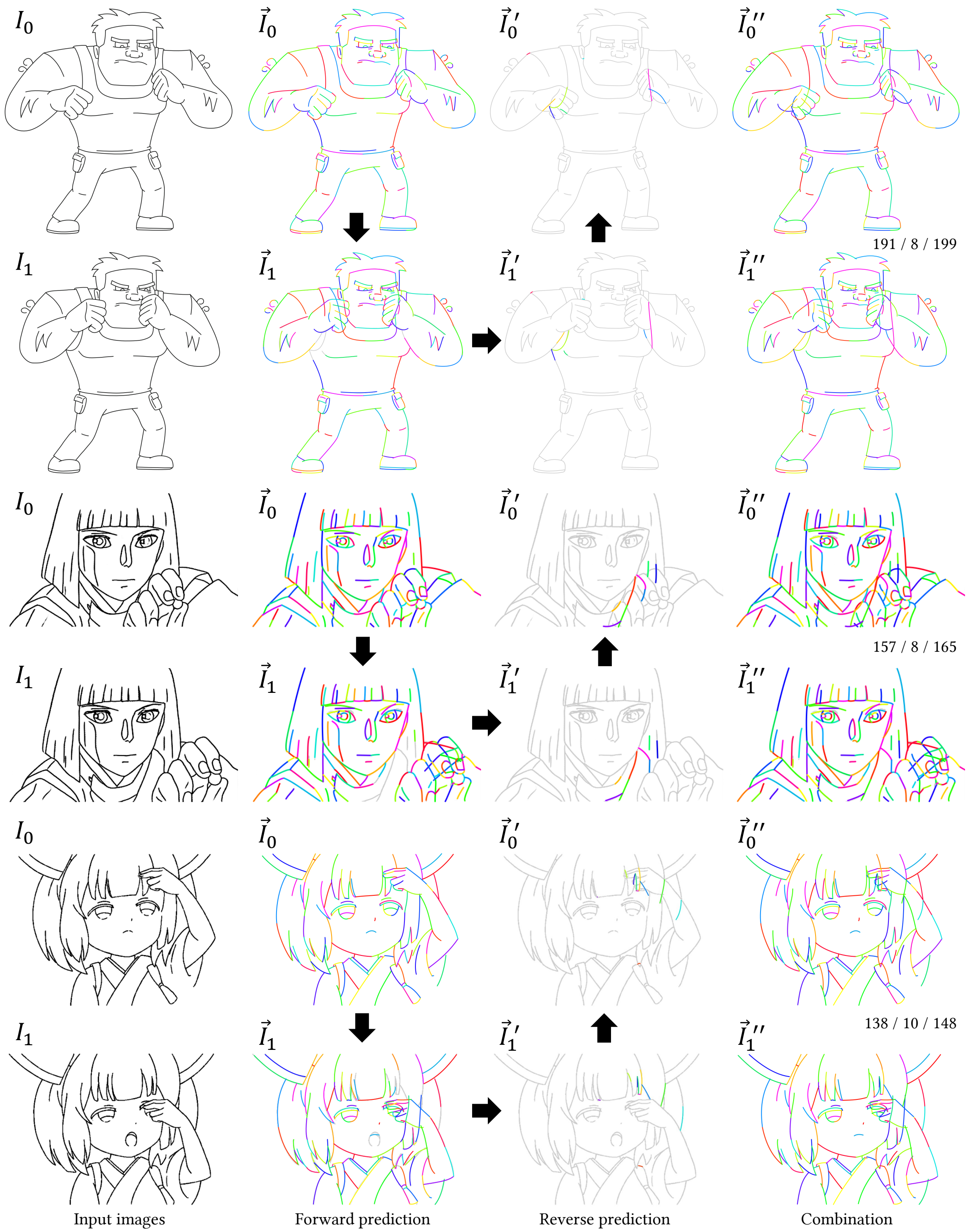

Fig. 4. Bidirectional prediction for stroke correspondence. Numbers on the right of each case denote forward stroke number / backward stroke number / total stroke number.

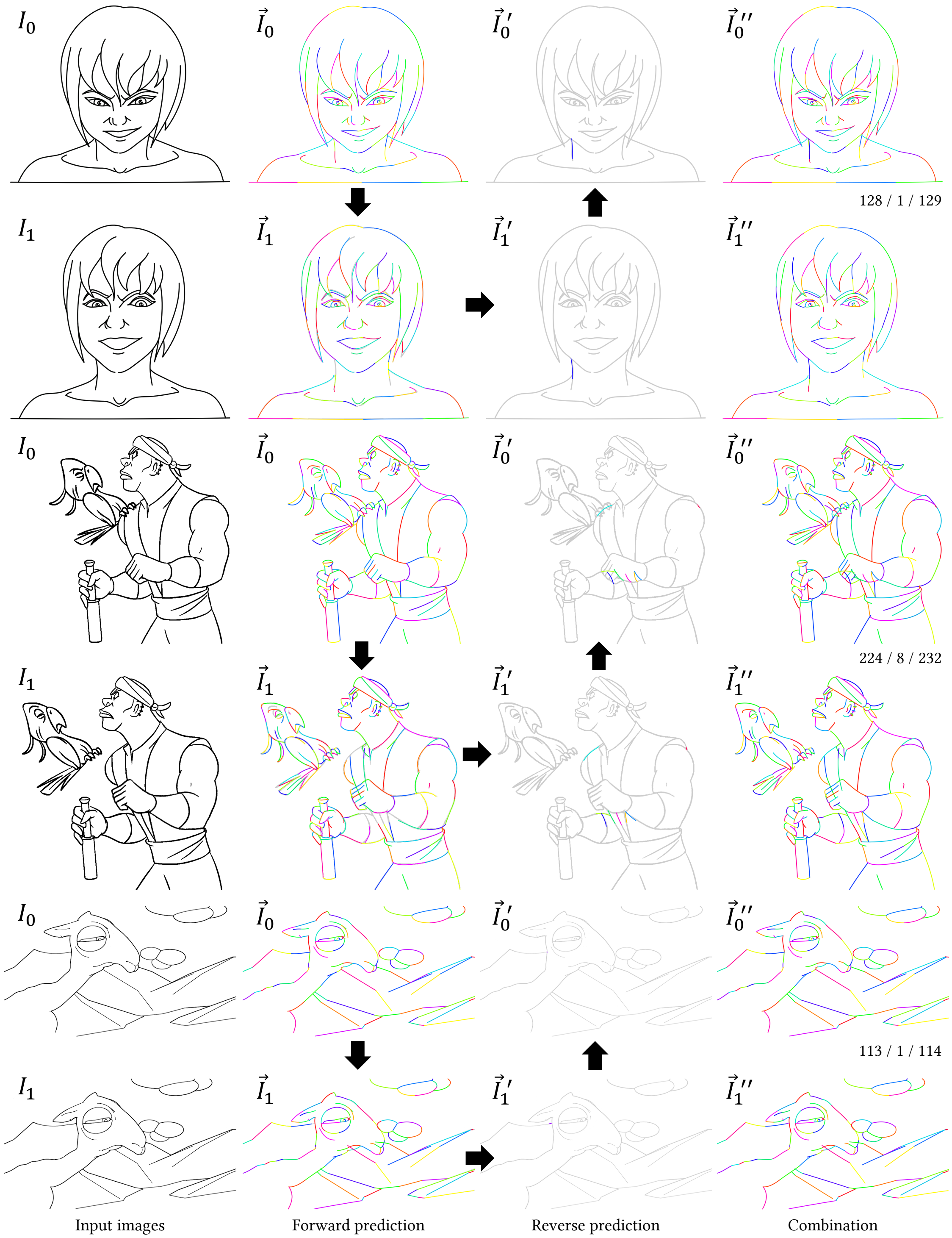

Fig. 5. Bidirectional prediction for stroke correspondence. Numbers on the right of each case denote forward stroke number / backward stroke number / total stroke number.

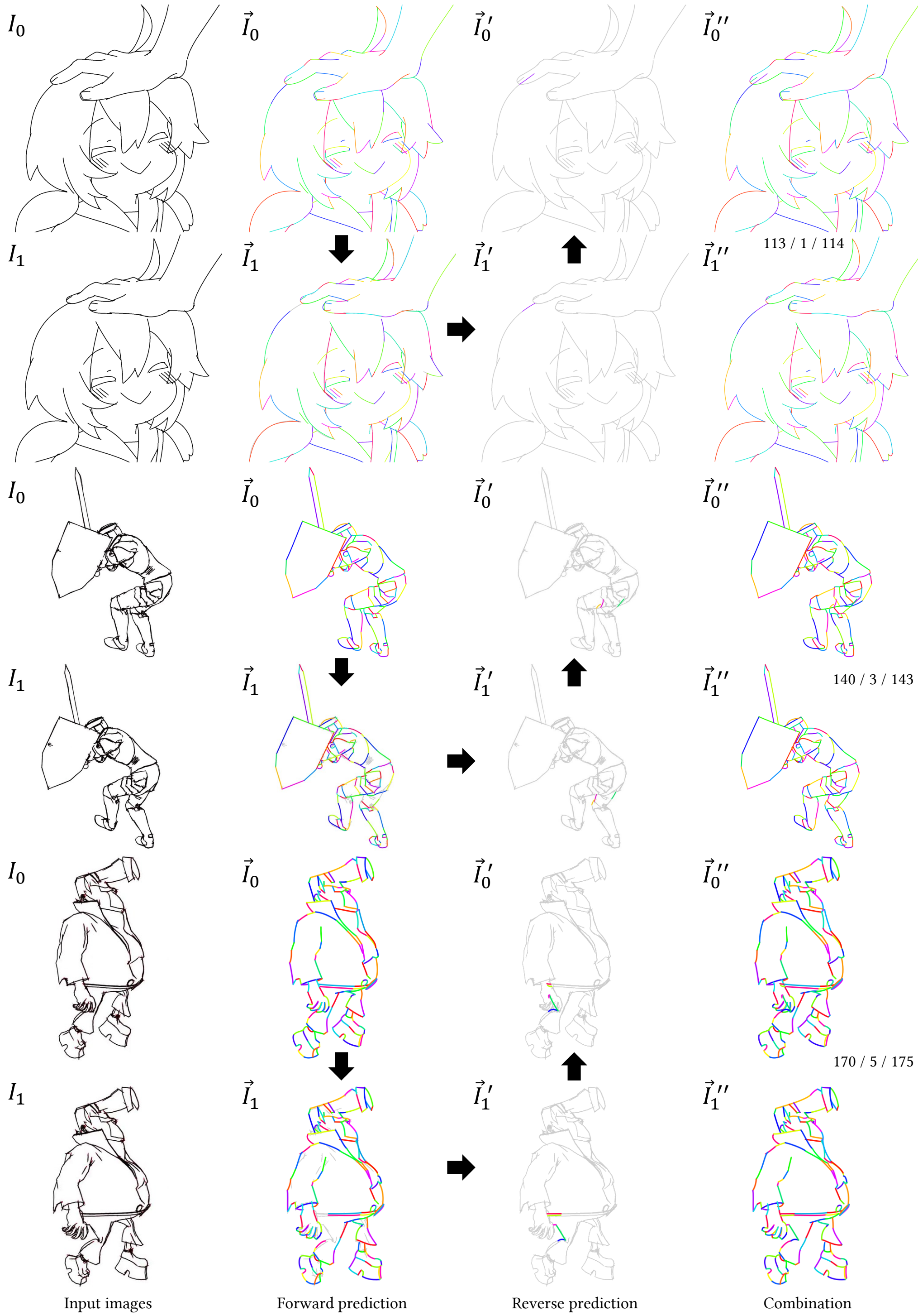

Fig. 6. Bidirectional prediction for stroke correspondence. Numbers on the right of each case denote forward stroke number / backward stroke number / total stroke number.

3 Results of Occluded Stroke Fixing

Fig. 7 shows the fixed strokes with the occluded stroke fixing strategy. Note that this strategy is applied to all the strokes during reverse prediction. Refer to Figures 3, 4, 5, and 6 for the visualization of these strokes.

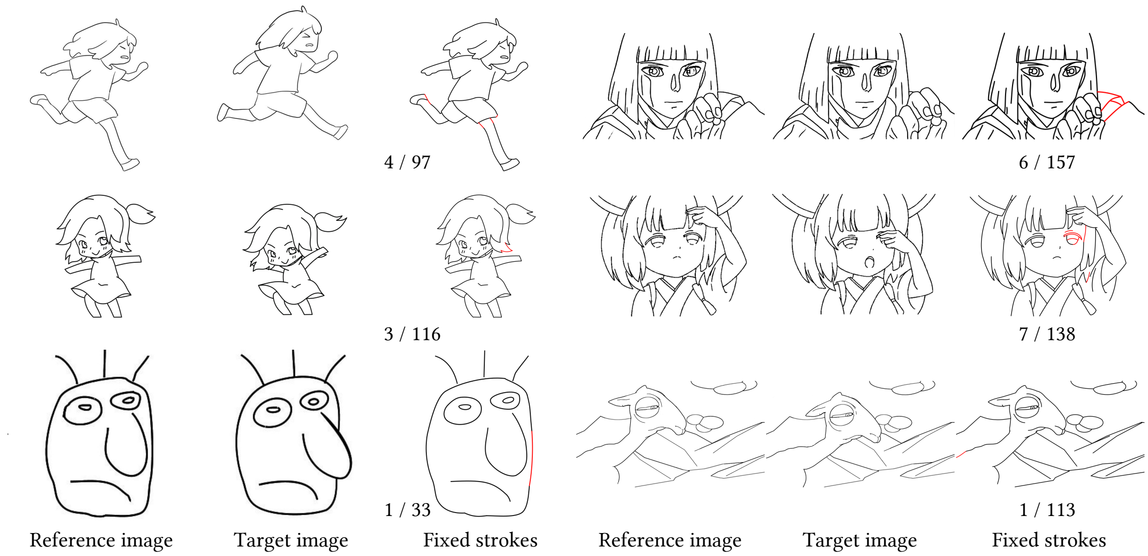

Fig. 7. Visualization of fixed strokes (in red) with occluded stroke fixing strategy. Examples without detected strokes for fixing are not shown here. The numbers denote fixed stroke number / total stroke number.

4 Runtime Comparisons

Table 1 shows runtimes of the other inbetweening methods. Table 2 shows breakdown timings of each sub-process.

Table 1. Runtime comparisons.

	Fig. 11a	Fig. 12a	Fig. 12c	Fig. 16b	Fig. 12b	Fig. 16a	Fig. 11b	Fig. 10c	Fig. 12d	Fig. 16c	Fig. 1
JoSTC [Mo et al. 2024]	1.438	1.523	5.079	4.706	4.312	5.261	5.083	6.252	5.723	9.822	9.482
RIFE [Huang et al. 2022]	0.23	0.202	0.161	0.402	0.289	0.221	0.329	0.457	0.31	0.31	0.687
AnimeInbet [Siyao et al. 2023]	0.502	0.343	2.033	1.539	0.726	1.328	0.505	0.582	1.506	2.253	1.954
TPS-Inbetween [Zhu et al. 2025]	1.046	0.652	0.808	0.665	0.682	1.018	0.678	0.743	0.761	0.659	0.949
Ours	6.027	10.153	11.832	13.858	14.023	17.850	22.031	23.920	26.389	33.185	44.944

Table 2. Breakdown timings. "/" means that the sub-process is not needed.

	Fig. 11a	Fig. 12a	Fig. 12c	Fig. 16b	Fig. 12b	Fig. 16a	Fig. 11b	Fig. 10c	Fig. 12d	Fig. 16c	Fig. 1
Forward Prediction											
Data pre-processing	0.433	0.320	1.253	0.926	1.365	1.949	2.270	3.137	3.096	4.988	5.597
Reference stroke grouping	0.142	0.746	0.176	0.233	0.812	0.209	0.265	1.451	1.007	0.255	0.305
Layer construction[1]	0.598	0.894	0.489	0.689	0.858	0.786	1.360	1.303	1.113	1.032	1.910
Global layer transformation	0.158	0.174	0.151	0.159	0.204	0.185	0.203	0.203	0.212	0.189	0.249
Local layer transformation	0.971	0.917	2.994	2.631	3.442	4.903	5.487	7.990	6.243	9.974	12.148
Endpoint matching	0.373	0.378	0.958	0.857	1.095	1.463	1.702	2.208	1.928	3.219	5.020
Control point matching	0.429	0.412	1.008	1.019	1.231	1.533	1.855	2.417	1.892	3.130	4.504
Backward Prediction											
Data pre-processing	/	/	0.658	0.338	/	0.890	/	1.306	1.721	2.586	2.964
Reference stroke grouping	/	/	0.259	0.230	/	0.312	/	1.535	1.044	0.301	0.334
Global layer transformation	/	/	0.154	0.157	/	0.185	/	0.211	0.187	0.222	0.220
Local layer transformation	/	/	0.254	0.196	/	0.228	/	0.227	0.517	0.538	0.614
Endpoint matching	/	/	0.181	0.164	/	0.182	/	0.280	0.313	0.320	0.362
Control point matching	/	/	0.199	0.185	/	0.224	/	0.362	0.303	0.295	0.407
Stroke interpolation	0.540	0.550	0.810	0.780	0.800	0.850	1.300	1.290	1.090	1.460	1.560
Occlusion resolving[2]	2.383	5.762	2.288	5.294	4.216	3.951	7.589	/[3]	5.723	4.676	8.750
Total	6.027	10.153	11.832	13.858	14.023	17.850	22.031	23.920	26.389	33.185	44.944

[1] Including timings for depth estimation and target layer mask generation with SAM2 tracking.
[2] Including timings for rendering layer-drawing sequences, layer/occlusion mask generation with SAM2 tracking, and hiding occluded strokes.
[3] Occlusion resolving was not applied to this example due to the insignificant occlusion.

References

Zhewei Huang, Tianyuan Zhang, Wen Heng, Boxin Shi, and Shuchang Zhou. 2022. Real-Time Intermediate Flow Estimation for Video Frame Interpolation. In *European Conference on Computer Vision*. 624–642.

Haoran Mo, Chengying Gao, and Ruomei Wang. 2024. Joint Stroke Tracing and Correspondence for 2D Animation. *ACM Transactions on Graphics (TOG)* 43, 3 (2024), 1–17.

Li Siyao, Tianpei Gu, Weiye Xiao, Henghui Ding, Ziwei Liu, and Chen Change Loy. 2023. Deep Geometrized Cartoon Line Inbetweening. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 7291–7300.

Tianyi Zhu, Wei Shang, and Dongwei Ren. 2025. Thin-Plate Spline-Based Interpolation for Animation Line Inbetweening. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 39. 10995–11003.