Supplemental Materials on

Guided Interactive Video Object Segmentation Using Reliability-Based Attention Maps

Yuk Heo, Yeong Jun Koh, and Chang-Su Kim

S-1. User Study

We conducted a user study with 12 participants to assess the proposed GIS algorithm in real applications, as described in Section 4.3 in the main paper. Table S-1 lists detailed results for each user: seconds per video (SPV), rounds per video (RPV), and J&F scores. Note that the average performances over all 12 users are reported in Table 4 in the main paper. Proposed-RS1 enabled most users to finish the interactive segmentation most quickly, requiring the fewest SPV. This is because it does not need the inspection time to select a frame to be annotated. Also, in terms of the three metrics, the proposed algorithm, regardless of its method (w/o RS, RS1, or RS4), is preferred to the conventional algorithm [7] by all users, except for only a few exceptions (*e.g.* User 7 required 1.52 RPV using [7], while 1.57 RPV using Proposed-RS1). Figure S-1 illustrates how a user performed the interactive VOS using Proposed-RS4.

Table S-1: Detailed user study results. The best results are boldfaced.

		User 1	User 2	User 3	User 4	User 5	User 6	User 7	User 8	User 9	User 10	User 11	User 12
SPV	Heo *et al.* [7]	52.8	70.9	85.3	46.4	54.9	116.5	39.8	52.7	79.7	36.8	57.6	108.6
	Proposed w/o RS	33.2	38.9	47.7	33.6	37.6	70.8	27.4	38.1	52.0	25.1	57.6	89.9
	Proposed-RS1	**23.4**	**35.5**	**27.4**	**29.1**	**23.2**	**48.8**	**24.5**	**30.6**	38.1	**21.1**	**42.2**	71.2
	Proposed-RS4	24.8	41.0	33.0	29.9	30.0	55.7	27.2	33.7	**31.0**	25.4	49.2	**64.6**
RPV	Heo *et al.* [7]	2.13	2.90	3.43	2.13	1.90	3.30	1.52	2.07	2.80	1.47	2.23	3.10
	Proposed w/o RS	1.60	**2.00**	2.03	**1.40**	1.57	2.27	**1.33**	1.57	2.30	**1.33**	2.23	3.07
	Proposed-RS1	1.43	2.17	**1.63**	1.80	**1.53**	**1.97**	1.57	1.50	1.97	1.40	**1.87**	3.00
	Proposed-RS4	**1.33**	2.03	1.83	1.53	1.67	**1.97**	1.47	**1.43**	**1.53**	1.37	1.90	**2.30**
J&F	Heo *et al.* [7]	0.753	0.775	0.776	0.750	0.756	0.799	0.754	0.768	0.782	0.698	**0.813**	0.808
	Proposed w/o RS	0.785	0.801	**0.803**	**0.792**	0.773	0.814	0.781	0.801	0.815	0.725	**0.813**	0.830
	Proposed-RS1	0.771	**0.805**	0.777	0.782	0.759	0.807	0.779	**0.804**	**0.818**	0.725	0.810	0.825
	Proposed-RS4	**0.786**	0.801	0.790	0.790	**0.793**	**0.818**	**0.782**	0.789	0.789	**0.749**	**0.813**	**0.832**

S-2. Analysis on R-scores

We analyze the correlation between R-scores and J&F scores. In Figure S-2, we compare the average R-scores and J&F scores over frames for each sequence in DAVIS2017 [26] according to the interaction rounds. To clarify the relationship between the average R-score and the average J&F score, we omit sequences that have higher J&F scores than 0.9 in the first round, because those sequences do not require guidance using R-scores. We see that R-scores and J&F scores are highly correlated and both of them tend to increase as the round progresses.

S-3. Implementation Details

Intersection-aware propagation: As mentioned in Section 3.2, Y_n is used to obtain the overlapped object feature. Specifically, Y_n is downsampled using the nearest-neighbor interpolation. Also, two convolution layers in the IAP module in Figure 5 have 256 filters of size 5×5.

Segmentation head: We adopt the decoder architecture in [7] as the segmentation head in Figure 2. $\mathbf{F}_t$, $\mathbf{G}_t$, and $\mathbf{H}_t$ are concatenated and then fed into a 1×1 convolution, which yields a feature of 512 channels. Using the output feature, the head generates segmentation results.

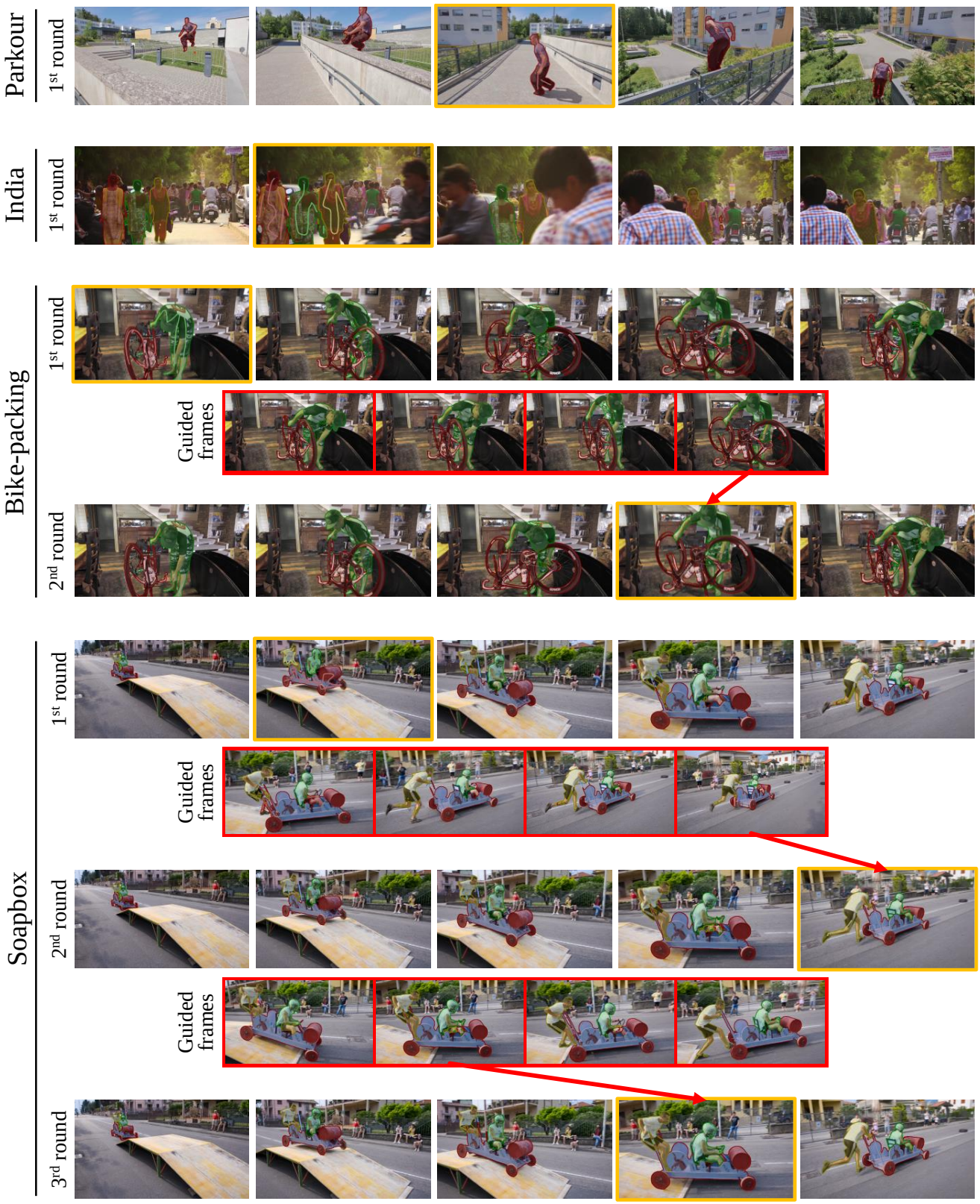

Figure S-1: Summary of how a user performed interactive VOS using Proposed-RS4. Yellow boxes are annotated frames, and red boxes are four guided frames provided by Proposed-RS4 in each round.

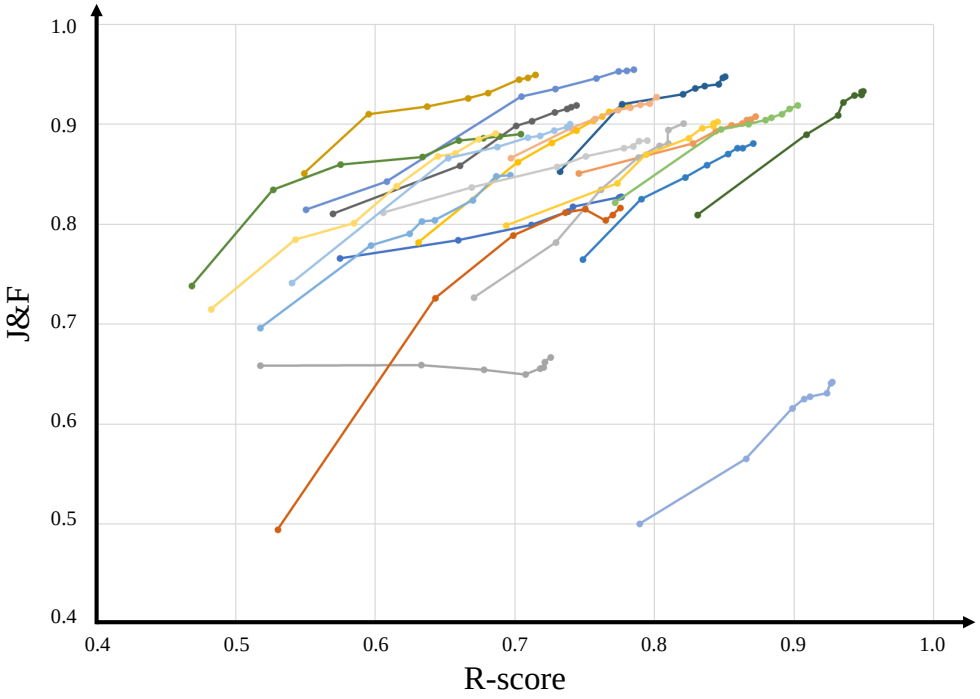

Figure S-2: Correlation between R-scores and J&F scores. Each video is represented in a different color.

Training: We employ the Adam optimizer to minimize the cross-entropy loss between segmentation prediction and the ground-truth. Starting with a learning rate of 1.0×10^{-5}, we decrease it by a factor of $\frac{1}{5}$ in every 20 epochs. The training is iterated for 60 epochs with 6 mini-sequence batches. It takes three days for the convergence.

S-4. Alternatives to Intersection-Aware Propagation

In Table 3, we replace the intersection-aware propagation (IAP) module with two existing propagation methods, local distance map (LDM) [20] and local transfer module (LTM) [7]. Instead of using the proposed overlapped object feature $\mathbf{H}_t$, we perform the replacement as follows. First, as in LDM, we compute a distance between two feature vectors $\mathbf{F}_t(p)$ and $\mathbf{F}_n(q)$ for pixels p and q, which is given by

$$d(p,q) = 1 - \frac{2}{1 + \exp\left(\|\phi_D(\mathbf{F}_t(p)) - \phi_D(\mathbf{F}_n(q))\|^2\right)} \tag{1}$$

where ϕ_D is a feature transform for LDM. Then, the local distance map L_t for target frame I_t is defined as

$$L_t(p) = \begin{cases} \min_{q \in \mathcal{O}_n \cap \mathcal{N}(p)} \; d(p,q) & \text{if } \mathcal{O}_n \cap \mathcal{N}(p) \neq \emptyset, \\ 1 & \text{otherwise,} \end{cases} \tag{2}$$

where $\mathcal{O}_n$ is the union set of pixels in the segmented object regions in the neighbor frame and $\mathcal{N}(p)$ is a 13×13 neighbor set of pixel p. Second, to implement LTM, we estimate a local affinity $\mathbf{W}^{\mathrm{L}}$ between two feature vectors $\mathbf{F}_t(p)$ and $\mathbf{F}_n(q)$,

$$\mathbf{W}^{\mathrm{L}}(p,q) = \begin{cases} \phi_T(\mathbf{F}_t(p)) \cdot \phi_T(\mathbf{F}_n(q))^T & q \in \mathcal{N}(p), \\ 0 & \text{otherwise,} \end{cases} \tag{3}$$

where ϕ_T is a feature transform for LTM. Then $\mathbf{W}^{\mathrm{L}}$ is normalized column-by-column to the yield the transition matrix $\mathbf{A}^{\mathrm{L}}$. By multiplying $\mathbf{A}^{\mathrm{L}}$ and the downsampled probability $\hat{Y}_n^{\mathrm{L}}$ from the neighboring probability map $\hat{Y}_n$, we obtain the output of LTM. Eventually, LDM and LTM replace IAP by employing $L_t \in \mathbb{R}^{HW \times 1}$ and $\mathbf{A}^{\mathrm{L}}\hat{Y}_n^{\mathrm{L}} \in \mathbb{R}^{HW \times 1}$, instead of $\mathbf{H}_t \in \mathbb{R}^{HW \times C_3}$, respectively. Note that ϕ_D and ϕ_T output 128-channel features by employing 1×1 convolution.