

Supplemental Material

Zhimeng Huang¹[0000-0001-8026-9349], Chuanmin Jia²[0000-0002-7418-6245],
Shanshe Wang¹[0000-0002-7665-7434], and Siwei Ma¹[0000-0002-2731-5403]

¹ National Engineering Research Center of Visual Technology, Peking University,
Beijing 100871, China

² Wangxuan Institute of Computer Technology, Peking University, Beijing 100871,
China

1 Supplemental Experimental Configurations

In this section, the source codes, instructions and configurations of codecs and VOS methods are provided.

1.1 Codecs

HEVC. We choose the x265 library deployed in FFMPEG software³ as the implementation of HEVC standard. The command line for generating x265 is provided as follows,

```
ffmpeg -f rawvideo -video_size WxH -i input.mkv -c:v libx265 -preset veryfast  
-tune zerolatency -x265-params "crf=Q:keyint=GOP:verbose=1" output.mkv -y
```

Among them, W, H, Q, GOP represents the width, height, QP and GOP size, respectively. Specifically, GOP is 8 for all of the dataset.

VVC. Considering the efficiency and effectiveness, we choose VVenC⁴ as the implementation of VVC standard. The command line for encoding is provided as follows,

```
vvencFFapp -c cfg/experimental/lowdelay-faster.cfg -InputFile input.yuv -s  
WxH -fr 25 -QP Q -BitstreamFile B
```

And the command line for decoding is provided as follows:

```
vdecapp -b B -o output.yuv
```

Among them, W, H, Q, B represent the width, height, QP and bitstream of the to-be-coded videos. Note that the configuration of GOP size is set in the cfg file.

1.2 VOS Models

AOT As all of the system is conducted by PyTorch, the chosen version⁵ of AOT is also implemented by PyTorch. Specifically, the model is **R50-AOTL**

³ <https://github.com/FFmpeg/FFmpeg>

⁴ <https://github.com/fraunhoferhhi/vvenc>

⁵ <https://github.com/yoxu515/aot-benchmark>

045 **STCN** We use the official released version⁶ of STCN as the implementation. 045

046

047

048

1.3 VOS Evaluation

049 **DAVIS** For DAVIS dataset, we use the valid dataset to verify the performance 049
050 of the proposed framework. Thus we utilize the official released tools⁷ to evaluate 050
051 the performance. 051

052

053

054 **YouTube-VOS** As the source code of the evaluation of YouTube-VOS dataset 054
055 is not provided, we evaluate the performance of YouTube-VOS on the competi- 055
056 tion server⁸. 056

057

058

2 Complexity Analysis

059

060

2.1 Experimental results about complexity

061

062

063 We conduct an additional experiment to compare the runtime complexity be- 063
064 tween the baseline and our proposed framework. As the traditional codecs are 064
065 not neural network based methods, it is difficult to calculate the flops to evalu- 065
066 ate the complexity. As an alternative, we further provide the runtime analysis to 066
067 measure the complexity of the whole model. The experiment is employed on the 067
068 DAVIS 2017 dataset. The simulation environment is based on Ubuntu 18.04 with 068
069 one NVIDIA 3080ti graphic card. The experimental results are shown in Table 1, 069
070 in which t_c ; t_v ; t_s represent the runtime of the codecs, VOS model, and the whole 070
071 framework, respectively. Table 1 presents that our framework still outperforms 071
072 x265 when the runtime is similar. Note that we choose a faster VOS model to 072
073 align the runtime. The experimental results also show that the state-of-art codec 073
074 VVEnc (an implementation of the Volatile Video Coding (VVC) standard.) is 074
075 much slower. 075

076

077

Table 1. Runtime Analysis on DAVIS 2017 Dataset

078

079

Codec	VOS	t_c	t_v	$t_s = t_c + t_s$	Bitrate#	$(J \& F)_m$ "
x265	STCN	499.5s	74.5s	574.0s	0.0209	0.6947
x265	AOT	499.5s	116.0s	615.5s	0.0209	0.7386
Ours	STCN	535.2s	74.4s	609.6s	0.0158	0.8265
Ours	AOT	531.7s	116.0s	647.7s	0.0178	0.8354
VVEnc	STCN	2475.9s	74.2s	2550.1s	0.0179	0.7440

085

086

087

⁶ <https://github.com/hkchengrex/STCN>

088

⁷ <https://github.com/davisvideochallenge/davis2017-evaluation>

089

⁸ <https://competitions.codalab.org/competitions/20127#results>

045

046

047

048

049

050

051

052

053

054

055

056

057

058

059

060

061

062

063

064

065

066

067

068

069

070

071

072

073

074

075

076

077

078

079

080

081

082

083

084

085

086

087

088

089

2.2 Experimental results about decoding complexity

We conduct an additional experimental on DAVIS 2017 dataset to compare the runtime complexity of the encoder and the decoder. The experimental results in shown in Table 2, in which t_e , t_d , t denote the runtime on the encoder side, decoder side, and the whole framework, respectively. Table 2 presents that our proposed method is much faster than traditional codecs because the VOS model is employed on the encoder side.

Table 2. Encoder/Decoder Complexity Analysis on DAVIS 2017 Dataset

Codec	VOS	t_e	t_d	$t = t_e + t_d$	Bitrate#	$(J \& F)_m$
x265	STCN	489.7s	84.3s	574.0s	0.0209	0.6947
Ours	STCN	575.9s	33.7s	609.6s	0.0158	0.8265
x265	AOT	489.7s	125.8s	615.5s	0.0209	0.7386
Ours	AOT	614.2s	33.5s	647.7s	0.0178	0.8354

3 Supplemental Experimental Results

3.1 DAVIS 2017

In the submitted paper, we provide the J_m and F_m of the proposed framework on DAVIS 2017. In this subsection the $\{\text{Recall}\uparrow, \text{Decay}\downarrow\} \times \{\mathcal{J}, \mathcal{F}\}$ for DAVIS 2017 are provided.

Table 3. Supplemental Experimental Results on DAVIS 2017 dataset.

VOS Model	Method	Bitrate#	J_r	J_d	F_r	F_d
AOT	Original	-	0.9129	0.0428	0.9430	0.0617
	x265(baseline)	0.0209	0.7987	0.0926	0.8168	0.1483
	x265+Ours	0.0178	0.9051	0.0398	0.9420	0.0566
STCN	Original	-	0.9142	0.0603	0.9458	0.0861
	x265(baseline)	0.0209	0.7923	0.1470	0.8204	0.2123
	x265+Ours	0.0158	0.8820	0.0427	0.9201	0.0727

3.2 More Samples

In this subsection, we provide more visualizations about the experimental results in Fig. 1 to Fig. 3. The first column indicates the original video sequences and the ground truth annotation. The second represents the original video sequences and masks extracted by AOT model. The third column denotes the video sequence compressed by x265 and the masks extracted by AOT. And the forth column is the video sequence compressed by x265 and then enhanced by our work.

135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179

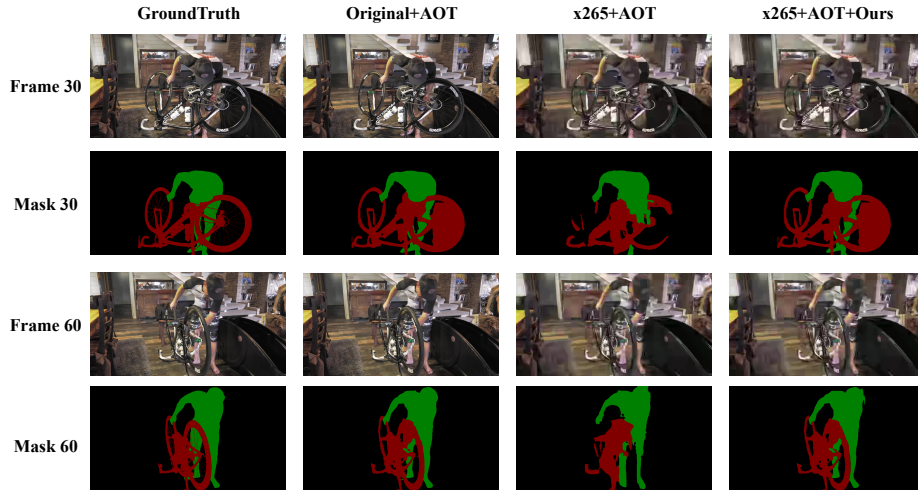


Fig. 1. Visualization of the sequence *bike-packing* in DAVIS dataset.

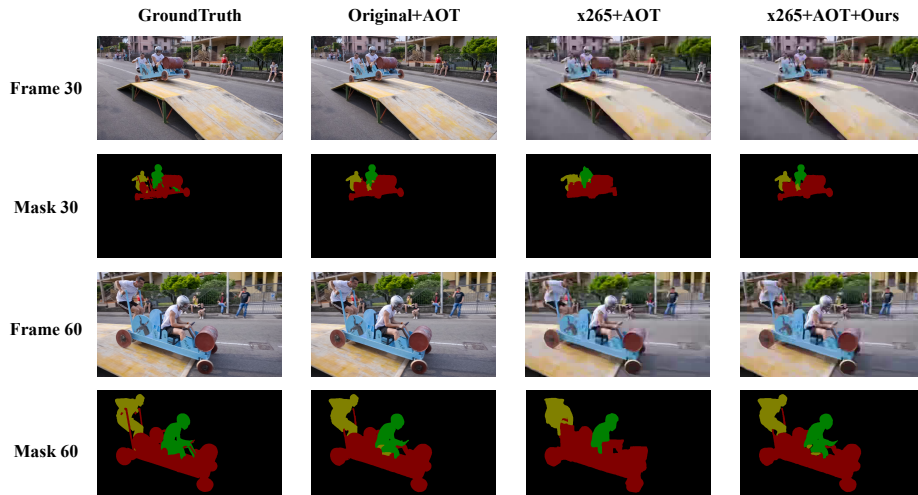


Fig. 2. Visualization of the sequence *soapbox* in DAVIS dataset.

135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179

180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224

180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224

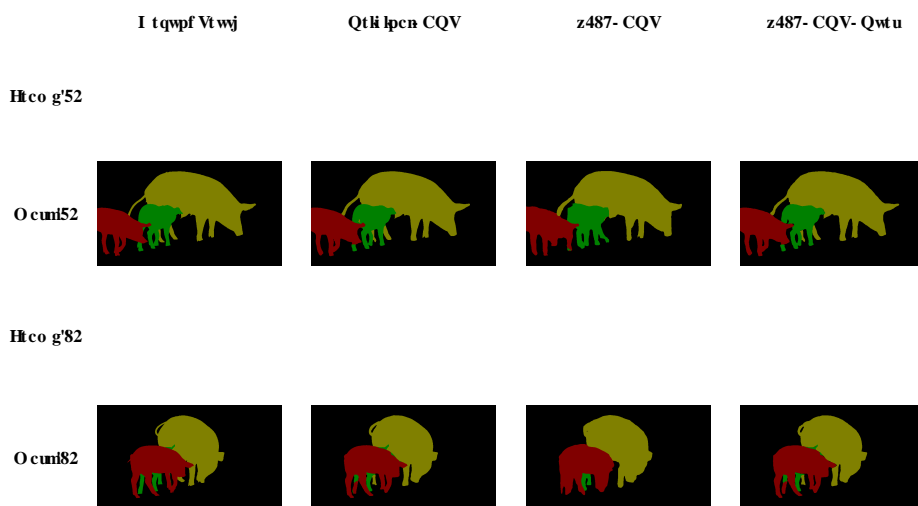


Fig. 3. Visualization of the sequence *pig* in DAVIS dataset.