Supplementary Material for CVPR paper

Paper ID 5030

March 13, 2023

1 Details about Query

1.1 What is Query?

The Query operation in this article is implementated by $torch.nn.functional.grid\_sample$ in Pytorch, which is defined as: Given an $\boldsymbol{input}$ and a $\boldsymbol{flow-fieldgrid}$, computes the $\boldsymbol{output}$ using input values and pixel locations from grid.

See more in `https://pytorch.org/docs/stable/generated/torch.nn.functional.grid_sample.html?highlight=grid_sample#torch.nn.functional.grid_sample`

1.2 Query Neighborhood

Following the previous work RAFT, when querying forward optical flow features, we also query the neighborhood points within a 7×7 range around the query point to provide a global view and optimization direction. Similarly, when querying the OE features of scale $\boldsymbol{s}$, we also query two neighborhood scales $(\boldsymbol{s} - l, \boldsymbol{s} + l)$ to provide optimization directions, In our experiment, l was set to 0.125, and a small increase or decrease of l did not significantly affect the performance.

1.3 Proportional conversion in OE Querying

The shape of multiscale correlation optical flow feature C_f is $H \times W \times (S+1)$, assuming $S = 4$, the scales corresponding to the third dimension of C_f are $\{0.5, 0.75, 1, 1.33, 2\}$. if we want to sample from C_f based on the current OE estimation result $\boldsymbol{s}$, we need first to convert $\boldsymbol{s}$ into coordinates between $[0, 4]$. We give the conversion relationship between $\boldsymbol{s}$ and sampling coordinates $\boldsymbol{p}$ as follows:

$$\boldsymbol{p} = \begin{cases} 4(\boldsymbol{s} - 0.5) & \boldsymbol{s} <= 1 \\ 4(1 - \dfrac{1}{\boldsymbol{s}}) + 2 & \boldsymbol{s} > 1 \end{cases}$$

2 Network structure details of RAFT+TPCV

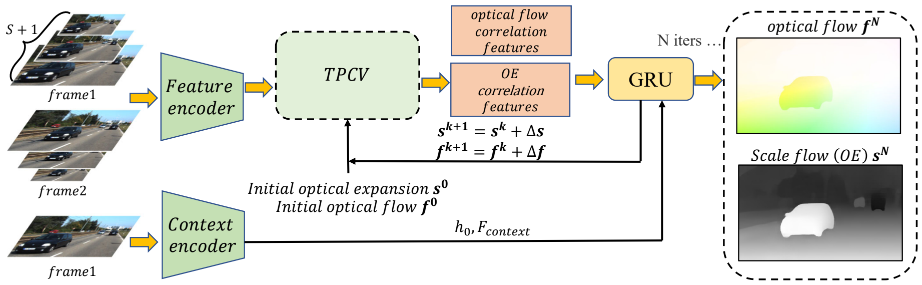


Figure 1: **Network Structure of RAFT+TPCV.** TPCV queries the correlation feature according to the current optical flow and OE results, and then sends it to the GRU module to estimate the updated residual amount. The final result is output after N iterations. Compared with RAFT, we only replace the original 4D correlation volume, and add an output header to output the OE update residuals. The upgrade of the method hardly brings any additional computation.

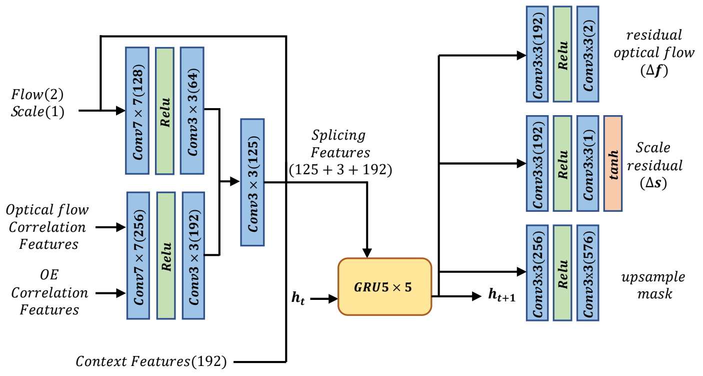


Figure 2: **Detail of GRU Optimizer.**

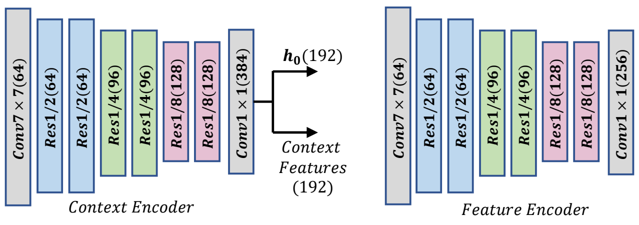


Figure 3: **Feature encoder of TPCV+RAFT.**

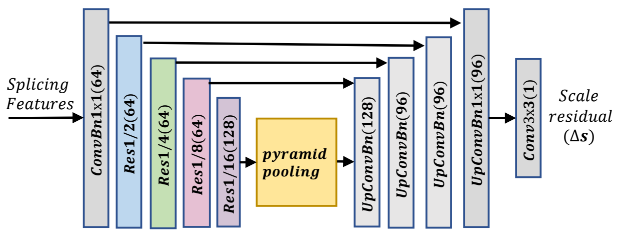
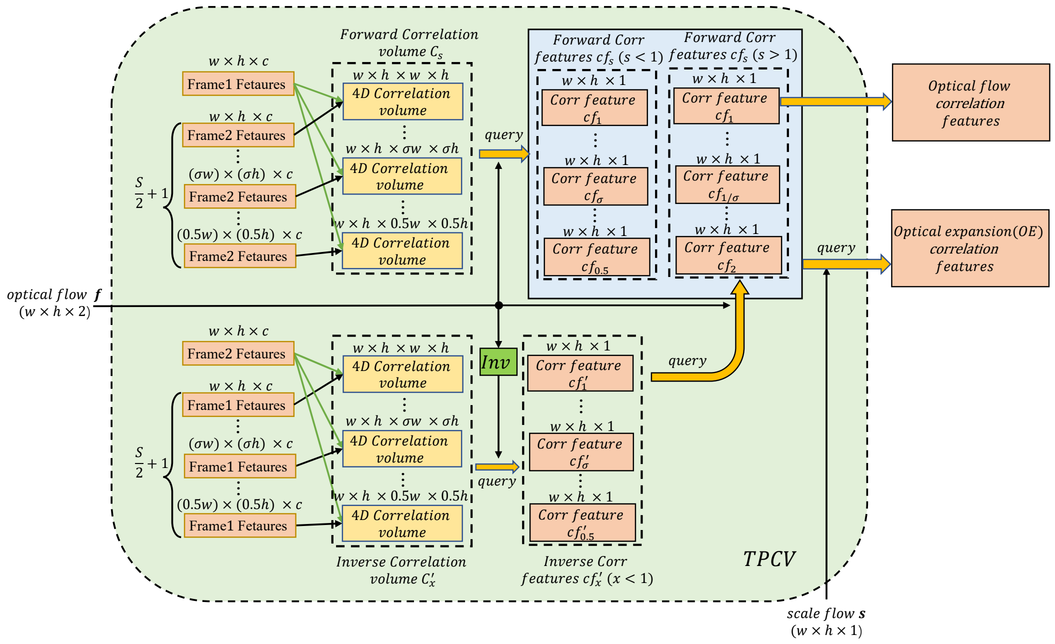


Figure 4: **Additional updates.** To further refine the OE results, we use a small Unet network instead of GRU to perform an additional refinement of the OE results.



Forward Correlation volume C_s
$w \times h \times c$
Frame1 Fetaures
Frame2 Fetaures
$(\sigma w) \times (\sigma h) \times c$
$(0.5w) \times (0.5h) \times c$
$\frac{S}{2}+1$
$w \times h \times w \times h$
4D Correlation volume
$w \times h \times \sigma w \times \sigma h$
$w \times h \times 0.5w \times 0.5h$
query
Forward Corr features cf_s ($s<1$)
Forward Corr features cf_s ($s>1$)
$w \times h \times 1$
Corr feature cf_1
Corr feature cf_σ
Corr feature $cf_{0.5}$
Corr feature $cf_{1/\sigma}$
Corr feature cf_2
Optical flow correlation features
Optical expansion(OE) correlation features
optical flow f ($w \times h \times 2$)
Inv
Corr feature cf'_1
Corr feature cf'_σ
Corr feature $cf'_{0.5}$
Inverse Correlation volume C'_x
Inverse Corr features cf'_x ($x<1$)
TPCV
scale flow s ($w \times h \times 1$)

Figure 5: **TPCV process.** This figure shows the complete TPCV process: establish the forward and inverse correlation volume C_s and $C_x^{'}$; query the correlation features $cf_s(s<1)$ and $cf_x^{'}(x<1)$ from the C_s and $C_x^{'}$; The inverse $cf_x^{'}$ is transferred to the forward correlation feature $cf_s(s>1)$; the scale correlation feature is queried from cf_s, and cf_1 is exported as the optical flow correlation feature.

Method	Input	2D Metrics		3D Metrics		
		$\delta_{2D} < 1px(\uparrow)$	$EPE(\downarrow)$	$\delta_{3D} < 0.05m(\uparrow)$	$\delta_{3D} < 0.10m(\uparrow)$	$EPE(\downarrow)$
RAFT3D	RGB-D	86.4%	2.46	87.8%	91.5%	0.062
RAFT3D+TPCV (ours)	RGB-D	**87.8**%	**1.95**	**88.1**%	91.8%	0.048

Figure 6: **Test results on Flyingthings3D.** We followed the same test process as RAFT3D, and all indicators have been greatly improved after the application of TPCV.