CRAFT: Cross-Attentional Flow Transformer for Robust Optical Flow

Xiuchao Sui[1*] Shaohua Li[1*] Xue Geng[2] Yan Wu[2]
Xinxing Xu[1] Yong Liu[1] Rick Goh[1] Hongyuan Zhu[2]
[1]Institute of High Performance Computing, A*STAR
{xiuchao.sui,shaohua}@gmail.com, {xuxinx,liuyong,gohsm}@ihpc.a-star.edu.sg
[2]Institute for Infocomm Research, A*STAR
{geng_xue,wuy,zhuh}@i2r.a-star.edu.sg

Abstract

Optical flow estimation aims to find the 2D motion field by identifying corresponding pixels between two images. Despite the tremendous progress of deep learning-based optical flow methods, it remains a challenge to accurately estimate large displacements with motion blur. This is mainly because the correlation volume, the basis of pixel matching, is computed as the dot product of the convolutional features of the two images. The locality of convolutional features makes the computed correlations susceptible to various noises. On large displacements with motion blur, noisy correlations could cause severe errors in the estimated flow. To overcome this challenge, we propose a new architecture "CRoss-Attentional Flow Transformer" (CRAFT), aiming to revitalize the correlation volume computation. In CRAFT, a Semantic Smoothing Transformer layer transforms the features of one frame, making them more global and semantically stable. In addition, the dot-product correlations are replaced with transformer Cross-Frame Attention. This layer filters out feature noises through the Query and Key projections, and computes more accurate correlations. On Sintel (Final) and KITTI (foreground) benchmarks, CRAFT has achieved new state-of-the-art performance. Moreover, to test the robustness of different models on large motions, we designed an image shifting attack that shifts input images to generate large artificial motions. Under this attack, CRAFT performs much more robustly than two representative methods, RAFT and GMA. The code of CRAFT is is available at https://github.com/askerlee/craft.

1. Introduction

Optical flow estimates pixel-wise 2D motions between two consecutive video frames by matching corresponding pixels. It is a fundamental computer vision task with broad applications in action recognition [31, 34, 37], video segmentation [43, 45], video frame interpolation [17], medical image registration [28], representation learning [10, 41], autonomous driving [26], and robot navigation [5].

In recent years, deep learning based methods have advanced optical flow estimation tremendously [7, 13, 18, 30, 36, 38, 42, 47]. Although newest methods are very accurate on benchmark data, under certain conditions, such as large displacements with motion blur [9], flow errors could still be large. It spurs us to dig deeper to identify the root causes.

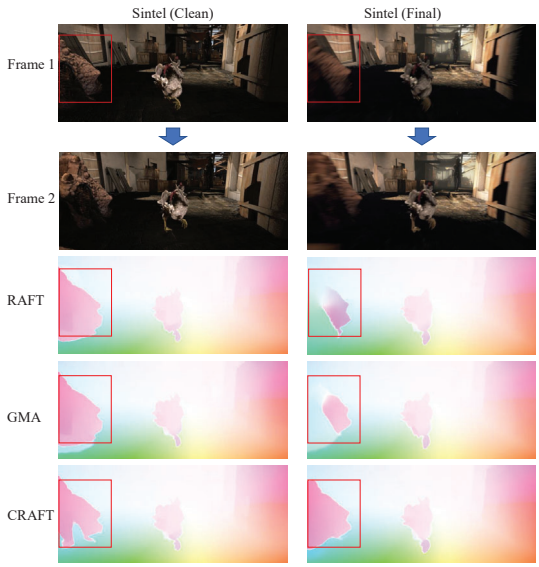


Figure 1. The optical flow fields estimated by RAFT, GMA and CRAFT on two frames from Sintel test set, in which a dragon is chasing a chicken. On the Clean pass, all the three methods perform similarly. On the Final pass, as the area enclosed in the red rectangle has large motions (80 $\sim$ 100 pixels) with motion blur, RAFT and GMA only identified part of the motions. Nonetheless, CRAFT still performs well.

*Equal contribution.

Most of these methods perform optical flow estimation based on a *correlation volume* (also known as a cost volume), which stores the pairwise similarity between each pixel in Frame 1 and another in Frame 2. Given the correlation volume, subsequent modules try to match the two images, with an aim of maximizing the overall correlations between matched regions. The current paradigm computes the pairwise pixel similarity as the *dot product* of two *convolutional* feature vectors. Due to the locality and rigid weights of convolution, limited contextual information is incorporated into pixel features, and the computed correlations suffer from a high level of randomness, such that most of the high correlation values are spurious matches (Figure 6). Noises in the correlations increase with noises in the input images, such as loss of texture, lighting variations and motion blur. Naturally, noisy correlations may lead to unsuccessful image matching and inaccurate output flow (Figure 1). This problem becomes more prominent when there are large displacements. Reducing noisy correlations can lead to substantial improvements of flow estimation [11,46].

Recent years have witnessed the widespread adoption of transformers for computer vision tasks [4, 6]. An important advantage of Vision Transformers (ViTs) over convolution is that, transformer features better encode global context, by attending to pixels with dynamic weights based on their contents. For the optical flow task, useful information can propagate from clear areas to blurry areas, or from non-occluded areas to occluded areas [18], to improve the flow estimation of the latter. A recent study [29] suggests that, ViTs are low-pass filters that do spatial smoothing of feature maps. Intuitively, after transformer self-attention, similar feature vectors take weighted sums of each other, smoothing out irregularities and high-frequency noises.

Inspired by the feature denoising property of ViTs, we propose "CRoss-Attentional Flow Transformer" (CRAFT), a novel architecture for optical flow estimation. With two novel components, CRAFT revitalizes the computation of the correlation volume. First, a *semantic smoothing transformer* layer fuses the features of one image, making them more global and semantically smoother. Second, a *cross-frame attention* layer replaces the dot-product operator for correlation computation. It provides an additional level of feature filtering through the Query and Key projections, so that the computed correlations are more accurate.

We performed extensive evaluations of CRAFT on common optical flow benchmarks. On Sintel (Final) and KITTI (foreground) benchmarks, CRAFT has achieved new state-of-the-art (SOTA) performance. In addition, to test the robustness of different models on large motions, we designed an image shifting attack that shifts input images to generate large artificial motions. As the motion magnitude increases, CRAFT performs robustly, while two representative methods, RAFT and GMA, deteriorate severely.

2. Related Work

FlowNet [7] is a pioneering work that uses deep neural networks to do end-to-end optical flow learning. It inspires a series of deep learning methods, such as FlowNet2.0 [13], DCFlow [42], SpyNet [30], PWC-Net [36], MaskFlowNet [47] LiteFlowNet3 [11], ScopeFlow [2] and IRR [12]. Most of these methods use a *correlation volume* as the basis of pixel matching.

RAFT [38] is an important development of deep learning flow methods. By using multi-scale correlation volumes and iterative flow refinement, RAFT achieves good performance, and is the precursor of a few successive works, such as GMA [18], RAFT-Stereo [21] and CRAFT. GMA [18] is among the first works to incorporate transformer into optical flow methods. In the motion regression stage (cf. Figure 2), it uses self-attention to propagate motion features from non-occluded areas to occluded areas, and helps estimate more accurate flow of occluded areas. It complements with the improvements of CRAFT on correlation volumes.

All the aforementioned methods compute correlations using dot-product or cosine similarity of convolutional features. Within this paradigm, some works improve the efficiency of the correlation volume, such as VCN [44] and DICL [40]. Similar to our objective, Separable Flow [46] aims to improve the accuracy of the correlation volume, by decomposing the 4D correlation volume into two 3D volumes, for the u- and v-directional flow regression, respectively. Separable Flow essentially imposes stronger inductive biases to obtain more accurate correlations than RAFT, as well as more accurate flow[1]. In contrast, CRAFT improves correlation computation by using contextualized frame features and reducing feature noises.

Optical flow training requires large, expensive annotated datasets. SelFlow [22] and Autoflow [35] are two self-supervised methods that generate synthetic annotations. SMURF [33] integrates a set of techniques to do self-supervised learning on unannotated video frames and has achieved promising results.

3. The CRAFT Architecture

Figure 2 presents the architecture of CRAFT. It inherits the influential flow estimation pipeline of RAFT [38]. Our main contribution is to revitalize the correlation volume computation part (the dashed green rectangle) with two novel components: the Semantic Smoothing Transformer on Frame-2 features, and a Cross-Frame Attention Layer to compute the correlation volume. These two components help suppress spurious correlations in the correlation volume, as visualized in Figure 6.

[1] Unfortunately, we could not compare Separable Flow with CRAFT wrt. the correlation volume accuracy, as their source code is unavailable.

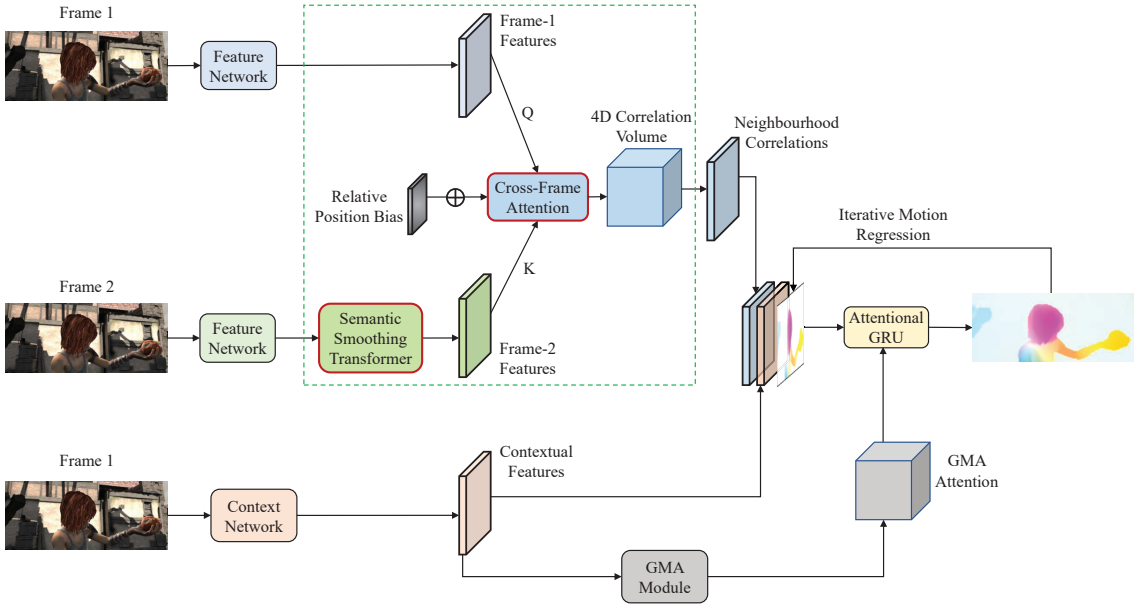


Figure 2. CRAFT architecture. In the correlation volume computation part (the dashed green rectangle), two novel components are highlighted as boxes with red borders: the *Semantic Smoothing Transformer* fuses and smooths the Frame-2 features, and the *Cross-Frame Attention* layer computes the correlation volume. The GMA module at the bottom is a Global Motion Aggregation module [18].

3.1. Semantic Smoothing Transformer

Given two consecutive images – Frame 1 and Frame 2 – as input, the first step of the flow pipeline is to extract frame features using a convolutional feature network.

To enhance the frame features with better global context, the *Semantic Smoothing Transformer* (or simply *SSTrans*) is used to transform the Frame-2 features.

To better accommodate diverse features, we adopt the Expanded Attention proposed in [20] as the SSTrans, instead of the commonly used Multi-Head Attention (MHA) [39]. Expanded Attention is a type of Mixture-of-Experts [32] with higher capacities, and has demonstrated advantages over MHA for image segmentation tasks.

An Expanded Attention (EA) layer consists of N modes (sub-transformers), computing N sets of features, which are aggregated into one set using dynamic mode attention [20]:

$$\boldsymbol{X}_{out}^{(k)} = \text{Transformer}^{(k)}(\boldsymbol{X}), \tag{1}$$

$$\boldsymbol{B}^{(k)} = \text{Linear}^{(k)}(\boldsymbol{X}_{out}^{(k)}), \tag{2}$$

$$\text{with } k \in \{1, \cdots, N\}, \tag{3}$$

$$\boldsymbol{G} = \text{softmax}\left(\boldsymbol{B}^{(1)}, \cdots, \boldsymbol{B}^{(N)}\right), \tag{4}$$

$$\text{EA}(\boldsymbol{X}) = \boldsymbol{G}^{\top} \cdot \left(\boldsymbol{X}_{out}^{(1)}, \cdots, \boldsymbol{X}_{out}^{(N)}\right), \tag{5}$$

where $\boldsymbol{B}^{(k)}$ are mode attention scores, and the mode attention probabilities G are softmax of all $\boldsymbol{B}^{(k)}$ along the mode dimension. The output features $\text{EA}(\boldsymbol{X})$ are a linear combination of all mode features.

To better preserve original frame features, we add a weighted skip connection with a learnable weight w_1:

$$\text{SSTrans}(\boldsymbol{X}) = w_1 \boldsymbol{X} + (1 - w_1)\,\text{EA}(\boldsymbol{X}), \tag{6}$$

To impose spatial biases, we found conventional positional embeddings do not form meaningful biases, and use a *relative position bias* [8, 23] instead. The bias is a matrix $B \in \mathbb{R}^{(2r+1)\times(2r+1)}$, added to the computed attention, where r is the radius specifying the local range of the bias.

Specifically, suppose the original attention matrix is reshaped to a 4-dimensional tensor $A \in \mathbb{R}^{H\times W\times H\times W}$, where H, W are the height and width of the frame feature maps. For each pixel at i, j, where $i \in \{1, \cdots, H\}, j \in \{1, \cdots, W\}$, $A(i, j)$ is a matrix, specifying the attention weights between pixel (i, j) with all the pixels in the same frame. The relative position bias B is added to the neighborhood of radius r of pixel (i, j):

$$A'(i, j, i + x, j + y) = \begin{cases} A(i, j, i + x, j + y) + B(x, y), & \text{if } |x| \le r, |y| \le r \\ A(i, j, i + x, j + y). & \text{otherwise} \end{cases} \tag{7}$$

In our implementation, we choose the number of modes to be 4, and the radius r of the relative position bias to be 7.

Figure 3 visualizes the learned relative position bias of CRAFT trained on Sintel. Two interesting patterns are observed:

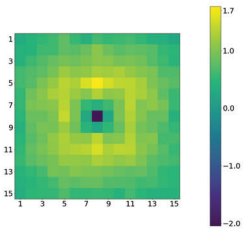


Figure 3. Learned relative positional bias with radius $r = 7$. Two interesting patterns can be observed, as detailed below.

1. The minimum bias value is around -2 located at $(0, 0)$, which means that, when computing the new features of a pixel (i, j), this bias term will *reduce the weight of its own features* by 2. Without this term, the attention weight of pixel (i, j) to itself will probably dominate the weights to other pixels, as a feature vector is most similar to itself. This term reduces the proportions of the old features of a pixel in the combined output features, effectively *encouraging inflow of new information from other pixels*.

2. The largest weights are $2 \sim 3$ pixels[2] away from the center pixel, meaning that features of these surrounding pixels are most often used to supplement the features of the central pixel.

These two observations are confirmed in Figure 8, where each query draws new features from a nearby area. Setting the position bias to 0 leads to performance degradation.

It is tempting to apply transformers on the features of both frames. However, in our experiments, doing so leads to performance drop. Our hypothesis is based on the common belief that image matching heavily relies on high-frequency (HF) features that are local and structural [14]. Meanwhile, there are abundant HF noises that pollute informative features and hinder matching. SSTrans serves as a low-pass filter to suppress HF noises [29], but at the same time, may reduce HF features and enhance low-frequency (LF) features. Hence, the model learns to trade off between the LF and HF components in Frame 2 for matching with Frame 1. After applying SSTrans on both frames, both frames contain less HF and more LF components. Matching them may yield many spurious correlations and hurt flow accuracy. This intuition is confirmed in Figure 7.

3.2. Cross-Frame Attention for Correlation Volume

In the current paradigm, a correlation volume is the basis of cross-frame pixel matching. After the frame features $\boldsymbol{f}_1 \in \mathbb{R}^{H\times W\times D}$ and $\boldsymbol{f}_2 \in \mathbb{R}^{H\times W\times D}$ are computed, the correlation volume is computed as a 4D tensor $\boldsymbol{C} \in \mathbb{R}^{H\times W\times H\times W}$ (dashed green rectangle in Figure 2).

Traditionally, the correlation volume is computed as the pairwise dot-product between $\boldsymbol{f}_1$ and $\boldsymbol{f}_2$ [38]:

$$C(i, j, m, n) = \frac{1}{\sqrt{D}} \boldsymbol{f}_1(i, j)^\top \cdot \boldsymbol{f}_2(m, n). \tag{8}$$

Conceptually, the correlation volume is essentially Cross Attention [39] in transformers, without feature transformation by the Query and Key projections. The query/key projections can be viewed as feature filters that separate out most informative features for correlations. In addition, to capture diverse correlations, we could use multiple query and key projections, as with Expanded Attention (EA) [20]. Similar multi-faceted correlations are pursued in VCN [44] with multiple channels. These benefits motivate us to replace the dot-product with a simplified EA:

$$C_k(i, j, m, n) = \frac{1}{\sqrt{D}} (\boldsymbol{f}_1(i, j)\boldsymbol{Q}_k)^\top \cdot \boldsymbol{K}_k \boldsymbol{f}_2(m, n), \tag{9}$$

$$C(i, j, m, n) = \sum_{k=1}^{K} \mathrm{softmax}(C_k(i, j, m, n)) C_k(i, j, m, n), \tag{10}$$

where $\boldsymbol{Q}_k$, $\boldsymbol{K}_k$ are the k-th query and key projections, respectively; $C_k(i, j, m, n)$ is the correlation computed with the k-th mode. The $\mathrm{softmax}$ operator is taken along the K modes, and aggregates the K correlations. The EA here is simplified by removing the value projection and the feed-foward network. The weights of $\boldsymbol{Q}_k$ and $\boldsymbol{K}_k$ are tied, as the correlation between two frames is symmetric.

Global correlation normalization Sometimes extreme values may appear in the correlation volume, which may disrupt the pixel matching. To match a pixel, intuitively the relative orders of the correlations with candidate pixels are more important than absolute correlation values. In this light, we perform layer normalization [1] on the whole correlation volume to stabilize correlations. Empirically, this leads to slightly improved performance.

4. Experiments

Our experiments consist of six parts:

1. **Standard evaluation.** We evaluate different methods on Sintel [3] and KITTI [27]. On the two public leaderboards, CRAFT has achieved the state-of-the-art performance on both Sintel (final pass) and KITTI (foreground regions).

2. **Error distribution wrt. motion magnitudes.** To study the model behavior when the motion becomes larger, we calculate the flow error distribution wrt. different magnitudes of motions. CRAFT is significantly more accurate than other methods on *large motions*, and performs equally well on small motions.

[2] Here "pixels" mean points in feature maps, which correspond to $\times 8$ pixels in the input image.

3. **Ablation studies.** To analyze the impact of different components in CRAFT, i.e., the Semantic Smoothing transformer, the Cross-Frame Attention and the GMA module, we remove each of them and evaluate the ablated models on the KITTI-2015 benchmark. All these components show importance to the final performance.

4. **Image Shifting attack.** To test the robustness of models, we manually create large motions by shifting the first frames. At very large shifts, RAFT and GMA deteriorate severely. CRAFT is significantly more robust.

5. **Visualization of correlation volumes.** We visualize the correlations between a query point in Frame 1 and all pixels in Frame 2, to intuitively learn the differences between the correlation volumes computed by different models. CRAFT has the fewest spurious correlations compared with RAFT and GMA.

6. **Visualization of semantic smoothing transformer attention.** To gain an intuitive idea how a pixel draws information from surrounding pixels through the SS transformer, we visualize the self-attention between a query point and all pixels in Frame 2.

Training Loss Following RAFT [38], the loss function we adopt is a weighted multi-iteration l_1 loss.

Training Schedule We follow the same optical flow training procedure [18, 38] of first pretraining the models on FlyingChairs ("C") [7] for 120k iterations (batch size = 8), then on FlyingThings ("T") [25] for another 120k iterations with (batch size = 6). For Sintel evaluation, we fine-tune all models on a combination of FlyingThings, Sintel ("S") [3], KITTI 2015 ("K") [27] and HD1K ("H") [19] for 120k iterations (batch size = 6). For KITTI evaluation, we fine-tune all models on KITTI 2015 for 50k iterations (batch size = 6). Following [18, 38], we adopt the one-cycle learning rate scheduler with the same learning rates, in which 5 percent of the iterations are used for warm-up.

Evaluation Metrics The main evaluation metric, also used by the Sintel leaderboard[3], is the average end-point error (AEPE), which is the average pixelwise flow error, measured by number of pixels. The KITTI leaderboard[4] uses the Fl-fg (%) and Fl-All (%) metrics, which refer to the percentage of outliers (pixels whose end-point error is > 3 pixels or 5% of the ground truth flow magnitude), averaged over foreground regions and all pixels, respectively.

4.1. Standard Evaluation

Seven recent methods are compared, most of which are selected from the top-performing methods on the Sintel and KITTI leaderboards:

- **RAFT** [38]: an important recent methods, and was previous SOTA before being surpassed by GMA.
- **RAFT-A** [35] uses the synthesized AutoFlow dataset (instead of "C+T") to pretrain RAFT, followed by the standard fine-tuning steps.
- **Perceiver-IO** [15] is a general architecture not specifically designed for optical flow estimation. It is pretrained on Autoflow, the same as RAFT-A. The performance on the test sets is not reported in their paper.
- **RFPM** [24] replaces the downsampling layers of RAFT to improve the flow estimation on fine details. The performance under "C+T / Autoflow" training is not reported in their paper.
- **Separable Flow** [46] decomposes the 4D correlation volume as two 3D volumes for the u and v directions.
- **GMA** [18]: a recent method that enhances RAFT with a Global Motion Aggregation module to better estimate the motions of occluded pixels.
- **CRAFT**: with 4 modes in expanded attention layers.

Table 1 summarizes the evaluation results of the seven methods on Sintel and KITTI. The results on the training sets (in parentheses, left side of the table) can hardly reflect how well the models generalize to new data, and are only listed for reference. The results on the test sets are evaluated on held-out data by the Sintel and KITTI servers and obtained from their leaderboards, and better reflect model performance. Although performing closely to the other methods on the training sets, CRAFT shows clear advantages on the test sets, and outperforms all other optical flow methods[5] on Sintel (Final) and KITTI Fl-fg (i.e., fewest foreground outliers).

We argue that these two performance metrics (AEPE on Sintel Final pass, and Fl-fg on KITTI) has important practical implications. For real world performance, the results on Sintel (Final) are more indicative than on Sintel (Clean), as the final-pass images more closely resemble real world videos, with various lighting variation, shadows and motion blur. In addition, as the foreground objects in KITTI are usually cars, pedestrians, etc., which naturally are more important than the background. Hence, smaller pixel errors in foreground regions as measured by Fl-fg, probably imply greater practical benefits than smaller errors in background.

4.2. Error Distribution wrt. Motion Magnitudes

To analyze the behavior of different models when facing varying magnitudes of motions, we divide the pixels into

[3] http://sintel.is.tue.mpg.de/quant?metric_id=0&selected_pass=0

[4] http://www.cvlibs.net/datasets/kitti/eval_scene_flow.php?benchmark=flow

[5] As of November 2021.

Training Data	Method	On Training Sets				On Test Sets from Leaderboards			
		Sintel (train)		KITTI-15 (train)		Sintel (test)		KITTI-15 (test)	
		Clean	Final	AEPE	Fl-all (%)	Clean	Final	Fl-fg (%)	Fl-all (%)
C + T / Autoflow	RAFT [38]	(1.43)	(2.71)	(5.04)	(17.4)	-	-	-	-
	RAFT-A [35]	(1.95)	(2.57)	(**4.23**)	-	-	-	-	
	Perceiver-IO [15]	(1.81)	(**2.42**)	(4.98)	-	-	-	-	
	Separable Flow [46]	(1.30)	(2.59)	(4.60)	(**15.9**)	-	-	-	-
	GMA [18]	(1.30)	(2.74)	(4.69)	(17.1)	-	-	-	-
	CRAFT	(**1.27**)	(2.79)	(4.88)	(17.5)	-	-	-	-
C + T + S/K + H	RAFT [38]	(0.76)	(1.22)	(0.63)	(1.5)	1.61	2.86	6.87	5.10
	RAFT-A [35]	-	-	-	-	2.01	3.14	5.99	4.78
	RFPM [24]	(0.61)	(**1.05**)	(0.60)	(1.41)	1.41	2.90	-	4.79
	Separable Flow [46]	(0.69)	(1.10)	(0.69)	(1.60)	1.50	2.67	6.24	**4.64**
	GMA [18]	(0.62)	(1.06)	(**0.57**)	(**1.20**)	1.39	2.47	7.03	5.15
	CRAFT	(**0.60**)	(1.06)	(0.58)	(1.34)	1.45	**2.42**†	**5.85**†	4.79

Table 1. **Results on Sintel and KITTI 2015 benchmarks.** We report the average end-point error (AEPE) where not otherwise stated, as well as the Fl-fg and Fl-all metrics for the KITTI dataset, which are the percentages of optical flow outliers (pixels with significant flow errors), calculated on the foreground regions and all pixels, respectively. "C + T / Autoflow" refers to methods that are pretrained either on the combined Chairs and Things datasets, or on the Autoflow dataset [35]. "S/K + H" refers to methods that are fine-tuned on the Sintel, KITTI and HD1K datasets. All results on Sintel (test) are generated with the "warm-start" strategy [38].
†Results are ranked as the top-1 (as of November 2021) on the two public leaderboards, which include many other methods not listed here. (Result) denotes a result on *training sets*, listed here for reference purposes.

GT range	< 1	[1,10]	(10,20]	(20,30]	> 30	All
Things-Clean						
RAFT	0.45	0.54	0.75	1.40	7.55	3.14
GMA	**0.42**	**0.46**	**0.68**	1.29	7.71	3.14
CRAFT	0.43	**0.46**	**0.68**	**1.26**	**6.64**	**2.77**
Things-Final						
RAFT	0.46	0.52	0.74	1.44	7.11	2.98
GMA	**0.41**	**0.45**	0.68	1.25	6.76	2.80
CRAFT	0.42	**0.45**	**0.65**	**1.21**	**6.11**	**2.57**

Table 2. **AEPE on Things (validation set) in different motion ranges**. CRAFT has significantly lower AEPE on large motions.

five subsets according to their groundtruth motion magnitudes, and evaluate the AEPE within each subset. As the validation/test splits of Sintel and KITTI are unavailable, the evaluation is done on the validation split of FlyingThings, Clean pass and Final pass, respectively. Three models, RAFT, GMA and CRAFT are evaluated. All the models are trained on "C+T".

Table 2 presents the AEPE on different magnitudes of motions. When the motion is < 20 pixels, CRAFT performs on par with GMA. On large motions that are > 30 pixels, CRAFT makes 10$\sim$15% less AEPE than RAFT and GMA.

4.3. Ablation Studies

KITTI-15 (test)	Fl-fg (%)	Fl-all (%)
CRAFT	5.85	4.79
-SS trans	6.41	5.06
-CFA	6.15	4.90
-GMA	6.21	4.93

Table 3. **Ablated models on KITTI-2015 (test) leaderboard**.

CRAFT has three important components: the Semantic Smoothing transformer ("SS trans"), the Cross-Frame Attention ("CFA"), and the GMA module. To study their individual contributions, in each turn we remove one of them, train the ablated models with the standard schedule, and evaluate on the KITTI-2015 leaderboard.

Table 3 shows that all the three components make important contributions to the overall performance.

4.4. Image Shifting Attack

Typically, most pixels in standard benchmark images are with small motions, and large motions only appear in local areas. As a result, when the model makes big errors on large local motions, as these errors are local, they may

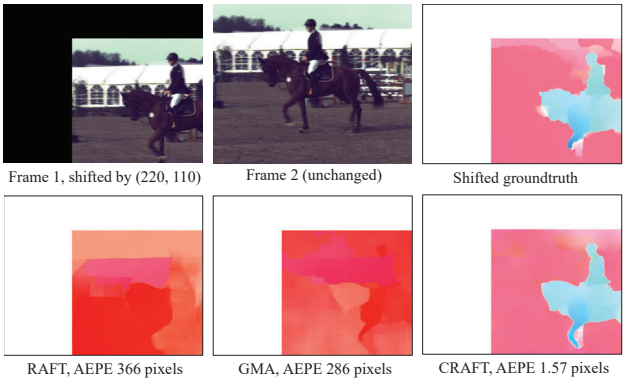


Figure 4. Flows fields estimated by RAFT, GMA and CRAFT on two frames from the Slow Flow dataset. $(\Delta u, \Delta v) = (220, 110)$ pixels. RAFT and GMA failed with huge AEPE. CRAFT still yielded accurate estimation.

be easily corrected by considering the contextual small motions, so that the final flow may still be accurate. Thereby, the fragility on large motions is hidden under small AEPE.

To fully reveal the model robustness on large motions, we design an image shifting attack, i.e., create large motions by shifting one image along the u, v plane. Local corrections would hardly work on such image pairs, as all the pixels will have large displacements.

Specifically, we shift the first frame I_1 by $(\Delta u, \Delta v)$ towards the bottom right, getting a new image $\text{shift}_{u,v}(I_1)$. The new image is truncated at the original image boundary.

Suppose a model M estimates the flow F_0 accurately on the original image pairs: $F_0 = M(I_1, I_2) \approx F_{gt}$, where F_{gt} is the groundtruth flow. We test M on the shifted pairs and get new flow: $F_1 = M(\text{shift}_{u,v}(I_1), I_2)$. Then we unshift F_1 and get F_2. If the model is robust against the shift, it can be proven that the following equation should hold:

$$F_2 \approx \text{shift}_{u,v}(F_0) - (\Delta u, \Delta v) \approx \text{shift}_{u,v}(F_{gt}) - (\Delta u, \Delta v). \tag{11}$$

Figure 4 presents an example of the shifting attack. The two frames are from Slow Flow [16], a dataset with motion blur (flow magnitude=100, blur duration=3). After downsampling the original images from (1280, 720) to (640, 360), the first image is shifted by $(220, 110)$. RAFT and GMA completely fail to estimate the flow, with huge AEPE. In contrast, CRAFT still yields accurate estimation.

Figure 5 presents the quantitative evaluations of RAFT, GMA and CRAFT under the shifting attack. The models are trained with "C+T+S+K+H", and evaluated on the training split of Sintel (Clean) and Sintel (Final), as well as on Slow Flow (flow magnitude=100, blur duration=3), under varying $(\Delta u, \Delta v)$. In our experiments, the horizontal shift $\Delta u \in [100, 300]$, and the vertical shift $\Delta v \doteq \frac{1}{2}\Delta u$. When $\Delta u \leq 160$, all models perform well with AEPE < 8. When Δu goes beyond 160, RAFT and GMA quickly deteriorate; in contrast, CRAFT performs much more robustly with significantly smaller AEPE. Possibly due to motion blur, the AEPE of RAFT and GMA on Slow Flow is $80 \sim 100$ pixels larger than on Sintel, while the AEPE of CRAFT on Slow Flow is only 35 pixels larger, showing its robustness against motion blur.

4.5. Visualization of Correlation Volumes

The main reason that CRAFT performs more robustly is probably that the computed correlation volumes contains much fewer spurious correlations, thanks to the SS transformer and the cross-frame attention layer.

To gain an intuitive understanding of the differences between the correlation volumes computed by different models, we visualize the correlations between a query point in Frame 1 and all pixels in Frame 2. The query point is marked as a small red square in Frame 1 (projected to the small green square in Frame 2). It moves to the small red square in Frame 2. The dashed green rectangle is a 256×256-pixel square centered at the query point, truncated at the image boundary. It encloses the field of view (FoV) of the model at the first iteration of flow estimation. Only correlations within the FoV are shown.

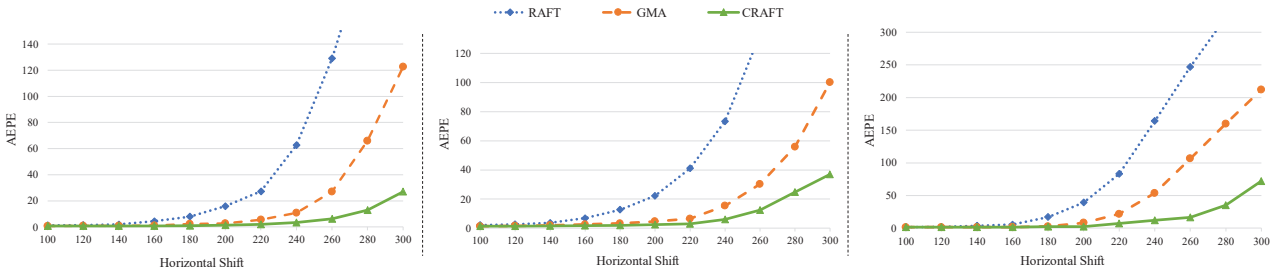
Figure 6 visualizes the correlation volumes on two frames from Sintel (Final), which is rendered with shadows and motion blur. Bright blobs in the heatmaps are high correlations, and those not at the groundtruth location (red square) are spurious and may be targets for mismatch. The correlation volumes[6] computed by RAFT and GMA contain many more spurious correlations than CRAFT. If removing the SS transformer (the cross-frame attention layer remains), CRAFT yields more noisy correlations, but they are still fewer than RAFT and GMA, suggesting that the cross-frame attention layer also helps denoising.

In addition, as stated in Section 3.1, we tested to apply the SS transformer to both Frame 1 and Frame 2 (referred to as "Double SSTrans"), and observed degraded performance. To shed light on why this happens, Figure 7 visualizes the computed correlations with Double SSTrans. Compared with the standard "Single SSTrans", many more spurious correlations are observed. This may explain the degradation of the flow accuracy.

4.6. Visualization of the Self-Attention of Semantic Smoothing Transformer

Figure 8 visualizes the SS transformer self-attention weights on three queries in Frame 2. For each query (the small red square), its attention weights with all pixels in the

[6] All matrices have been normalized into $[0, 1]$ to make sure the pattern differences are not caused by range discrepancy.



(a) Sintel (Clean) (b) Sintel (Final) (c) Slow Flow (100-3)

Figure 5. The AEPE of RAFT, GMA and CRAFT change differently with the magnitude of image shifts. (a)-(c) are on Sintel (Clean), Sintel (Final) and Slow Flow, respectively. The horizontal shift Δu change from 100 to 300, and the vertical shift $\Delta v \doteq \frac{1}{2}\Delta u$. When Δu goes beyond 160, RAFT and GMA quickly deteriorate, and CRAFT performs much more robustly.

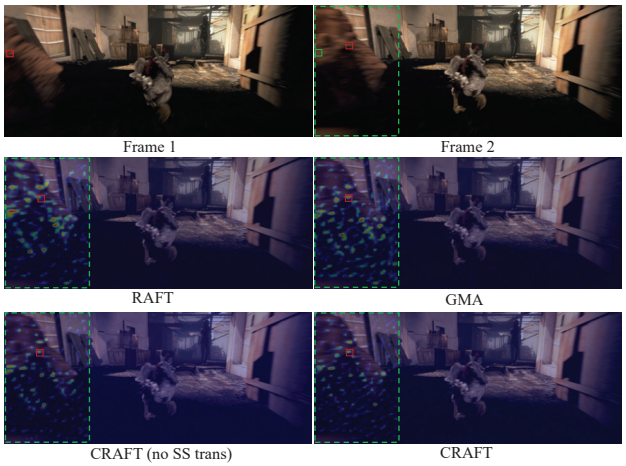


Figure 6. Heatmaps of the correlations between Frame 2 and a query point in Frame 1 (the small red square), on Sintel test set (Final pass). The small green square in Frame 2 indicates the original position of the query in Frame 1. As the images are blurry with coarser details, RAFT and GMA make many noisy correlations. In contrast, CRAFT has significantly fewer noisy correlations.

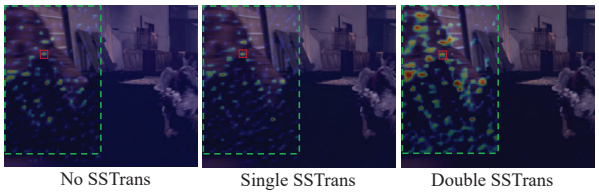


Figure 7. The correlations between Frame 2 and a query point in Frame 1, on Sintel test set (Final pass). Images are cropped. The standard CRAFT setting ("Single SSTrans") has fewest noisy correlations. "Double SSTrans" yields many more noisy correlations.

same image are displayed as a heatmap. The highest attention areas are somewhere around the query points (at different relative directions). We guess that these areas may provide texture or contextual information absent at the queries.

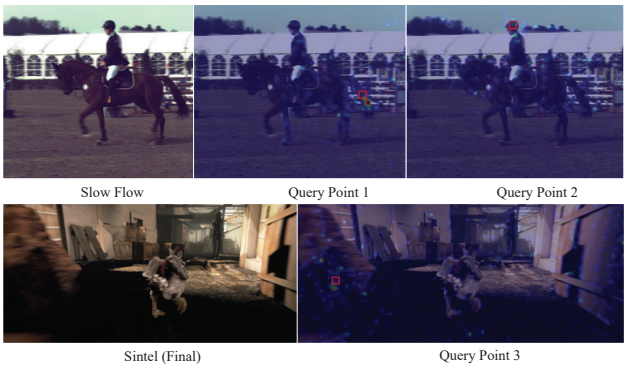


Figure 8. Heatmaps of the SS transformer self-attention, between a query point (a red rectangle) and all pixels in the same image. The most intense areas are where the query points pay the highest attention and draw features to enrich themselves.

5. Conclusions

We present a novel optical flow estimation method *Cross-Attentional Flow Transformer* (CRAFT). It revitalizes the computation of correlation volumes with two novel components: Semantic-Smoothing Transformer and Cross-Frame Attention. They help compute more accurate correlation volumes by spatially smoothing feature semantics and filtering out feature noises. CRAFT has achieved new state-of-the-art performance on a few metrics, and is especially robust on large displacements with motion blur.

Acknowledgements

This research is supported by A*STAR under its Career Development Fund (Grant Nos. C210812035 and C210112016), and its Human-Robot Collaborative AI for Advanced Manufacturing and Engineering programme (Grant No. A18A2b0046).

References

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv:1607.06450*, 2016. 4

[2] Aviram Bar-Haim and Lior Wolf. Scopeflow: Dynamic scene scoping for optical flow. In *CVPR*, June 2020. 2

[3] Daniel J. Butler, Jonas Wulff, Garrett B. Stanley, and Michael J. Black. A naturalistic open source movie for optical flow evaluation. In Andrew Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato, and Cordelia Schmid, editors, *ECCV*, 2012. 4, 5

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-End Object Detection with Transformers. In *ECCV*, 2020. 2

[5] G. C. H. E. de Croon, C. De Wagter, and T. Seidl. Enhancing optical-flow-based control by learning visual appearance cues for flying robots. *Nature Machine Intelligence*, 3(1):33–41, 2021. 1

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 2

[7] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Häusser, Caner Hazirbas, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *ICCV*, 2015. 1, 2, 5

[8] Philipp Dufter, Martin Schmitt, and Hinrich Schütze. Position information in transformers: An overview. *arXiv:2102.11090*, 2021. 3

[9] Fatma Guney, Laura Sevilla-Lara, Deqing Sun, and Jonas Wulff. "what is optical flow for?": Workshop results and summary. In *ECCV Workshops*, 2018. 1

[10] Tengda Han, Weidi Xie, and Andrew Zisserman. Self-supervised co-training for video representation learning. In *NeurIPS*, volume 33, 2020. 1

[11] Tak-Wai Hui and Chen Change Loy. LiteFlowNet3: Resolving Correspondence Ambiguity for More Accurate Optical Flow Estimation. In *ECCV*, 2020. 2

[12] Junhwa Hur and Stefan Roth. Iterative residual refinement for joint optical flow and occlusion estimation. In *CVPR*, June 2019. 2

[13] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 1, 2

[14] M. Irani and P. Anandan. Robust multi-sensor image alignment. In *ICCV*, 1998. 4

[15] Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock, Evan Shelhamer, Olivier J. Hénaff, Matthew M. Botvinick, Andrew Zisserman, Oriol Vinyals, and João Carreira. Perceiver IO: A general architecture for structured inputs & outputs. *arxiv:2107.14795*, 2021. 5, 6

[16] Joel Janai, Fatma Güney, Jonas Wulff, Michael Black, and Andreas Geiger. Slow flow: Exploiting high-speed cameras for accurate and diverse optical flow reference data. In *CVPR*, 2017. 7

[17] Huaizu Jiang, Deqing Sun, Varun Jampani, Ming-Hsuan Yang, Erik Learned-Miller, and Jan Kautz. Super slomo: High quality estimation of multiple intermediate frames for video interpolation. In *CVPR*, June 2018. 1

[18] Shihao Jiang, Dylan Campbell, Yao Lu, Hongdong Li, and Richard I. Hartley. Learning to estimate hidden motions with global motion aggregation. In *ICCV*, 2021. 1, 2, 3, 5, 6

[19] Daniel Kondermann, Rahul Nair, Katrin Honauer, Karsten Krispin, Jonas Andrulis, Alexander Brock, Burkhard Gussefeld, Mohsen Rahimimoghaddam, Sabine Hofmann, Claus Brenner, and Bernd Jahne. The hci benchmark suite: Stereo and flow ground truth with uncertainties for urban autonomous driving. In *CVPR Workshops*, 2016. 5

[20] Shaohua Li, Xiuchao Sui, Xiangde Luo, Xinxing Xu, Yong Liu, and Rick Goh. Medical image segmentation using squeeze-and-expansion transformers. In *IJCAI*, 2021. 3, 4

[21] Lahav Lipson, Zachary Teed, and Jia Deng. Raft-stereo: Multilevel recurrent field transforms for stereo matching. In *3DV*, 2021. 2

[22] Pengpeng Liu, Michael Lyu, Irwin King, and Jia Xu. Selflow: Self-supervised learning of optical flow. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2

[23] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, pages 10012–10022, October 2021. 3

[24] Libo Long and Jochen Lang. Detail preserving residual feature pyramid modules for optical flow. *arXiv:2107.10990*, 2021. 5, 6

[25] Nikolaus Mayer, Eddy Ilg, Philip Häusser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *CVPR*, 2016. 5

[26] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3061–3070, 2015. 1

[27] Moritz Menze, Christian Heipke, and Andreas Geiger. Joint 3d estimation of vehicles and scene flow. In *Proc. of the ISPRS Workshop on Image Sequence Analysis (ISA)*, 2015. 4, 5

[28] Sergiu Mocanu, Alan R. Moody, and April Khademi. Flowreg: Fast deformable unsupervised medical image registration using optical flow. *Machine Learning for Biomedical Imaging*, 1, 2021. 1

[29] Namuk Park and Songkuk Kim. How do vision transformers work? In *ICLR*, 2022. 2, 4

[30] Anurag Ranjan and Michael J. Black. Optical flow estimation using a spatial pyramid network. In *CVPR*, 2017. 1, 2

[31] Laura Sevilla-Lara, Yiyi Liao, Fatma Güney, Varun Jampani, Andreas Geiger, and Michael J. Black. On the integration of

optical flow and action recognition. In *German Conference on Pattern Recognition*, 2018. 1

[32] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *ICLR*, 2017. 3

[33] A. Stone, D. Maurer, A. Ayvaci, A. Angelova, and R. Jonschkowski. Smurf: Self-teaching multi-frame unsupervised raft with full-image warping. In *CVPR*, 2021. 2

[34] Chen Sun, Abhinav Shrivastava, Carl Vondrick, Kevin Murphy, Rahul Sukthankar, and Cordelia Schmid. Actor-centric relation network. In *ECCV*, 2018. 1

[35] D. Sun, D. Vlasic, C. Herrmann, V. Jampani, M. Krainin, H. Chang, R. Zabih, W. T. Freeman, and C. Liu. Autoflow: Learning a better training set for optical flow. In *CVPR*, 2021. 2, 5, 6

[36] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. In *CVPR*, 2018. 1, 2

[37] Shuyang Sun, Zhanghui Kuang, Lu Sheng, Wanli Ouyang, and Wei Zhang. Optical flow guided feature: A fast and robust motion representation for video action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 1

[38] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *European conference on computer vision*, pages 402–419. Springer, 2020. 1, 2, 4, 5, 6

[39] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, undefinedukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 3, 4

[40] Jianyuan Wang, Yiran Zhong, Yuchao Dai, Kaihao Zhang, Pan Ji, and Hongdong Li. Displacement-invariant matching cost learning for accurate optical flow estimation. In *NeurIPS*, 2020. 2

[41] Yuwen Xiong, Mengye Ren, Wenyuan Zeng, and Raquel Urtasun. Self-supervised representation learning from flow equivariance. In *ICCV*, 2021. 1

[42] Jia Xu, René Ranftl, and Vladlen Koltun. Accurate Optical Flow via Direct Cost Volume Processing. In *CVPR*, 2017. 1, 2

[43] Charig Yang, Hala Lamdouar, Erika Lu, Andrew Zisserman, and Weidi Xie. Self-supervised video object segmentation by motion grouping. In *ICCV*, 2021. 1

[44] Gengshan Yang and Deva Ramanan. Volumetric correspondence networks for optical flow. In *NeurIPS*, 2019. 2, 4

[45] Gengshan Yang and Deva Ramanan. Learning to segment rigid motions from two frames. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1266–1275, June 2021. 1

[46] Feihu Zhang, Oliver J. Woodford, Victor Adrian Prisacariu, and Philip H.S. Torr. Separable flow: Learning motion cost volumes for optical flow estimation. In *ICCV*, 2021. 2, 5, 6

[47] Shengyu Zhao, Yilun Sheng, Yue Dong, Eric I-Chao Chang, and Yan Xu. Maskflownet: Asymmetric feature matching with learnable occlusion mask. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 2