

# Frame interpolation in video stream using optical flow methods

**S Lyasheva, R Rakhmankulov and M Shleymovich**

Kazan National Research Technical University named after A. N. Tupolev - KAI,  
10 K. Marks, Kazan, 420111, Russia

E-mail: mpsheymovich@kai.ru

**Abstract.** The article deals with the issues of interpolation and extrapolation of video stream frames in real time. Results of research of the method of generation of intermediate frames of a video stream are given. The method is based on the optical flow calculate. The results of comparison of implementations of the method using the Farneback, Brox and Duality based TV-L1 approaches for the calculation of optical flow and CUDA technology are presented. It is shown that the Farneback method best result is gives. The Duality based TV-L1 method shows a smooth and high-quality optical flow, but gives a very slow calculation speed. The Brox method is not suitable at all, since it shows the worst result both in terms of calculation speed and optical flow quality. The results of comparison of implementations based on CUDA and OpenCL technologies are also presented. OpenCL implementations are almost two times slower than CUDA implementation. This leads to the inability to generate frames in 60 FPS in real time even for low resolution images. However, CUDA can work only on NVidia GPU accelerators. In conclusion, the ways of further research on this topic based on the use of deep neural networks are presented.

## 1. Introduction

Motion interpolation is a task in video processing. The goal is to create an intermediate frame or number of frames between existing frames. There are numerous works which apply to this problem [1-3]. But most of them aim motion interpolation to video streams where all existing frames are known. They use currently in TV devices and some software for smoothing videos by increasing frame rate to 60 or more Frames per second (FPS).

In opposite to these approaches there are ones in which only current and previous frames are known. Variation of this kind of methods were presented in VR by Oculus, Microsoft for WMR headsets and in SteamVR platform for HTC Vive headsets. But the main problem is that these methods are strongly connected with VR and hardware.

In this paper we present a method, which can handle hardware independent frame interpolation when only current and previous frames are known. By applying interpolation and extrapolation techniques in combination with optical flow intermediate frames can be generated.

In this paper the authors made the following contribution:

- Simple yet good method for frame interpolation;
- Generation for additional frames in video streams where only current and previous frames are known.



## 2. Related Work

### 2.1. Optical Flow

Optical flow is the method utilized in various fields of computer vision to determine shifts, segmentation, feature extraction, video compression.

Optical flow is a picture of visible motion, representing shift of each pixel between two images. In fact, it represents the velocity field.

To define pixels compliance one should take some pixel function, which is not changed as a result of displacement. It is generally considered that pixel intensity (brightness or color) remains unchanged. Obvious, intensity changes when light is changing or the light angle. However difference between frames in video stream usually are not so significant, because time delta between them is relatively small. So the intensity is usually used as pixel function.

In time perspective, a pixel  $I_1(x, y, t)$  is shifted by  $(\Delta x, \Delta y)$  in next frame after time  $\Delta t$ . Since the pixels on both frames should not be changed, we can assume that intensities should be the same, then:

$$I_1(x, y, t) = I_2(x + \Delta x, y + \Delta y, t + \Delta t) \quad (1)$$

Let approximate  $I_1(x, y, t)$  by Taylor series, then remove high order terms and divide by  $\Delta t$ :

$$I_2(x + \Delta x, y + \Delta y, t + \Delta t) = I_1(x, y, t) + \frac{\partial I}{\partial x} \frac{\Delta x}{\Delta t} + \frac{\partial I}{\partial y} \frac{\Delta y}{\Delta t} + \frac{\partial I}{\partial t} \quad (2)$$

If the  $I_1 = I_2$  then:

$$\frac{\partial I}{\partial x} \frac{\Delta x}{\Delta t} + \frac{\partial I}{\partial y} \frac{\Delta y}{\Delta t} + \frac{\partial I}{\partial t} = 0 \quad (3)$$

which result:

$$I_x u + I_y v + I_t = 0 \quad (4)$$

where  $u = \frac{\Delta x}{\Delta t}$  and  $v = \frac{\Delta y}{\Delta t}$  represent the velocity of  $x$  and  $y$  respectively and  $I_x = \frac{\partial I}{\partial x}$ ,  $I_y = \frac{\partial I}{\partial y}$ ,

$I_t = \frac{\partial I}{\partial t}$  are partial derivatives of pixel function of coordinates  $x$  and  $y$ . So now we have two unknowns and equation cannot be solved as it is. To solve this equation some additional constraints have to be provided. Example of optical flow calculate is shown in figure 1.

There exist some methods to solve this equation. In this paper used next methods: Farneback method [4], Brox method [5, 6] and Duality based TV-L1 method [7].



**Figure 1.** Example of optical flow calculate.

### 2.2. Motion Interpolation

Interpolation itself is a method which allows predicting data points inside the given range of the data. Using interpolation one can approximate function for some dataset or find additional points, in addition one can also make approximation of the complex functions. In perspective of this work, more interesting aspect is to find additional data points on given dataset.

Motion interpolation is the variation of interpolation method, which is mostly used in video processing as the way to smooth animations by interpolating intermediate frames between existing ones. In addition, it finds application in video compressing; instead of containing every frame in video stream some frames contain only motion difference according to previous frame and could be reconstructed from it.

Modern displays have this method as a feature, it allows to increase frame rate on a display. For example, some TV's can show 120 FPS, but content doesn't have such frame count, so the TV itself may interpolate additional frames and show 120 FPS instead of 60 FPS. Also, some media players implement such feature, one of them was mentioned before is SVP video player.

Extrapolation is another type of approximation where function approximates outside given data points interval instead of inside as in interpolation. Like in interpolation, approximated function should go through the given data points.

In other words, extrapolation is an approximate determination of the function  $f(x)$  values in points  $x$ , which lies outside of the range  $[x_0, x_n]$ , by the values in points  $x_0, x_1, \dots, x_n$ .

### 3. Frame extrapolation using optical flow

In this work, we present the method, which can generate frame not between two consecutive frames, but after these two frames in respect of time. This task can be simply described as frame extrapolation. But this method can also be used as frame interpolation.

After solving the equation (4) we get motion vector  $(u, v)$  for each pixel. Now by multiplying some coefficient  $k$  and motion vector we can get following equation:

$$I_{new}(x_0 + ku, y_0 + kv, t_0 + k\Delta t) = I_0(x_0, y_0, t_0) \quad (5)$$

the  $k$  coefficient represents position of new frame in two cases:

- Frame Interpolation in case of  $k \in [0, 1]$ .
- Frame Extrapolation in case of  $k \notin [0, 1]$ .

### 4. Experiments

The main criteria to evaluate method is the speed criteria. As mentioned before, this is the comparison method performed with different optical flow calculation methods: Farneback, Brox and Duality based TV-L1 methods. As well as API to calculate them: CUDA and OpenCL.

Evaluation process includes image process: implementations work with image sequences with different resolutions. Evaluation provides with and without read/write operations speed. Main evaluation parameter is work speed. The quality provided by evaluation is considered as well, but from more subjective view point. Examples of generated frames are shown in figures 2-4.

Evaluation provided on three image sets with different resolutions: 2560×1600 pixels, 1182×664 pixels and 578×425 pixels. A computer with characteristics was used for the experiments:

- CPU – Intel Core i7-4770 3.40 GHz.
- RAM – 16 GB.
- GPU – MSI GeForce GTX 1070 with 8GB video memory.
- Hard drive – Kingston SSD 240GB.

Farneback method for optical flow calculation show the best result above mentioned and best suitable for this task. Duality based TV-L1 method show smooth and good quality optical flow, but calculation speed is very slow. Brox method not suitable at all, it shows worst result in both calculation speed and optical flow quality. Results obtained using the CUDA API presented in figures 5-7 and tables 1-2.



Figure 2. Generated frame by Farneback method.



Figure 3. Generated frame by TV-L1 method.



Figure 4. Generated frame by Brox method.

In terms of API as expected CUDA shows much better results than OpenCL. CUDA is more suitable for the task, but has small disadvantage and that is CUDA can work only on NVidia GPU accelerators, according to Steam about 75% of users prefer NVidia GPU's. In other hand OpenCL is a cross-platform API supported by all GPU's and CPU's, but the performance level is not good enough to consider it as an option. OpenCL implementations almost two times slower than CUDA implementation. This leads to the inability to generate frames in 60 FPS in real time even for low resolution images, thus the OpenCL implementation is insufficient for the task.

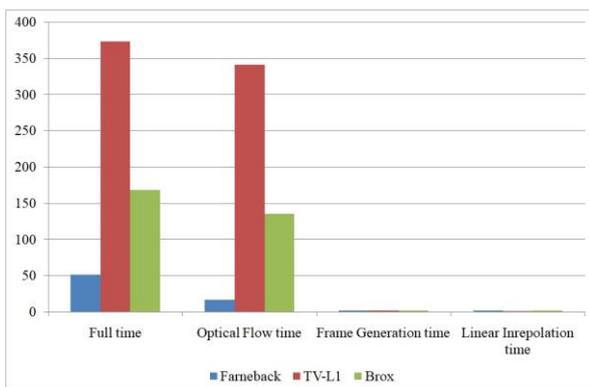


Figure 5. Work time of each method for low resolution.

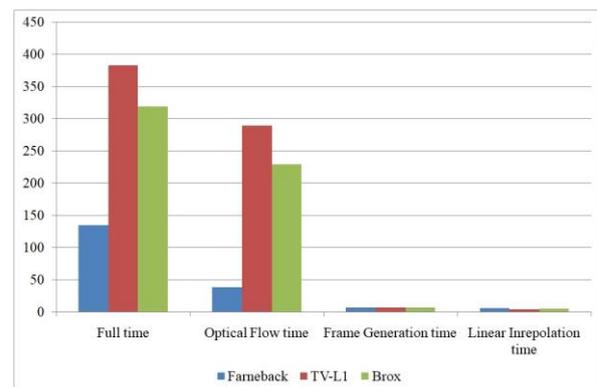
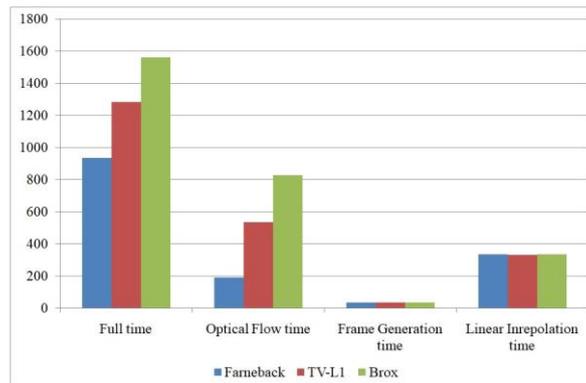


Figure 6. Work time of each method for medium resolution.



**Figure 7.** Work time of each method for high resolution.

**Table 1.** Work time for each resolution and each method in milliseconds.

Method	Resolution	Full time	Optical Flow time	Frame Generation time	Linear Interpolation time
Farneback	Low	51.03	17.14	1.84	1.92
	Medium	134.36	38.74	7.04	6.02
	High	935.31	189.55	36.47	335.89
TV-L1	Low	372.89	341.02	1.85	0.99
	Medium	382.67	288.88	7.31	4.36
	High	1284.47	533.75	36.31	331.45
Brox	Low	168.32	135.08	1.84	1.88
	Medium	319.41	228.98	7.37	5.32
	High	1559.35	827.25	36.81	336.85

**Table 2.** FPS for each resolution and each method.

Method	Resolution	FPS for Full time	FPS for Full time and Optical Flow Visualization time
Farneback	Low	19.61	40.16
	Medium	7.44	16.41
	High	1.07	1.64
TV-L1	Low	2.68	2.87
	Medium	2.61	3.23
	High	0.78	1.07
Brox	Low	5.94	7.02
	Medium	3.13	4.01
	High	0.64	0.81

## 5. Conclusion

In this work presented a new approach of frame generation based on optical flow. Proposed approach achieves frame interpolation based on optical flow information gained from current and previous frames. Analysed different optical flow methods in terms of speed and quality.

And show that this method is able to maintain performance in term of time to generated frames. However this performance only applicable to low resolution frames.

Further work may be conducted based on the results of this work. There are several ways which can improve this work and may give better results both in terms of calculation speed and quality.

Deep neural networks is one way to continue research on this topic. As shown in other works neural networks are good in image super resolution [8-10] as well as in image restoration [11]. By combining them new approach to frame interpolation may be presented. This approach may deliver good performance in both speed and quality.

## Acknowledgments

The work was prepared financial support in the course of the government contract 2.1724.2017/4.6.

## References

- [1] Bao W, Zhang X, Chen L and Gao Z 2017 High-quality and real-time frame interpolation on heterogeneous computing system *Proc. IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (Cagliari, Italy, 7-9 June 2017)* pp 1-4
- [2] Meyer S, Wang O, Zimmer H, Grosse M and Sorkine-Hornung A 2015 Phase-based frame interpolation for video *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Boston, USA, 7-12 June 2015)* pp 1410-18
- [3] Meyer S, Djelouah A, McWilliams B, Sorkine-Hornung A, Gross M and Schroers C 2018 Phasenet for video frame interpolation *Proc. Conference on Computer Vision and Pattern Recognition (Salt Lake City, USA, 18-23 June 2018)* pp 498-507
- [4] Farneback G 2003 Two-Frame Motion Estimation Based on Polynomial Expansion *SCIA 2003 (Lecture Notes in Computer Science vol 2749)* eds J Bigun and T Gustavsson (Berlin: Springer) pp 363-370
- [5] Brox T, Bruhn A, Papenbergh N and Weickert J 2004 High Accuracy Optical Flow Estimation Based on a Theory for Warping *Computer Vision - ECCV 2004 (Lecture Notes in Computer Science vol 3024)* eds T Pajdla and J Matas (Berlin: Springer) pp 25-36
- [6] Faisal M and Barron J 2007 High Accuracy Optical Flow Method Based on a Theory for Warping: Implementation and Qualitative/Quantitative Evaluation *ICIAIAR 2007 (Lecture Notes in Computer Science vol 4633)* eds M Kamel and A Campilho (Berlin: Springer) pp 513-525
- [7] Zach C, Pock T and Bischof H 2007 A Duality Based Approach for Realtime TV-L1 Optical Flow *DAGM 2007 (Lecture Notes in Computer Science vol 4713)* eds F A Hamprecht, C Schnorr and B Jahne (Berlin: Springer) pp 214-223
- [8] Dong C, Loy C C, He K and Tang X 2016 *IEEE Transactions on Pattern Analysis and Machine Intelligence* **38 (2)** 295-307
- [9] Ledig C, Theis L, Huszar F, Caballero J, Cunningham A, Acosta A, Aitken A, Tejani A, Totz, J, Wang Z and Shi W 2017 Photo-realistic single image super-resolution using a generative adversarial network *Proc. 30th IEEE Conference on Computer Vision and Pattern Recognition (Honolulu, United States, 21-26 July 2017)* pp 105-114
- [10] Lim B, Son S, Kim H, Nah S, Lee K M 2017 Enhanced Deep Residual Networks for Single Image Super-Resolution *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (Honolulu, United States, 21-26 July 2017)* pp 1132-40
- [11] Gao R and Grauman K 2017 On-demand Learning for Deep Image Restoration *Proc. IEEE International Conference on Computer Vision (Venice, Italy, 22-29 October 2017)* pp 1086-1095