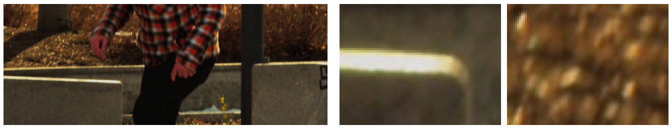
Phase-Based Frame Interpolation for Video

Simone Meyer[1], Oliver Wang[2], Henning Zimmer[2], Max Grosse[2], Alexander Sorkine-Hornung[2]
[1]ETH Zurich. [2]Disney Research Zurich.


(a) One of the input images [1] (b) Optical flow (global) [2]

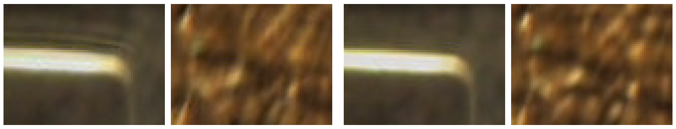

(c) Phase-based [3] (d) Our phase-based method

Figure 1: Interpolated frames using optical flow (b), the method of [3] (c) and our approach (d). Our approach produces better results than the method of [3] and similar quality results to optical flow, while being faster and simpler. (© Tom Guilmette [1])

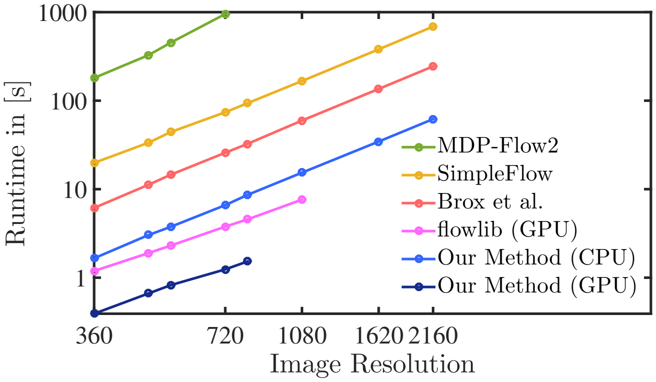


Figure 2: Log-log plot of running time versus (vertical) image resolution when interpolating one image. Our method scales favorably compared to flow-based methods.

Computing interpolated, in-between images is a classic problem in image and video processing, and is a necessary step in numerous applications such as frame rate conversion (e.g. between broadcast standards), temporal upsampling for generating slow motion video, image morphing, as well as virtual view synthesis. Traditional solutions to image interpolation first compute correspondences, followed by image warping. Due to inherent ambiguities in computing such correspondences, most methods are heavily dependent on computationally expensive global optimization. With today's trend in the movie and broadcasting industry to higher resolution and higher frame rate video, there is the need for interpolation techniques that can deal efficiently with this considerably larger volume of data.

Recently, phase-based methods have shown promise in applications such as motion and view extrapolation [3, 6]. These methods rely on the assumption that small motions can be encoded in the phase shift of an individual pixel's color. Currently, however, the spatial displacement which can be encoded in the phase information with these methods is highly limited, which narrows their application to image interpolation, visualized in Figure 1c.

To overcome this issue, we propose a method that propagates phase information across oriented multi-scale pyramid levels using a novel bounded shift correction strategy. Similar to prior work, we compute phase values on multiple frequency bands using a complex-valued steerable pyramid [5]. Our algorithm estimates and adjusts the phase shift information for each pixel using a coarse-to-fine approach, with the assumption that high frequency content moves in a similar way to lower frequency content. The shift correction is performed based on a newly formulated confidence estimate which improves upon simple strategies used in prior work [3]. Additionally we propose an adaptive upper bound on the phase shift that effectively avoids artifacts for large motions, and an extension to phase-based image synthesis that leads to smoother transitions between interpolated images. In combination, these extensions considerably increase the amount of displacement that can be successfully represented and interpolated, see Figure 1d.

Based on these extensions, we describe an efficient framework to synthesize in-between images, which is simple to implement and parallelize. As a result our implementation allows to interpolate a HD frame in a few seconds on the CPU and in around one second on a GPU. Importantly, the computation time scales linearly with image size, which makes this method a viable solution for frame interpolation and retiming of high resolution and high frame rate video, see Figure 2.

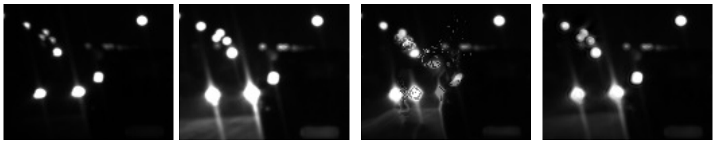

(a) Input images [4] (b) Optical flow (global) [2] (c) Our phase-based method

Figure 3: Optical flow can fail in the presence of strong lighting changes. Our phase based method does not suffer from these artifacts.

While our approach cannot handle large motion of high frequency content, we show that it performs similarly on a number of benchmarks to traditional optical flow methods with real world footage, and can even yield better results in cases where the fundamental assumptions of optical flow (such as the brightness constancy assumption) break down. Rather than creating distorted images, phase-based methods default to linear blending, which can sometimes be preferable. An example of this is shown in Figure 3.

In conclusion, we present a novel method for frame interpolation, that works directly on pixel values and requires a fraction of the time compared to typical flow-based methods, while having comparable memory requirements and obtaining visually similar results.

[1] http://www.tomguilmette.com/archives/593.

[2] Thomas Brox, Andrés Bruhn, Nils Papenberg, and Joachim Weickert. High accuracy optical flow estimation based on a theory for warping. In *ECCV*, pages 25–36, 2004. doi: 10.1007/978-3-540-24673-2_3.

[3] Piotr Didyk, Pitchaya Sitthi-amorn, William T. Freeman, Frédo Durand, and Wojciech Matusik. Joint view expansion and filtering for automultiscopic 3D displays. *ACM Trans. Graph.*, 32(6):221, 2013. doi: 10.1145/2508363.2508376.

[4] S. Meister, B. Jähne, and D. Kondermann. Outdoor stereo camera system for the generation of real-world benchmark data sets. *Optical Engineering*, 51(02): 021107, 2012.

[5] Javier Portilla and Eero P. Simoncelli. A parametric texture model based on joint statistics of complex wavelet coefficients. *IJCV*, 40(1):49–70, 2000. doi: 10.1023/A:1026553619983.

[6] Neal Wadhwa, Michael Rubinstein, Frédo Durand, and William T. Freeman. Phase-based video motion processing. *ACM Trans. Graph.*, 32(4):80, 2013. doi: 10.1145/2461912.2461966.

This is an extended abstract. The full paper is available at the Computer Vision Foundation webpage.