A Phase Difference Estimation Method for Video Frame Interpolation

Qian Yuan, Bo Huang

College of Information Science and Technology, Jinan University, Guangzhou 510632, China.

562591407@qq.com

Abstract

Video interpolation is a classic problem in image and video processing. There is a growing demand for higher frame video over the past decades. High frame video has a wide range of applications in different fields, such as, biomedical, remote sensing imaging, archaeological research, etc. Video interpolation is a technique that uses an correlation between two consecutive frames to obtain interpolated frames, and then inserts interpolated frames between the original two frames to achieve an higher frame rate video. In this paper, a phase-based video interpolation algorithm is designed by using the frequency domain characteristics of images. The algorithm of this paper can estimate the phase difference between different spatial phases of two frames. The method can solve the shortcomings of traditional algorithms running slowly and poorly, and greatly improves the practicability of the algorithm.

Keywords

Video interpolation algorithm, Frame rate, Steerable pyramid, Phase, Image Similarity.

1. Introduction

With the rapid development of information technology, video as a carrier for capturing, processing, recording, storing and transmitting information has long been an indispensable part of human life. Video is the use of the persistence effect of human vision. When images are played at a certain rate, these images appear to be smooth and continuous motion pictures. In most cases, a frame rate of video more than 24 frames per second is required. In recent years, with the rapid development of mobile Internet, low-frame video has not been able to meet the growing sensory experience and cultural needs of people, so the demand for high frame rate video technology is increasing. The so-called high frame rate video need to increase the normal 24 frames video to 120 frames or even 1000 frames. High frame rate leads to challenges in processing the video. Many applications in video processing need to synthesize new images based on the existing frames.

At present, according to the existing literature, video interpolation techniques can be divided into two categories, one is called Lagrangian method, and the other is called Euler method.

The Lagrangian method model motion as spatial pixel displacement [1-3]. The most popular Lagrangian method is the optical flow method. The main research is the classic Horn-Schunck optical flow method and Lucas-Kanade optical flow method. The optical flow algorithm is to calculate the optical flow of two adjacent frames, that is, the instantaneous change rate of grayscale between each pixel point. Horn-Schunck optical flow method needs to calculate the optical flow value for each pixel point, and Lucas-Kanade optical flow method only needs to calculate the optical flow of several points. Since most optical flow methods rely on global optimization to solve the matching problem, they will run slowly and very difficult to implement. There have been some parallel flow implementations on GPUs which allow to significantly reduce running times, but the size of the image is limited by GPU memory.

Eulerian methods consider the change of color per pixel over time [4]. Eulerian method is usually able to simulate a larger amount of motion in a video in a simple and effective way. It can be said that the simplest Euler method is the frame averaging. Since the motion of the pixel is not considered, the ghosting of the high-frequency content is unavoidable.

The method of this paper belongs to the Euler method. In this paper, two input images are converted from the spatial domain to the frequency domain using a steerable pyramid, and the interpolated image is generated by the phase of the spatial frequency domain. The processing of ghosting is mainly improved by adjusting the parameters of the frequency band. The closest method to this paper is the method of Didyk et al. that proposes a phase-based approach to extrapolate a pair of images to multi-view content for 3D displays [5]. Our method has made some improvements based on the method of Didyk et al.

2. Method

The phase-based method for video interpolate algorithm is divided into four parts: Steerable pyramid, phase difference correct, interpolate, and reconstruction. The flow diagram is shown in Fig 1.

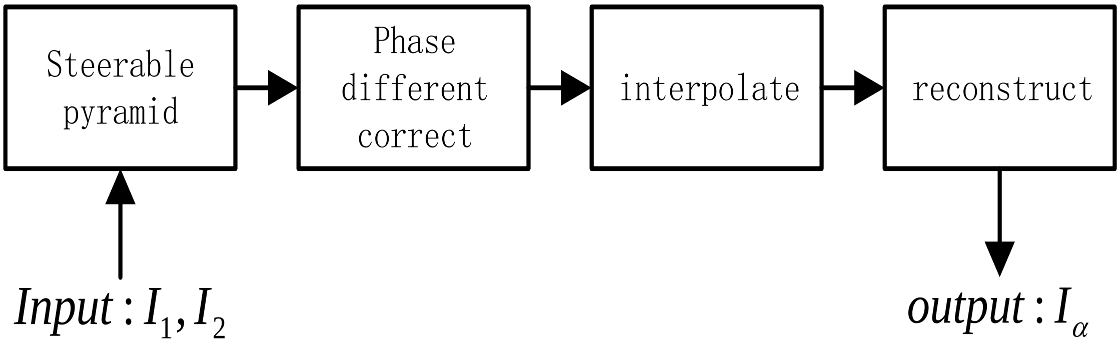


Fig.1 Algorithm flow diagram

2.1 Steerable Pyramid

In order to effectively obtain the information in the image, a lot of research has been done. In 1991, Freeman proposed a steerable pyramid [6,7]. The steerable pyramid is a multi-scale and multi-directional image decomposition method, which can independently decompose images in different scales and direction.

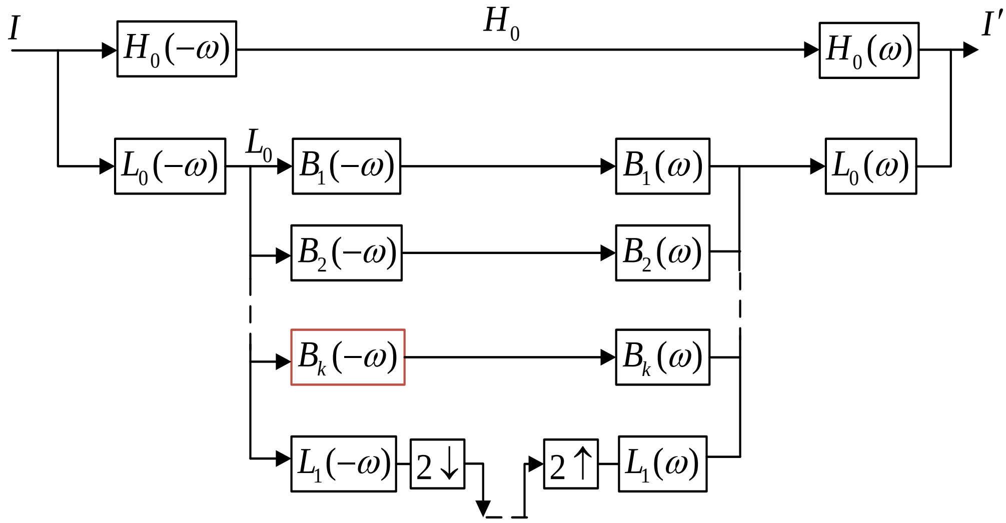


Fig.2 System diagram for the steerable pyramid

The steerable pyramid diagram is shown in Fig 2, which is a recursive decomposition process, $H_0(\omega)$ is the high-pass filter, $L_0(\omega)$ is the low-pass filter, and $B_k(\omega)$ is the band-pass filter. The reconstruction process of the steerable pyramid is in the right half of Fig 2.This reconstruction process is simple and is actually similar to the decomposition process.

2.2 Phase Different Correction

Before the interpolation, the phase difference ϕ_{diff} of the input images I_1, I_2 needs to be calculated. The phase difference is calculated as:

$$\phi_{diff} = a\tan 2(\sin(\phi_1 - \phi_2), \cos(\phi_1 - \phi_2)) \tag{1}$$

Large displacement with phase difference greater than π will result in phase ambiguity. Because of the periodicity of the phase value, the phase difference is only defined between [−π, π].

In order to reduce some of the blur caused by the incorrect phase of the inserted images, the phase difference need to limit to a constant limit ϕ_{limit}. If the phase difference is above this limit, i.e. $\left|\phi_{diff}^{l}\right| > \phi_{limit}$,the phase value from the next coarser level is used as the corrected phase difference: $\tilde{\phi}_{diff}^{l} = \lambda\phi_{limit}^{l+1}$, we define ϕ_{diff} depending on the current level l, the total number of levels L and the scale factor λ as [8]:

$$\phi_{limit} = \tau\pi\lambda^{L-l} \tag{2}$$

The parameter τ determines the percentage of the limit, and the τ value needs to be artificially set. It is often necessary to judge according to the situation of different pictures. This paper proposes a method of adaptively adjusting τ that can reduce phase ambiguity as described in Section 3.4.

2.3 Phase Interpolation

Due to the shift correction, $\phi_1 + \phi_{diff}$ is no longer guaranteed to match ϕ_2. It is necessary to maintain the origin phase difference for calculation, and also to consider the corrected phase difference [9]. Assuming the original phase difference is ϕ_{diff}, the corrected phase difference is $\tilde{\phi}_{diff}$. The algorithm uses the periodicity of the phase to match ϕ_{diff}, searches for a phase difference $\hat{\phi}_{diff}$ that is $2\pi\gamma^*$ the original phase difference ϕ_{diff} from Equation (1) while being as close as possible to $\tilde{\phi}_{diff}$.

$$\hat{\phi}_{diff} = \phi_{diff} + 2\pi\gamma^* \tag{3}$$

Where γ^* is determined as:

$$\gamma^* = \arg\min_{\gamma} \left\{ (\tilde{\phi}_{diff} - (\phi_{diff} + \gamma 2\pi))^2 \right\}$$

Due to adjustment we can now compute the phase ϕ_α as:

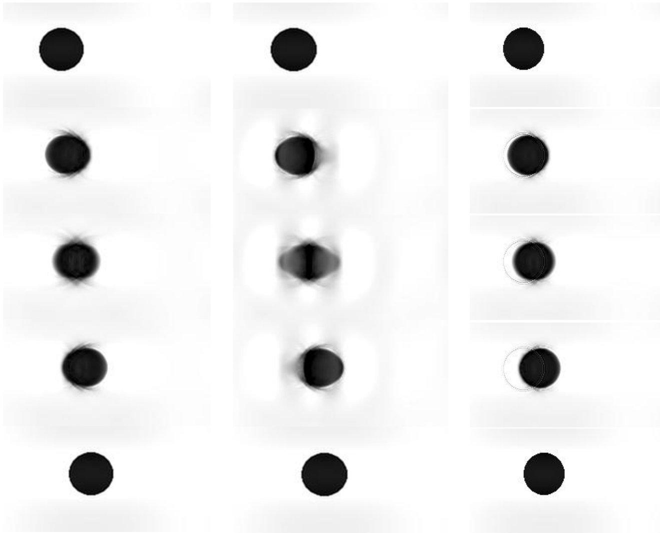
$$\phi_\alpha = \phi_1 + \alpha\hat{\phi}_{diff} \tag{4}$$

Constructing an interpolated image requires not only interpolation of the phase, but also interpolation of the amplitude and low-pass residuals. In order to achieve a smoother transition, we recommend linearly blending the amplitude as well as the low frequency residual.

2.4 Improvement based on similarity

Aiming at the problem that the parameter τ proposed in Section 3.2 needs to be adjusted under different circumstances, this paper proposes a method based on the similarity of two images to adaptively adjust the parameter τ. This method can effectively remove some of the blur caused by improper adjustment of parameter τ.

As shown in Fig 3, the input image is a circle at two different locations. The interpolated images changes as the correction parameter τ changes. Fig 3(a) is the result of the interpolated image when $\tau = 0.83$, and Fig 3(b) is the result of the interpolated image when $\tau = 0.5$. It can be seen from the figure that the result obtained when $\tau = 0.83$ is better, and the result when $\tau = 0.5$ has a significant ghosting. Fig 3(c) is the result of the interpolation when two circles are closer, in this case the parameter $\tau = 0.62$.



(a) $\tau = 0.83$ (b) $\tau = 0.50$ (c) Two circles are closer

Fig.3 Different similarity comparison

Based on the above analysis, our method defines τ as the similarity between images and uses the similarity to improve the phase difference. A schematic diagram of similarity is shown in Fig 4. When the two circles change relatively large, set τ to be larger. When the two circle change relatively small, set τ to be smaller. The similarity calculation uses the difference hash algorithm. τ ranges from 0 to 1. When $\tau = 0$, the two images are identical. When $\tau = 1$, the two images are completely different.

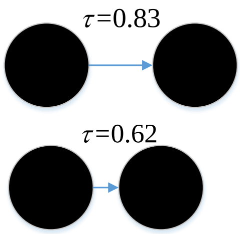


Fig.4 Schematic diagram of similarity

Since adjusting of similarity is mainly for the part of motion in image, we divide the method into two steps: first, to find the region with the largest motion in the video, and secondly to calculate the similarity of the region. The example in Fig 5 shows the step of similarity, using the Middlebury website on Backyard dataset [10]. After finding the region with the largest motion in the image (the white rectangular block in the figure), calculating the similarity τ of this region, so that adaptively adjusting ϕ_{diff} . The method of similarity improve the phase difference can adjust the image adaptively and reduce the ghosting of the interpolated image.

(a)Our interpolation (b) Before improvement (c)Our method

Fig.5 Comparison with similarity improvement from Backyard dataset

3. Results

3.1 Effect comparison

There are four main parameters that can be adjusted in this algorithm, namely *Twidth*, *Levels*, *Scale*, and *nOrientations* [11].Adjust the above four parameters, and finally get the optimal parameter values: $Twidth = 1$, $Levels = 23$, $Scale = 1.2$, $nOrientations = 8$ .The result of the interpolated image as Fig 6.

(a)Input image1 (b)Output image (c)Input image2

Fig.6 The result of the interpolated image

Fig 6 compares the our method with the LK optical flow method [12,13]. The input is two images with obvious color changes, which shows that the phase-based interpolation method has achieved good results. The interpolated image based on the optical flow method not only has a long running time, but also has an obvious ghost image and poor effect. When there is a change in this image, the method of this paper is better than the optical flow method.

(a)Input image

(b)Our phase-based method

(c)Optical flow method

Fig.7 : Interpolated frames using our method and optical flow

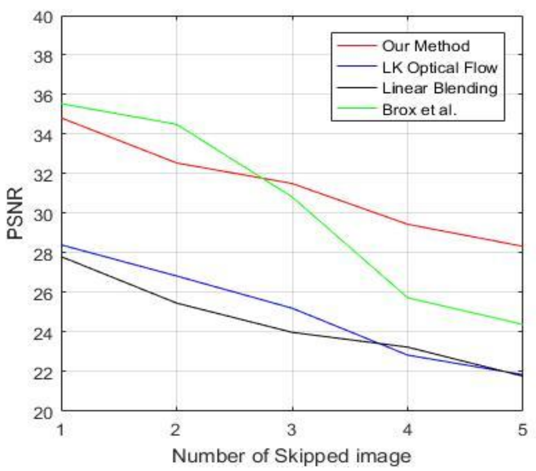


Fig.8 PSNR for different methods.

We computed error measurement for different Lagrangian, flow-based methods, as well as Eulerian methods. As error measurements we use Peak Signal to Noise Ratio(PSNR). The plot in Fig 7 visualizes the PSNR when leaving out an increasing number of intermediate images (causing larger displacements). Generally, as the PSNR value increases, the result is better. Fig 7 show that our phase-based method performs better than others approaches.

3.2 Running times

In Table 1 we report running times for interpolating between a pair of 640x480 images, all measured on a standard desktop PC (Intel Core i5 3.20 GHz), showing the runtime of image processing. Comparing the running times in Table 1, it is clear that the algorithm in this paper is significantly faster than any optical flow based method [14,15].

Table 1 Running times of different algorithms

Method	MDP-Flow	Brox et al.	LK optical flow	Our method
Running time	152.41s	50.65s	43.45s	28.09s

4. Conclusion

This paper proposes a new phase-based frame interpolation method. The results obtained by the algorithm are compared with the most advanced optical flow-based methods, and the algorithm is applied to have better visual quality on many real data sets. For strong illumination changes, the algorithm shows its better performance. Important advantages of our method are fewer parameters that can be adjusted. Most importantly, the method is computationally efficient. All of these advantages make this algorithm a useful tool for video frame interpolation.

References

[1] D. Sun, S. Roth, and M. J. Black: Secrets of optical flow estimation and their principles, In CVPR, 2010, p. 2432–2439.

[2] S. Baker, D. Scharstein, J. P. Lewis, S. Roth, M. J. Black, and R. Szeliski: A database and evaluation methodology for optical flow. IJCV, 2011,92(1):1–31.

[3] M. W. Tao, J. Bai, P. Kohli, and S. Paris: Simpleflow: A noniterative, sublinear optical flow algorithm, Comput. Graph. Forum, 2012,31(2):345–353.

[4] P. J. Burt and E. H. Adelson: A multiresolution spline with application to image mosaics. ACM Trans. Graph., 1983,2(4):217–236.

[5] P. Didyk, P. Sitthi-amorn, W. T. Freeman, F. Durand, and W. Matusik: Joint view expansion and filtering for automultiscopic 3D displays, ACM Trans. Graph, 2013, 32(6):221-228.

[6] E. P. Simoncelli and W. T. Freeman: The steerable pyramid: a flexible architecture for multi-scale derivative computation. In International Conference on Image Processing, IEEE, 1995, p. 444–447.

[7] E. P. Simoncelli, W. T. Freeman, E. H. Adelson, and D. J. Heeger: Shiftable multiscale transforms. IEEE Transactions on Information Theory, 1992, 38(2):587–607.

[8] Meyer. S, Wang. O, Zimmer, H. Grosse and M. Sorkine-Hornung: A Phase-based frame interpolation for video, CVPR, 2015, p. 1410–1418.

[9] K. Sunkavalli, M. K. Johnson, W. Matusik, and H. Pfister: Multi-scale image harmonization, ACM Trans. Graph, 2010 , 29(4):40-46.

[10] http://vision.middlebury.edu/flow/data/

[11] J. Portilla and E. P. Simoncelli. A parametric texture model based on joint statistics of complex wavelet coefficients. IJCV, 40(1):49–70, 2000.

[12] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert: High accuracy optical flow estimation based on a theory for warping, ECCV, 2004, p.25–36.

[13] L. Xu, J. Jia, and Y. Matsushita: Motion detail preserving optical flow estimation, IEEE Trans. Pattern Anal. Mach. Intell, 2012, 34(9):1744–1757.

[14] C. Zach, T. Pock, and H. Bischof: A duality based approach for realtime TV-L 1 optical flow. DAGM, 2007, p.214–223.

[15] Li Xu, Jiaya Jia, Yasuyuki Matsushita: Motion Detail Preserving Optical Flow Estimation , CVPR, 2010, p.1744 – 1757.