Friedrich-Alexander-Universität Erlangen-Nürnberg

Chair of Multimedia Communications and Signal Processing

Prof. Dr.-Ing. André Kaup

Internship Report

Motion Compensated Framerate Up-Conversion Using Frequency Selective Extrapolation

Junyue Wu

08.15.2015

Supervisor: Dipl.-Ing. Michel Bätz

Contents

1. Introduction

These years video cameras have increased very rapidly. They are applied to kinds of portable devices like smartphones or tablet PCs. However the resolution of the videos taken by these devices is always not high enough for the entertainment in the daily life. As a result, how to reconstruct videos with higher Framerates from low resolution videos is a desirable problem. With higher Framerates the video can be shown fluently and also used for monitoring systems, for example 360^{o} cameras. This kind of cameras must process a large amount of data, so videos with low Framerates are necessary. In order to get better result methods with or without motion compensation can be used for increasing the Framerates of the video. [1]

During this internship only temporal resolution enhancement is to be dealt with. At first the method with motion compensation is analyzed. After motion compensation, FSE which is short for frequency selective extrapolation, is applied to fill the holes in the interpolated image. Several parameters of the FSE algorithm are analyzed and then the more appropriate values are selected in order to get better interpolated image.

2. Motion-Compensated FRUC using FSE

There are two main steps to increase the Framerate of original video. At first motion estimation is applied to get an interpolated frame between two original frames. Then FSE is used for optimization, that is to say, the holes caused by motion compensation can be filled.

2.1 Motion Estimation

Video sequences can be seen as a series of individual frames. During a certain time interval, more frames mean higher resolution. Motion vectors are used to describe the transformation from one frame, which is usually 2D image, to another. The process of determining the motion vectors is known as motion estimation.

2.1.1 Motion Vectors

In this internship motion vectors are block-based. One of the advantages of block-based motion estimation is its low complexity. However block-based estimation assumes that the objects of a scene are in translatory motion, that is to say, no rotation, deformation, camera zoom or luminance alteration in a scene.

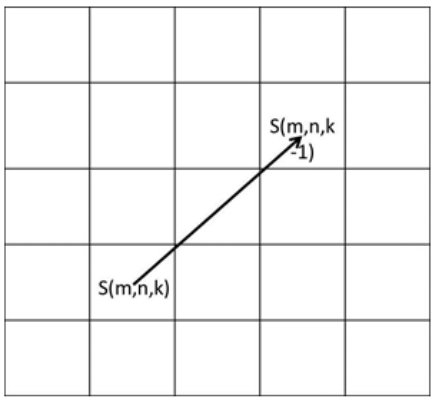


Figure 2.1 reference frame S(m,n,k-1)

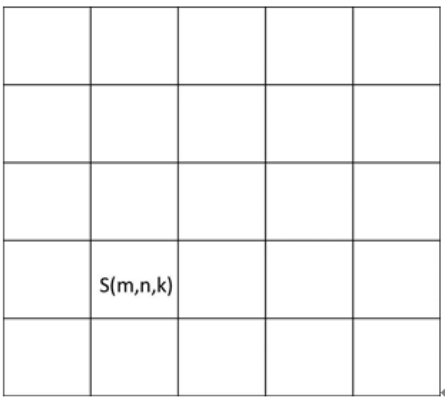


Figure 2.2 current frame S(m,n,k)

Figure 2.1 and 2.2 show the procedure of determining motion vectors. Block-based estimation first divides every frame into blocks, here there are 5×5 blocks in each frame. The block S(m,n,k) with the spatial coordinates m and n and the time index k is the block whose motion vector is to be determined. The previous frame at time k-1 is dealt as the reference frame. The method of determining motion vector is to find the block in reference frame which is the most similar to the block in current frame. As shown in Figure 2.1, S(m,n,k-1) is the block which we find, in other words, block S(m,n,k-1) and block S(m,n,k) have the greatest possible overlap. The arrow shown in Figure 2.1 describes the motion vector of block S(m,n,k). The direction of motion vector is from the current frame to the reference frame.

The method of searching for the best matching block can be described in the following formula, which is called the minimization of the sum of absolute differences (SAD):

$$\mathrm{SAD}(d_m, d_n) = \sum_{(m,n)\in S} |S(m,n,k) - S(m+d_m, n+d_n, k-1)| \tag{2.1}$$

When the block with the smallest SAD is chosen, $\hat{d} = (\widehat{d_m}, \widehat{d_n})$ is motion vector.

$$\hat{\mathrm{d}} = \left(\widehat{d_m}, \widehat{d_n}\right) = \arg\min_{(d_m,d_n)} \mathrm{SAD}(d_m, d_n) \tag{2.2}$$

When we are searching for the most matching block, the number of surrounding blocks to be considered with is determined by the search range. The search range limits the search to a reasonable range. For example, when the search range is R, the blocks surrounding the current block with R pixels to the left and right and R pixels above and below will be dealt with.

The accuracy of motion estimation can be defined by the step size, which determines how far the next step of searching goes from the current block. The simplest is full-pixel accuracy, that means the block shifts pixel-wise every step. In practice, Sub-pixel accuracy is also be applied.

2.1.2 Motion Compensation

In this internship motion vectors are used for temporal block-based prediction. At first an arbitrary frame of a video is taken as the reference frame, and the next frame is taken as the current frame. By block-based motion estimation we can get the motion vectors $\widehat{\mathrm{d}_1}$ from the current blocks to the reference blocks. Then half of the motion vectors can be seen as new motion vectors $\widehat{\mathrm{d}_2}$. With the current frame and $\widehat{\mathrm{d}_2}$ a new frame can be interpolated between the two original frames. In this way the number of the frames is increased, which achieves a

video sequence with higher resolution. Figure 2.3 shows the procedure of motion compensation. The black arrow is the motion vector from S(m,n,k) to S(m,n,k-1). The red arrow is half of the black arrow. S(m,n,k-0.5) is the compensated block which is not exited in the original video sequence. In this way the resolution of this video sequence can be increased. In fact, the number of compensated frames between two existing frames is not fixed. If the time interval is 0.25, we can get three compensated frames S(m,n,k-0.25), S(m,n,k-0.5), S(m,n,k-0.75). However, more compensated frames between two existing frames may lead to lower accuracy. In this internship the time interval is 0.5, that means I compensate only one additional frame between two existing frames.

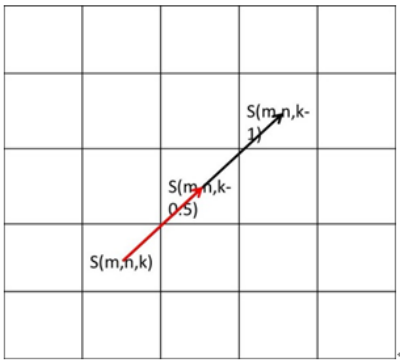


Figure 2.3 compensated frame S(m,n,k-0.5)

Here the file “man_in_restaurant_960×448_1440frames.yuv” is taken as an example. Here, in order to evaluate the algorithm, I take frame 568 as the reference frame and frame 570 as the current frame. The new compensated frame between frame 568 and 570 can be compared to the frame 569. The most ideal situation is that, the new frame is the same as frame 569.

Figure 2.4 shows the original frame 569 and figure 2.5 shows the compensated frame. It can be observed clearly that several pixels are empty in the compensated frame. The reason of these holes is that the interpolated frame is based on the motion vectors, which cannot be estimated very accurately as there may be several alike blocks around a certain block. When blocks in the current frame are moved to interpolate the new frame due to the motion vectors, some blocks will overlap, which leads to the holes. Here, block size is 8 and search range is 32.

Figure 2.4 original frame 569

Figure 2.5 compensated frame

2.2 FSE Extrapolation

The task of extending a signal from known blocks into unknown blocks, that is to say, filling the holes in the estimated frame is very important in image and video signal processing [2] [3] [4]. Here, FSE extrapolation is used, which is short for Frequency Selective Extrapolation. With FSE a high extrapolation quality can be achieved. The original FSE is real-valued frequency selective extrapolation which has relatively high computational complexity. When a complex-valued model of FSE is used for real-valued signals, it will be less complex. [4]

2.2.1 Real-Valued FSE

Because the compensated frame shown in the last section loses some blocks and the goal of FSE is to fill the green loss blocks which are considered as spatial extrapolation, here we only deal with two-dimensional data sets. As shown in Figure 2.6 [4], area B represents the green loss blocks and area A which is called support area represents the known blocks surrounding the loss blocks. These two areas are regarded together as the extrapolation area L with the coordinates m and n and the size is M×N.

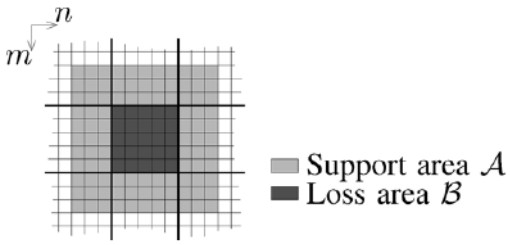


Figure 2.6 extrapolation area L [4]

The following function [4] is what the real-valued model of FSE is based on.

$$\varphi_{(k,l)}[\mathrm{m},\mathrm{n}] = e^{j(2\pi/M)km} e^{j(2\pi/N)ln} \tag{2.3}$$

For the extrapolation, the following model [4] is generated as weighted superposition of basis functions. $\hat{c}_{(k,l)}$ and $\hat{c}^*_{(k,l)}$ are the weighting factors.

$$\mathrm{g}[\mathrm{m},\mathrm{n}] = \frac{1}{2}\sum_{(k,l)\in\mathcal{R}}(\hat{c}_{(k,l)}\varphi^*_{(k,l)}[m,n] + \hat{c}^*_{(k,l)}\varphi_{(k,l)}[m,n]) \tag{2.4}$$

For the extrapolation area L shown in Figure 2.6, the weighting function is defined by following function [4].

$$\mathrm{w}[\mathrm{m},\mathrm{n}] = \begin{cases} \hat{p}^{\sqrt{(m-(M-1)/2)^2+(n-(N-1)/2)^2}}, & \text{for } (\mathrm{m},\mathrm{n}) \in \mathrm{A} \\ 0, & \text{for } (\mathrm{m},\mathrm{n}) \in \mathrm{B} \end{cases} \tag{2.5}$$

This function is applied for controlling the influence of every block. The area B is marked with 0. The block far away from B has a small weight value and therefore low influence on the area B. This process is carried out in frequency domain. At first the input signals are transformed into the frequency domain and at last the final model is transformed back into the spatial domain. In order to make the transforms efficient, the Fast Fourier Transform (FFT) is applied. [4]

One of the reasons why real-valued FSE has high computational complexity is that real-valued FSE generates a real-valued model from complex-valued basis functions. In order to reduce the computational complexity, complex-valued FSE is applied. [1]

2.2.2 Complex-Valued FSE

To generate a complex-valued model instead of a real-valued model can reduce the computational complexity. The following model [4] is generated by complex-valued FSE.

$$g(m,n) = \sum_{(k,l)\in\mathcal{R}} \hat{c}_{(k,l)} \varphi_{(k,l)}[m,n] \tag{2.6}$$

Complex-valued FSE is also applied in the frequency domain. The weighting function (2.5) is also applied here to control the influence of every block on the model. The coefficients are defined by [4]:

$$p_{(k,l)}^{(v)} = \frac{\sum_{(m,n)\in L} r^{(v-1)}[m,n]\varphi^{*}_{(k,l)}[m,n]w[m,n]}{\sum_{(m,n)\in L} \varphi^{*}{}_{(k,l)}[m,n]w[m,n]\varphi_{(k,l)}[m,n]} \tag{2.7}$$

The following figure from [4] shows the process of complex-valued FSE. Similar to real-valued FSE, one transform into the frequency domain at first and one back into the spatial domain after the iterations are required. Complex-valued FSE is less complex than real-valued FSE and therefore it can be carried out fast. [4]

Alg. 1 Complex-valued Frequency Selective Extrapolation

input: distorted signal $s[m,n]$, weighting function $w[m,n]$
  /* Transform input signals into Fourier domain */
  $R_w[k,l] = \mathrm{FFT}\{s[m,n]\, w[m,n]\}$
  $W[k,l] = \mathrm{FFT}\{w[m,n]\}$
  $\bar{W}_0 = \frac{1}{W[0,0]}$
  for all $\nu = 1, \ldots, I$ **do**
    /* Basis function selection */
    $(u,v) = \mathrm{argmax}_{(k,l)} |R_w[k,l]|^2$
    /* Expansion coefficient estimation */
    $\hat{c} = \gamma R_w[u,v]\, \bar{W}_0$
    /* Model update */
    $G[u,v] = G[u,v] + MN\hat{c}$
    /* Residual update */
    for all $k = 0, \ldots, M-1 \wedge l = 0, \ldots, N-1$ **do**
      $R_w[k,l] = R_w[k,l] - \hat{c}W[k-u, l-v]$
    end for
  end for
  /* Retransform model into spatial domain*/
  $g[m,n] = \mathrm{IFFT}\{G[k,l]\}$
  /* Replace distorted signal parts */
  for all $(m,n) \in \mathcal{B}$ **do**
    $s[m,n] = \mathrm{Re}\{g[m,n]\}$
  end for
output: extrapolated signal $s[m,n]$

Figure 2.7 algorithm of complex-valued FSE [4]

3. Results

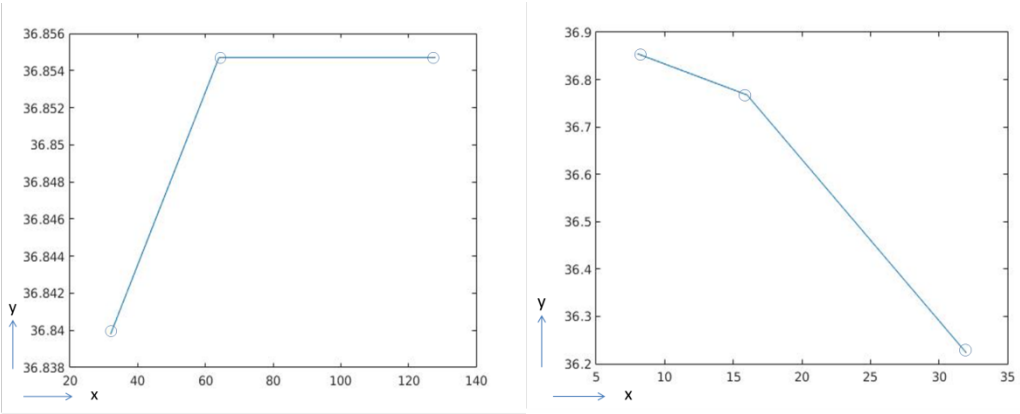
In this section the final results will be shown. There are several parameters when applying FSE, for example FFT size, block size and border width. In this internship these parameters are tested in order to determine the most appropriate value. Then we can achieve the highest extrapolation quality.

Here, PSNR (peak signal to noise ratio) is used to evaluate the quality of extrapolation. Higher PSNR means better extrapolation quality.

$$\text{PSNR} = 10\ log_{10}\left(\frac{\sum_{x=1}^{X}\sum_{y=1}^{Y}\hat{S}^2}{\sum_{x=1}^{X}\sum_{y=1}^{Y}\left(V(x,y)-U(x,y)\right)^2}\right)dB \tag{3.1}$$

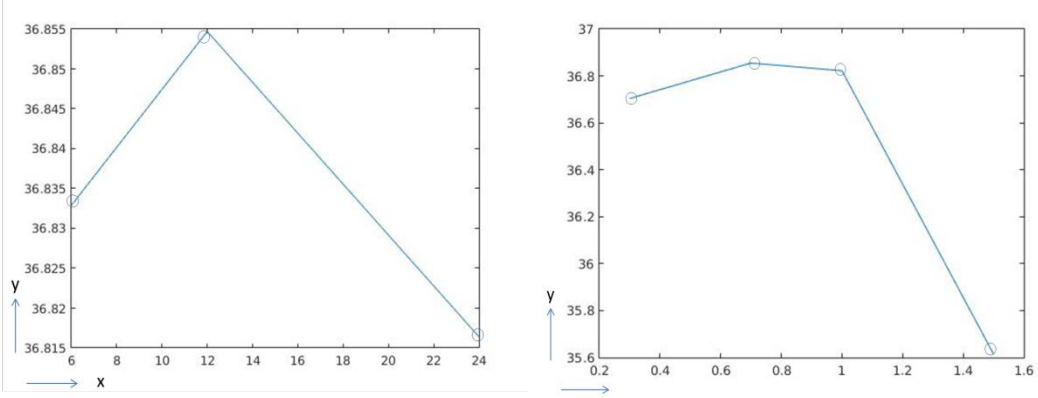
The original image is denoted by U(x,y) and the compensated one by V(x,y). $\hat{S}$ is the maximum possible power of the image and used for normalization.

The file “man_in_restaurant_960×448_1440frames.yuv” is taken as an example. The original frame 568 and 570 are taken as the input signals. With different parameters I get several compensated frame 569. Then the compensated frame 569 is compared to the original frame 569. The following figures show the relation between the parameters and PSNR.



y: PSNR x: FFT size y: PSNR x: block size

Figure 3.1 relation between parameters and PSNR (1)



y: PSNR x: border width y: PSNR x: rho

Figure 3.2 relation between parameters and PSNR (2)

It is obvious that the best extrapolation quality can be achieved when FFT size is 64, block size is 8, border width is 12 and rho is 0.7. Two other parameters odc_factor and conc_weight can also affect the quality. When odc_factor is 0.5 and conc_weight is 1 the extrapolation quality is the highest.

Figure 3.3 shows the final extrapolated frame 569 with the parameters determined in the last step.

Figure 3.3 frame 569 before and after FSE extrapolation

Conclusion

With motion estimation and FSE extrapolation a new frame between two original frames can be interpolated. Then the total number of the frames is increased, which means higher resolution. This process is considered as temporal resolution enhancement. Motion estimation also affects the extrapolation quality. For a certain video sequence search range and block size are two important parameters of motion estimation. The correct motion vectors cannot be accurately determined when search range is too large or too small. The task of determining an appropriate search range is very necessary.

References

[1] A. Kaup, K. Meisinger, and T. Aach, "Frequency selective signal extrapolation with applications to error concealment in image communication," Int. J. Electron. Commun. (AEU), vol. 59,pp. 147-156, June 2005.

[2] J. Seiler and A. Kaup, “Fast orthogonality deficiency compensation for improved frequency selective image extrapolation,” in Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2008), Las Vegas, NV, USA, Mar. 30–Apr. 4, 2008, pp. 781– 784. 272.

[3] J. Seiler and A. Kaup, “Motion compensated frequency selective extrapolation for error concealment in video coding,” in Proc. European Signal Processing Conference (EUSIPCO), Lausanne, Switzerland, 25.-29. Aug. 2008.

[4] J. Seiler and A. Kaup, “Complex-valued frequency selective extrapolation for fast image and video signal extrapolation,” IEEE Signal Processing Letters, vol. 17, pp. 949–952, 2010.