Ph.D. Dissertation of Engineering

Multiple Vectors based MEMC and a Deep CNN for Video Frame Interpolation

비디오 프레임 보간을 위한 다중 벡터 기반의 MEMC 및 심층 CNN

February 2019

Graduate School of
Seoul National University
Electrical and Computer Engineering Department

Nguyen Van Thang

Multiple Vectors based MEMC and a Deep CNN for Video Frame Interpolation

Supervisor Hyuk-Jae Lee

Submitting a Ph.D. Dissertation of Engineering

February 2019

Graduate School of
Seoul National University
Electrical and Computer Engineering Department

Nguyen Van Thang

Confirming the Ph.D. Dissertation written by
Nguyen Van Thang
February 2019

Chair	최기영	(Seal)
Vice Chair	이혁재	(Seal)
Examiner	조남익	(Seal)
Examiner	채수익	(Seal)
Examiner	이규중	(Seal)

Abstract

Block-based hierarchical motion estimations are widely used and are successful in generating high-quality interpolation. However, it still fails in the motion estimation of small objects when a background region moves in a different direction. This is because the motion of small objects is neglected by the down-sampling and over-smoothing operations at the top level of image pyramids in the maximum a posterior (MAP) method. Consequently, the motion vector of small objects cannot be detected at the bottom level, and therefore, the small objects often appear deformed in an interpolated frame. This thesis proposes a novel algorithm that preserves the motion vector of the small objects by adding a secondary motion vector candidate that represents the movement of the small objects. This additional candidate is always propagated from the top to the bottom layers of the image pyramid. Experimental results demonstrate that the intermediate frame interpolated by the proposed algorithm significantly improves the visual quality when compared with conventional MAP-based frame interpolation.

In motion compensated frame interpolation, a repetition pattern in an image makes it difficult to derive an accurate motion vector because multiple similar local minima exist in the search space of the matching cost for motion estimation. In order to improve the accuracy of motion estimation in a repetition region, this thesis attempts a semi-global approach that exploits both local and global characteristics of a repetition region. A histogram of the motion vector candidates is built by using a voter based voting system that is more reliable than an elector based voting system.

Experimental results demonstrate that the proposed method significantly outperforms the previous local approach in term of both objective peak signal-to-noise ratio (PSNR) and subjective visual quality.

In video frame interpolation or motion-compensated frame rate up-conversion (MC-FRUC), motion compensation along unidirectional motion trajectories directly causes overlaps and holes issues. To solve these issues, this research presents a new algorithm for bidirectional motion compensated frame interpolation. Firstly, the proposed method generates bidirectional motion vectors from two unidirectional motion vector fields (forward and backward) obtained from the unidirectional motion estimations. It is done by projecting the forward and backward motion vectors into the interpolated frame. A comprehensive metric as an extension of the distance between a projected block and an interpolated block is proposed to compute weighted coefficients in the case when the interpolated block has multiple projected ones. Holes are filled based on vector median filter of non-hole available neighbor blocks. The proposed method outperforms existing MC-FRUC methods and removes block artifacts significantly.

Video frame interpolation with a deep convolutional neural network (CNN) is also investigated in this thesis. Optical flow and video frame interpolation are considered as a chicken-egg problem such that one problem affects the other and vice versa. This thesis presents a stack of networks that are trained to estimate intermediate optical flows from the very first intermediate synthesized frame and later the very end interpolated frame is generated by the second synthesis network that is fed by stacking the very first one and two learned intermediate optical flows based warped frames. The primary benefit is that it glues two problems into one

comprehensive framework that learns altogether by using both an analysis-by-synthesis technique for optical flow estimation and vice versa, CNN kernels based synthesis-by-analysis. The proposed network is the first attempt to bridge two branches of previous approaches, optical flow based synthesis and CNN kernels based synthesis into a comprehensive network. Experiments are carried out with various challenging datasets, all showing that the proposed network outperforms the state-of-the-art methods with significant margins for video frame interpolation and the estimated optical flows are accurate for challenging movements. The proposed deep video frame interpolation network to post-processing is applied to the improvement of the coding efficiency of the state-of-art video compress standard, HEVC/H.265 and experimental results prove the efficiency of the proposed network.

Keyword: **frame interpolation, MEMC, CNN, small objects, repetition regions, FRUC**

Student Number: 2012-31285

Table of Contents

List of Tables

List of Figures

Chapter 1. Introduction

Video frame interpolation, also called frame-rate up-conversion (FRUC) is widely used in various applications from computer vision to visual display applications such as slow motion, animation, play back, and so on. In order to increase the video frame rate, intermediate frames are generated from two consecutive original frames. For Liquid Crystal Display (LCD) display applications, high frame rate video is desired in order to reduce blurring, particularly for fast motion videos. Visual quality is improved by up-converting the frame rate of standard video captured at 30 Hz or 60 Hz by a factor of two or more. In order to increase the frame rate of the videos, video frame interpolation is performed to generate intermediate frames. In addition, for media broadcast of movies, frame-rate up-conversion is critical to accommodate the frame rate difference of the industry standards. The movie industry typically operates with a 24 frame/second capture rate, while media broadcasts employ a 30 Hz standard. Indeed, there are many applications in which frame-rate up-conversion is necessary and high quality is important. Slow-motion is another application of the video frame interpolation. Instead of using expensive high-speed camera to capture many frames in a second, users can increase number of frames by using video frame interpolation algorithms that generate new frames from existing captured frames.

Typically, a video frame interpolation algorithm is consisting of two steps such that the first step is a motion estimation (ME) or optical flow (OF) estimation that derives the motion trajectories between two consecutive frames. The second step is motion compensated frame interpolation (MCFI) that synthesizes the intermediate

frames by using estimated motion trajectories. Consequently, the visual quality of the interpolated frames highly depends on the accuracy of the estimated motion trajectories and performance of frame interpolation algorithm also. Even extensive research efforts have been made to handle challenges in video frame interpolation problem, there are still existing very difficult cases for frame interpolation. These cases include the movement of small objects, the movement of a repetition pattern region, text objects, occlusion, reveal, and the complex movements of fast-moving objects and so on.

1.1. Hierarchical Motion Estimation of Small Objects

Conventional block-based hierarchical motion estimation suffers from a fundamental limitation in handling motion details such as the diverse movements of a small object in the background. In general, it is not easy to define a general size to classify an object into small one or not. Because a part of a large object in a block can also be defined as a small object when the motion vector of the small part is different from the motion vector of the block that includes the small part. The two primary reasons for the fundamental limitation in handling motion of small objects are image down-sampling and motion over-smoothing. As down-sampling reduces the size of an object at the top pyramid level, it has little effect on the motion estimation for a block that includes the object. Therefore, the motion of a small object is often neglected in the motion estimation at the top level. The over-smoothing of a motion vector occurs in a conventional maximum a posterior (MAP) [9], [11] – [16], [22], including the operation to increase the smoothness of a motion vector field. Over-smoothing occurs when a motion vector of a block is different from the motion

vectors of neighboring blocks. MAP replaces the motion vector of a small object block with a background motion vector, which often results in the selection of a wrong motion vector of a block that includes a small object.

Fig. 1-1 shows an example that illustrates the limitation of the conventional block-based hierarchical motion estimation. In this example, an area of 16 × 16 pixels in an input image is used, and the number of pyramid levels is three. The scaled image of each level is partitioned into 4 × 4 blocks, and each of these blocks is represented by a square. A small object is represented by a shaded area while the background is represented by a white one. For a simple illustration, the number in each block represents only the horizontal component of the estimated motion vector. In this example, a small object moves in a different direction from the direction of movement of the background. The ground truth of the motion vectors is shown in Fig. 1-1 (a), in which a shaded block corresponds to a small object. In block-based hierarchical motion estimation, an input image is down-sampled and motion vectors are estimated from the top level. Fig. 1-1 (b) shows a top level block obtained by down-sampling the image area in Fig. 1-1 (a). The portion of the small object in the block at the top level is small; therefore, the motion of the small object has little effect on the motion estimation of the block. Thus, the estimated motion vector only represents the motion of the background whereas the motion of the foreground is ignored. As shown in Fig. 1-1 (c) and (d), the motion vector of the small object is not propagated from the higher level. Consequently, the motion vector of a small object is inaccurately determined, as shown in Fig. 1-1 (d), which is different from the ground-truth motion vector shown in Fig. 1-1 (a). The wrong motion vector causes the small objects to appear distorted or to disappear in the interpolated frame.

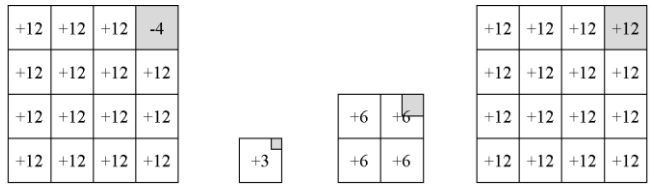


Figure 1-1. Example of an inaccurate motion vector of a small object in a hierarchal motion estimation

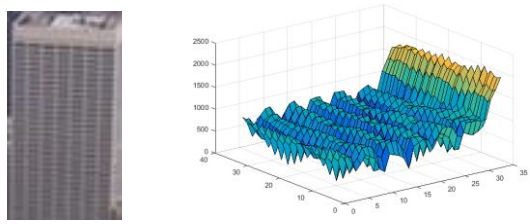
(a) the ground-truth motion vector at the bottom level. (b) a block with a motion vector at top level; (c) blocks with motion vectors at lower levels; (d) error of the estimated motion vector of a small object at the bottom level

This thesis addresses the difficulty in the motion estimation of a small object described in Fig. 1-1 and proposes a new hierarchical motion estimation algorithm for MC-FRUC with two primary contributions.

1.2. Motion Estimation of a Repetition Pattern Region

In many video sequences, the existence of repetition pattern objects is frequent such as in urban scenes with high building, the fence of gardens, decorative images, and so on. For frame rate up conversion, the derivation of an accurate motion vector is important to ensure the high visual quality of the interpolated frame. However, a repetition pattern in an image makes it difficult to derive an accurate motion vector because multiple similar local minima exist in the search space of the matching cost for motion estimation. Fig. 1-2 shows an example of the repetition pattern region with multiple similar local minima in the SAD surface of a block in the region. From

this SAD surface, it is hard to decide which one becomes the smallest minimal among many ambiguous similar local minima. Consequently, the motion vector obtained by the smallest SAD value become unreliable in this case.



(a) (b)

Figure 1-2. An example of a repetition pattern region and its SAD surface.

(a) Repetition pattern region, (b) SAD surface

The thesis tackles the multiple local minima problem by using a semi-global approach that obtains an accurate motion vector for a repetition pattern region.

1.3. Motion-Compensated Frame Interpolation

Even though motion estimation is a primary component to contribute on the performance of video frame interpolation algorithms, as described on previous sections, there are still existing many challenging cases that the state-of-arts algorithms fail to estimate motion trajectories of pixels. Consequently, artifacts appear in interpolated frames are un-avoidable. In order to alleviate those of un-avoidable artifacts, in the second stage of FRUC, motion compensated frame interpolation algorithm can generate a better visual quality given errors in the estimated motion vectors. Motion Compensation (MC) algorithm is the second stage

among two main elements of an FRUC algorithm. The task of MC is to generate the intermediate frames, given the motion vectors fields obtained from the previous step, motion estimation.

This thesis proposes a new method that uses multiple motion vectors as a tool for alleviating the errors of motion trajectories obtained by motion estimation step.

1.4. Video Frame Interpolation with Deep CNN

Typically, a video frame interpolation algorithm is composed of two decomposed steps such that the first step is a motion estimation (ME) or optical flow (OF) estimation that derives the motion trajectories between two consecutive frames. The second step is motion compensated frame interpolation (MCFI) that synthesizes the intermediate frames by using estimated motion trajectories. These flow-based methods inevitably generate ghost or blurry artifacts, owing to the errors in estimated optical flows. In other words, this classic approach highly depends on the accuracy of the motion trajectories.

Recently, the break-through of Convolutional Neural Networks (CNN) in computer vision [49, 50, 51], [52, 54, 55] allows a formulation of video frame interpolation as an end-to-end learning process without optical flow estimation. In those methods, however, the objective function or loss function only focuses on pixel difference. Consequently, it usually fails in the estimation of fast and/or complex movement which requires a critical role of motion estimation for high-quality frame interpolation. A phase-based frame interpolation, proposed in [60], [61] is another approach to generate the intermediate frames without estimating optical flow.

However, similar to the above CNN based methods, the phased based approaches also fail in the fast movement.

This thesis presents a comprehensive framework that glue two above previous approaches into a single stacked CNN network that is composed of a back-to-back stack of two CNN networks. In addition, with high performance of the deep video frame interpolation, this thesis extends the application of the network to two related problems, enhance quality of compressed videos, or also called post-processing for compressed videos. The second extension is to improve coding efficiency of the latest video compression standard, HEVC/H.265 by applying the proposed deep video frame interpolation algorithm into HEVC.

1.5. Outline of the Thesis

This section outlines the main contributions of this thesis. The first and second contributions of this thesis are accurate motion estimation algorithms that handle very challenging cases in FRUC, such as the movement of small objects, the movement of a repetition pattern region, respectively. Next, a new motion compensation with multiple motion vectors are proposed to enhance visual quality of the interpolated frames. Last but not least, an end-to-end learning deep convolutional neural network is proposed to generate intermediate frames with high accuracy and outperforms previous state-of-art algorithms. The extension of this deep video frame interpolation algorithm for other applications such as post-processing for compressed videos and improving coding efficiency of the latest video compression standard HEVC/H.265 are also presented.

Previous works of the mentioned problems are introduced in chapter 2.

Chapter 3 presents the proposed method for hierarchical motion estimation for small objects in FRUC.

The proposed semi-global accurate motion estimation for a repetition pattern region in FRUC is shown in chapter 4.

Chapter 5 presents the proposed multiple motion vectors based motion compensation in FRUC.

In chapter 6, the proposed deep video frame interpolation with a CNN network and its extensions to related tasks are presented.

Finally, chapter 7 concludes contributions of this thesis.

Chapter 2. Previous Works

2.1. Previous Works on Hierarchical Block-Based Motion Estimation

Extensive research efforts have been made to handle the motion estimation for challenging cases in frame-rate up-conversion, from repetition pattern objects [17], [18], [19] to small objects [3]. A previous study proposes a SIFT feature-based optical flow in order to explore the motion vector of a small object [59]. The method proposed in this study successfully improves accuracy but results in higher computational complexity. For example, the time required to estimate the flow of an urban test sequence in the Middlebury test bench is 342 s. Another method uses variable block sizes at motion boundary blocks to provide a dense motion vector field [4]. This method succeeds in deriving accurate motion vectors at boundary blocks but it requires extensive computations. In the method proposed by [5], a pixel-based motion vector selection is derived from neighboring block-based motion vectors. The motion vectors of the pixels are generated from the estimated motion vectors of the blocks that include them. The pixel-based estimation further improves the accuracy of motion estimation, although small objects may remain undetected if the motion estimation for a block is inaccurate. Recently, Jeong, Lee and Kim propose the use of video segmentation for estimating motion vectors of pixels [6]. The method can generate a dense motion vector field and successfully reduce block artifacts. However, the computational complexity is high owing to the derivation of video segmentation and graph cut algorithm. Variable block size approaches have

been also studied in previous works [4], [6], [7]. The method proposed in [9] increases the density of a motion vector field in a hierarchical manner. In other words, the motion vector of a sub-block is derived from the motion vector of a parent block and those of the neighboring blocks of the parent block. This method successfully reduces the computational complexity while offering a reasonable level of accuracy. However, it cannot reduce the motion vector errors owing to the disappearance of the motion vector of small objects from the motion estimation of the original blocks.

2.1.1. Maximum a Posterior (MAP) Framework

Recently MAP-based motion estimation has achieved better performance than the conventional block matching algorithm (BMA) because it exploits smoothness constrains of motion fields [9], [12], [13]. The smoothness constraint on neighboring motion vectors can improve the estimation accuracy thanks to the property that motion vectors in an object do not change abruptly. The smoothness constraint is a key contribution of many optical flow methods [20], [21], and block based motion estimation methods [9], [13], [22]. In [20], Horn and Schunck propose an algorithm that uses a smoothness constraint as a penalty for pixel-matching scores in dense motion field estimation. Zach et al. compute the smoothness term with a norm L1 of motion vector difference between neighboring ones [21]. In recent block based motion estimation [9], [13], [22], smoothness constraint is used as a key approach to find true motion vectors. The BMA is an unconstrained optimization; meanwhile, MAP applies prior probability to the optimization in order to make a smooth change in motion vector field. In MAP, the objective function is to minimize an energy function that is composed of two components. The first term is a data cost that

represents the block matching value or likelihood, and the second one is a smoothness cost that encodes a prior probability of the motion vector field. The combination of two components, which are likelihood and prior probability, is to estimate the posterior probability of the motion vector field as shown in the following equations:

$$E(u) = \text{SAD}(u) + \lambda * \Sigma P(u, v_c) \text{ with } v_c \in N_c \quad (2.1)$$

$$\boldsymbol{E(u) = \sum_{x \in Block} |I_c(x) - I_r(x+u)| + \lambda * \sum_{v_c \in N_c} \{\|u - v_c\| * \theta(u, v_c)\}} \quad (2.2)$$

where $u = (u_x, u_y)$, is the motion vector variable, SAD(u) is the sum of absolute difference of the block that corresponds to u, $P_c(u, v_c)$ is the smoothness function that corresponds to the motion vector difference between neighboring blocks, N_c is a neighboring system of current motion vector u, and λ is a weighting parameter. Eq. (2.2) is a specific formulation of Eq. (2.1), where $I_c(x)$ is the intensity of a pixel at position x in the current block. $I_r(x+u)$ is the intensity of the corresponding pixel (x+u) in a reference block, $\theta(u,v_c)$ is a threshold continuity function that is equal to zero when the difference between u and v_c is larger than a predefined threshold value, otherwise, it is equal to one. During the optimization of the energy function E(u), u varies within the search range S that is a 2-D value table, i.e (±16, ±16). The final estimated motion vector $\hat{u}$ that optimizes the energy function E(u) is defined as following:

$$\boldsymbol{\hat{u} = argmin_{u \in S}\{E(u)\}} \quad (2.3)$$

In general, MAP-based methods outperform the conventional BMA method. However, in areas with small objects in which the motion vectors are different from the motion of the surrounding background, over-smoothing in motion vector

typically occurs. In these areas, BMA tends to yield more accurate motion vectors. Thus, this thesis proposes an algorithm for using motion vectors obtained by BMA for the motion estimation of small object areas.

2.1.2. Hierarchical Motion Estimation

For real time operation of LCD TVs, an FRUC algorithm must be sufficiently fast to process 60 frames per second. In order to satisfy this strict requirement, a hierarchical motion estimation framework has been used for reducing the computational complexity [8], [9], [10]. To obtain precise motion vector field, the MAP method is used at the top level [9]. Subsequently, the top motion vectors are propagated to the bottom level to produce finer motion vector fields. In this manner, the images at all pyramid levels are partitioned into blocks of the same size. To estimate motion vector of a block at a level, three motion vectors from the upper level are used as its initial motion vectors. The first one is from its parent block, and the other two are from the blocks both horizontally and vertically adjacent to the parent block. Fig. 2-1 illustrates an example. The full search around the three initial motion vectors with a search distance of ±d pixels are performed to choose the best motion vector in three search windows. The motion estimation for each layer is recursively performed in this manner from the top to the bottom level in the image pyramid. If there are missing motion vectors at the top level, the propagation cannot discover the missing ones at the bottom level. This is the primary drawback of the conventional hierarchical motion estimation. Thus, this thesis proposes a new hierarchical motion estimation algorithm that discovers the missing motion vector of small objects at the top level and propagates it into the bottom level. With this

manner, the proposed algorithm successfully preservers the motion vector of small objects in hierarchical motion estimation framework.

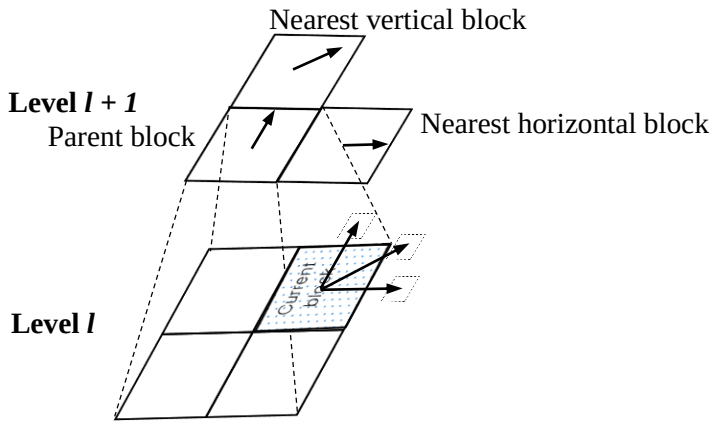


Figure 2-1 Hierarchical Motion Estimation [9].

2.2. Previous Works on Motion Estimation for a Repetition Pattern Region

For frame rate up conversion, the derivation of an accurate motion vector is important to ensure the high visual quality of the interpolated frame. However, a repetition pattern in an image makes it difficult to derive an accurate motion vector because multiple similar local minima exist in the search space of the matching cost for motion estimation. A number of previous algorithms have been proposed to reduce the matching errors in the estimation of motion vectors in repetition regions. In [24], an exhaustive full-search motion estimation is used to find solutions for repetition pattern regions. In [25], the motion vectors of repetition pattern blocks are corrected by recursive average operations. In [26], the spectral image is analyzed to

estimate the motion vectors for repetition regions. In [27], a new design methodology is explored by using suboptimal measures for two different motion estimation algorithms. In [28-31], a Maximize-A-Posterior (MAP) based method is proposed for motion estimation. The matching cost is regularized by a smoothness constraint in order to improve the accuracies of motion vectors. These previous methods use a local approach that estimates or corrects the motion vector of a repetition pattern block by using only the information from the block itself and its neighbors. These methods sometimes miss the corrected motion vector because multiple similar local minima exist in the search space of the matching cost for motion estimation.

2.3. Previous Works on Motion Compensation

Conventional MC-FRUCs utilize two approaches for motion trajectory estimation: unidirectional motion estimation [9], [23], [34] and bi-directional motion estimation [35] – [37]. The former approach is based on a typical motion estimation algorithm that divides one frame into non-overlap blocks, estimates the motion vector of each block by searching the best matching block in the other frame, and finally generates the intermediate frame that composes of the blocks resulting from the motion compensated interpolation of the corresponding blocks along the estimated motion vector. The main problem of this approach is that the interpolated blocks may not be contiguous in the interpolated frame, that is, some blocks are overlapped while some areas are not filled with the interpolated blocks resulting in a hole in the interpolated frame. An overlap is generated by crossing of multiple motion vectors while a hole results from no motion vectors crossing at the blocks in the interpolated frame. In case multiple motion vectors point through the same an interpolated block, an

expanded-block weighted motion compensation is proposed in [23] in order to reduce block artifact and overlap issues. In the method, from each unidirectional motion vector field, three kinds of intermediate images are generated to accumulate weighted motion compensated pixels, weighted motion compensated difference and contributing weights. Each kind of the intermediate image is high bit depth one because it accumulates contribution of all passing candidates. Therefore, the buffer size for storing those intermediated images is very large. In addition, the weighting coefficients are predefined by using a fix window size, does not relate to the accuracy of passing motion vectors. As a side effect, the hole shape is produced by the method is arbitrary, it causes challenging for hole filling problem.

The Overlaps and holes can be avoided by the bidirectional approach that divides the frame to be interpolated into non-overlap blocks before it is generated, estimates the motion vector of each block by searching two symmetric matching blocks in the two original frames. Each block has symmetric two motion vectors, one points to the previous frame and the other points to the next frame. However, this approach offers less accurate motion vectors than the former one. In addition, bi-directional motion estimation methods usually go along with an overlapped block motion compensation (OBMC) frame interpolation algorithm [32] or its variant adaptive OBMC (AOBMC) [36] to alleviate the blocking artifacts. However, both OBMC and AOBMC use motion vector of neighbor blocks, therefore they can produce over-smoothing artifacts as adjacent blocks have substantially different motions.

Recently, a hybrid approach [38] – [40] has been proposed to combine unidirectional ME and bidirectional MC. This approach intends to avoid overlaps

and holes in the unidirectional approach and at the same time to reduce inaccuracy in bi-directional motion field. In [38], Yoo et al estimate unidirectional motion fields first, then it computes bidirectional one by using scaled motion vectors of the collocated blocks in original frames obtained by unidirectional ME as the search centers with a small search range for bi-directional ME. However, mapping motion vectors from collocated blocks cause errors when the motion vectors are large. In [39, 40] projection based motion vector mapping is used to generate bi-directional motion vectors from unidirectional ones. In case an interpolated block has multiple overlapped projected blocks, a Sum of Bilateral Absolute Difference (SBAD) is applied as the metric for selection, the motion vector with the minimum SBAD is selected as the final one of the interpolated block. A problem with this method is that it will select wrong motion vector even with the smallest value of SBAD because the minimum SBAD doesn't guarantee that it represents the truth motion vector of the interpolated block. For example, at smoothness areas in images where there exist multiple local minima, it causes many ambiguous motion vectors.

2.4. Previous Works on Video Frame Interpolation with Deep CNN

Video frame interpolation: Extensive research efforts have been made to handle the challenges in video frame interpolation. A typical approach in video frame interpolation estimates dense motion vector fields, or optical flows, between two original input frames and then interpolates intermediate frames guided by the estimated motion [47, 48, 53, 59, 62]. To synthesize an output image from the input frames, the estimated flows based warping operations using bilinear interpolation are

done first, and later the warped frames are blended together. Consequently, the flow-based methods generate ghost or blurry artifacts when the warped frames are not aligned well, owing to the errors of the estimated optical flows. In order to replace simple blending operations, Nikaus et al. [56] propose to use a context-based synthesis network to generate the intermediate frames from the pre-warped frames. It is shown that the frame synthesis network outperforms simple blending algorithms. Recently, inspired by the success of applying deep learning to optical flow estimation [46, 67, 69, 70, 62], CNNs are used for video frame interpolation with the loss function to calculate the pixel difference between the synthesized one and its corresponding ground-truth. CNN based methods remove optical flow step and handle video frame interpolation as a convolution process [52, 53, 54, 55, 57, 64]. In other words, the network can be trained to synthesize images without explicit motion estimation step. Consequently, it usually fails at regions with fast, complex moving objects where accurate motion information is crucial for synthesis task.

Starting from the work by Long et. al. in [52] which employs an auto-encoder network, several recently-proposed deep neural networks successfully improves the quality of video frame interpolation. The auto-encoder architecture or U-net architecture used in [55], [57] extract features that are given to the sub-nets for the synthesis of the intermediate frame. SepConv network in [55] successfully handles blurry artifacts by estimating independently four 1D kernels which are then convolved with the input frames to generate interpolated frames. However, SepConv network does not take into consideration the motion constraints among neighboring kernels because the kernels for each pixel are trained independently from those of neighboring pixels. A deep neural network is also used to directly estimate the phase

decomposition of the intermediate frame in [51] based on the application of the phase based frame interpolation which is originally proposed by Meyer et al. in [60] to generate intermediate frames by modifying a per-pixel phase.

A stack of networks: A stack of component networks is proved to enhance the performance of the whole network in various tasks including pose estimation [45], object detection [41], document image unwarping [42], optical flow [69] and so on. In [45], stacked hourglass networks are proposed for human pose estimation and they outperform long single hourglass networks as claimed by authors. In [41], the stack of two hourglass networks is the backbone network of CornerNet to generate features for two prediction modules. In [42], a stacked U-Net with intermediate supervision is used to directly predict the forward mapping between the warped images and the refined version. For optical flow, Flownet 2.0 [69] also employs a stack of several sub-networks and achieves a significant improvement from the previous version. This thesis adopts the idea of a stack of sub-networks for video frame interpolation.

Chapter 3. Hierarchical Motion Estimation for Small Objects

3.1. Problem Statement

In hierarchical motion estimation, an input image is down-sampled for generating the top pyramid layer in which the size of an object becomes smaller than that in the bottom layer. Consequently, a small object typically occupies only a small part of a block at the top level. Therefore, the small object may be ignored in motion estimation, and the remaining region of a block contributes more significantly to motion estimation than the small object does. If the motion information for a small object is ignored at the top level, it cannot be recovered at the bottom level. Thus, hierarchical motion estimation often fails in the generation of a correct motion vector for a small object. However, this small object may be sufficiently large to occupy an entire block at the bottom level, and the erroneous motion vector of a small object can deteriorate the image quality in MC-FRUC. Therefore, it is necessary to discover and store the motion information of a small object at the top level and to pass it to be used for motion estimation at the bottom level.

This study addresses the difficulty in the motion estimation of a small object described in Fig. 1-1 and proposes a new hierarchical motion estimation algorithm for MC-FRUC with two primary contributions.

- The hidden motion information of a small object at the top level is represented by *an alternative motion vector candidate*. The alternative candidate is propagated to the lower levels and used for the motion estimation of small objects at the bottom layer.

- A matching algorithm for determining the alternative motion vector is proposed. If pixels with high residual costs are detected in a block, the matching algorithm is performed for the high cost pixels, otherwise it maintains the motion vector estimated by a full search block matching algorithm as the alternative motion vector.

This thesis aims to propose a novel algorithm for hierarchical motion estimation that avoids the artifact in the region that includes a small object. Unlike the algorithm proposed in [9], each block at a lower level has three motion vector candidates: one from the motion vector of the parent block, and the other two from the motion vectors of the nearest neighboring blocks of the parent block in the horizontal and vertical directions. This thesis proposes the use of an additional motion vector candidate that represents the motion information of a small object at the top level. The additional candidate is propagated to the lower level and used for motion estimation of small objects.

3.2. The Alternative Motion Vector of High Cost Pixels

The proposed algorithm attempts to detect a small object that has a motion vector different from that of the block that includes a small object. In this case, it is possible to have a case that the movement of a small object is different from that of the surrounding area in the block. The matching error of the block may be high because small object pixels may not have matching pixels in a reference block. In this case, the matching error of the pixels that belong to a small object is high. The pixel difference, ΔI, is defined by the following equation:

$$\Delta I = |I_c(i,j) - I_r(i+u, j+v)| \quad (3.1)$$

where $I_c(i, j)$ is the intensity of the pixel at position (i, j) in the current frame. $I_r(i+u, j+v)$ is the intensity of the corresponding pixel (i+u, j+v) in the reference frame. Vector (u, v) is the motion vector of the current block to be derived.

Herein, a pixel with a large pixel difference is referred to as a high-cost pixel that has a potential to be a pixel of a small object. If the pixel difference is larger than the predefined threshold, it is determined as a high-cost pixel. When a block contains high-cost pixels, the second full search motion estimation for the high-cost pixels is performed to estimate the motion vector of a small object that consists of these pixels. In the second motion estimation, only the matching cost of the high-cost pixels is considered, and thus, a motion vector of a small object can be found. A motion vector that represents the motion of a small object is referred to as an alternative motion vector.

Fig. 3-1 shows an example of motion estimation for a 4x4 block at the top level. In Fig. 4 (a), the current block includes a part of a small object that is represented in black, and it does a rest part of a background that is in white. In the first motion estimation, a motion vector of a block is estimated as +3. Fig. 3-1 (b) shows a matching block in the reference frame. When the current block is compared with the matching block, difference of the small object is high, and thus, these pixels are determined as high-cost pixels which are represented by shaded pixels in Fig. 3-1 (c). In the second motion estimation, only high-cost pixels are used for computing SAD, and an alternative vector of -1 is derived in this example. In the proposed

algorithm, two motion vectors of +3 and -1 are to be propagated to the lower layers. If the number of layers is three as shown in Fig. 1-1, the motion vectors of +12 and -4 can be obtained at the bottom layer. For each block at the top level, two motion vectors are derived. The first one represents the motion of the block, and the second one represents a motion of a small object in the block. The propagation of both motion vectors to the finer levels allows the motion of the small objects to be preserved from the top layer to the bottom layer.

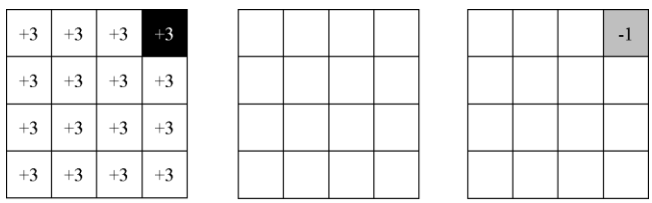


Figure 3-1 Example of the alternative motion vector

(a) A current block with a small object in black pixels, (b) A matched block in a reference frame, (c) A determined high-cost pixels and the alternative vector obtained by the second motion estimation.

Each block has two motion vectors. One is the motion vector of the block from the first motion estimation, and the other is the motion vector of high-cost pixels from the second motion estimation. Even when no high-cost pixel exists in a block, the motion vector of a block can be wrong owing to over-smoothing of the MAP-based methods. This case may occur for a block in which all pixels belong to a small object with its size almost the same as the block size. In this case, the BMA can obtain a true motion vector of the block. However, the true motion vector can be replaced

with a false one by MAP when the small object moves in a direction different from that of the background. In the proposed algorithm, the motion vector from the BMA is assigned to an alternative vector for the blocks that do not contain high-cost pixels. This ensures that all potential motion vectors for a small object are propagated to the lower layers.

3.3. Modified Hierarchical Motion Estimation

In the proposed algorithm, four motion vectors from the upper level of an image pyramid are used as the initial motion vectors for motion estimation. The three motion vectors are the same as those of the conventional algorithm [9]. The additional candidate is the alternative motion vector discussed in the previous subsection. If a block at the top level includes high-cost pixels, the alternative motion vector is the motion vector of the high-cost pixels. Otherwise, a motion vector obtained by the BMA for a block at the top level is used for an alternative motion vector. Four motion vectors are propagated to the lower level, and then the full-search BMA around the four motion vectors with a search distance of ±d pixels are performed to choose the best among the four search windows. Even when the alternative motion vector is not selected as the best one, it is still propagated to the next lower level to preserve the motion vector of the small object. Motion estimations for the lower layers are performed in this manner again in the image pyramid as shown in Fig. 3-2. At a finer layer or level l, in each current block (pattern fill block in Fig. 3-2), three dashed arrows represent the three conventional motion vectors, the other is the alternative motion vector.

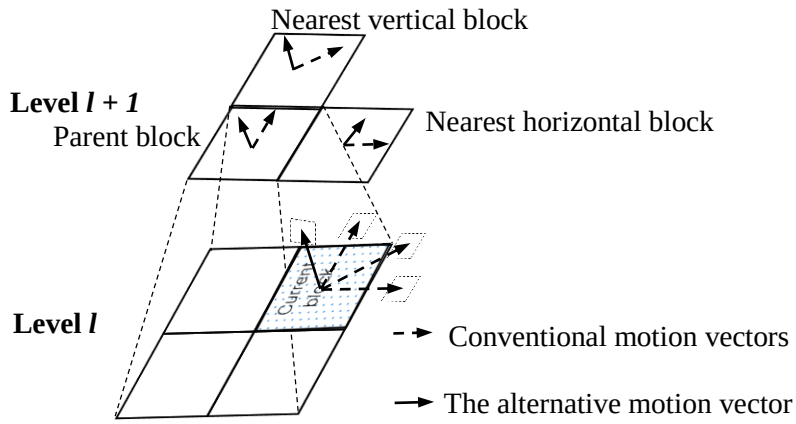


Figure 3-2 The Modified Hierarchical Motion Estimation

3.4. Framework of the Proposed Algorithm

The proposed algorithm is shown in Fig. 3-3. First, from input frames, image pyramids are constructed for hierarchical motion estimation. Then, a conventional full search BMA is performed at the top pyramid level, and the high-cost pixels of each block are detected. In the next step, two operations are performed in parallel. One is a MAP-based motion estimation that is performed as a refinement of the BMA [9]. The other is a full-search motion estimation for high-cost pixels. When a block does not include high-cost pixels, the motion vector estimated by BMA is used for an alternative motion vector. Therefore, all blocks have two motion vectors. One is estimated by the MAP-based motion estimation, and the other is the alternative motion vector. These two motion vectors of the top level are propagated to the lower level in which these vectors are used for generating search windows. After the motion estimation of the level is completed, the motion vector from BMA and the scaled alternative motion vector for each block are propagated to the next level. The alternative motion vector is propagated to the next pyramid level irrespective of whether it is chosen as the motion

vector of the block or not, thereby guaranteeing that the motion vector of the small object is propagated to the bottom layer.

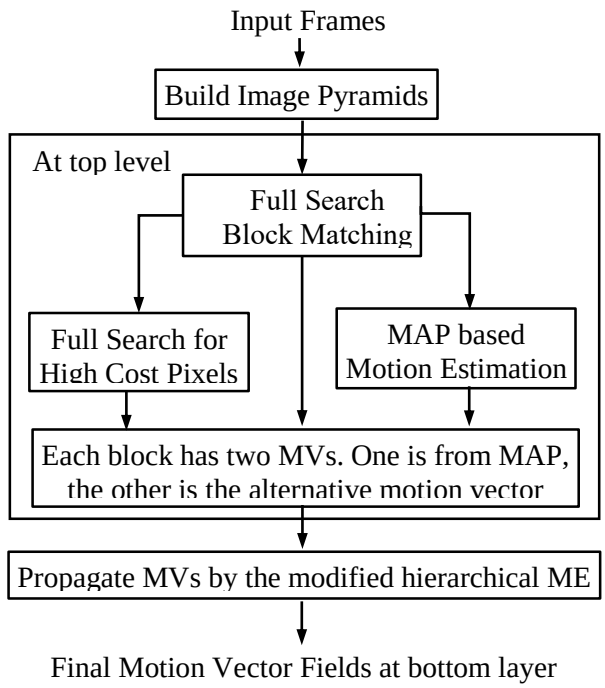


Figure 3-3 Motion estimation of the proposed algorithm

3.5. Experimental Results

For experiments, the proposed algorithm is evaluated with four full-HD video sequences that contain small objects: tennis ball, rim ball, basketball and soccer ball. For video frames in the dataset, odd frames are removed, and these frames are used for ground-truth frames. Motion compensated frame-rate up-conversion algorithms are applied to even frames to generate intermediate frames, which are compared to the corresponding ground-truth frames. The performance of the proposed motion

estimation is compared to that of the previous method that uses the MAP algorithm at top pyramid level and conventional hierarchical motion estimation [9]. For motion estimation, experiments are performed with the previous and proposed algorithms under identical conditions as follows: three temporally consecutive original frames are used for estimating both forward and backward motion vector fields as suggested by [9], the number of pyramid levels is four, and the block size is fixed to 8×8 for all pyramid levels. One block at level l is a parent of four blocks at level l + 1. At the top pyramid level, the search range is ±16 pixels in the horizontal direction and ±8 pixels in the vertical direction in order to reduce search space in the vertical direction. At the other levels, the small search range d is ±1 for both horizontal and vertical directions. The image size at the top level is 240×135 pixels, at the bottom level is 1920x1080 pixels. For frame interpolation, the algorithm in [23] is used for both previous and proposed motion estimations. The peak signal-to-noise ratio (PSNR) values of interpolated frames are used as an objective comparison metric. In addition, the subjective visual image quality is also compared.

3.5.1. Performance Analysis

Fig.3-4 presents an example of over-smoothing of MAP approach. Figs. 3-4 (a) and (b) show two consecutive input frames. Fig. 3-4 (c) shows magnified input image that includes a small object. In this figure, white lines represent blocks corresponding to blocks at the top pyramid level. The blue arrows represent the motion vectors estimated by the BMA, the red arrows represent the motion vectors estimated by MAP. For the two center blocks that contain a part of the ball, the BMA estimates correctly the motion of the part of the ball while the MAP over-smooths it to make

the motion vector of the ball similar to those of neighboring blocks that belong to background with different movement. Fig. 3-4 (d) shows the interpolated frame when MAP is used, and broken artifact is generated owing to some parts of the object generated with the erroneous motion vectors. Fig. 3-4 (e) shows the interpolated one when the alternative motion vector with BMA is used. The interpolated frame with the alternative motion vector preserves well the shape of the ball. This proves that the efficiency of the preservation of the motion vector of small objects with the alternative motion vector obtained by the BMA.

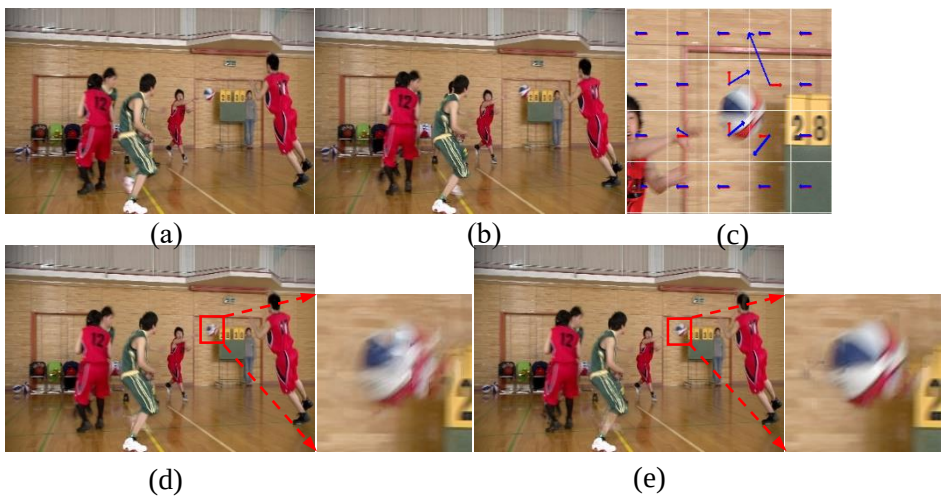


Figure 3-4 Effect of over-smoothing of MAP and the alternative motion vector with BMA.

(a) The first frame, (b) The second frame, (c) The scaled motion fields with the alternative motion vectors obtained by BMA (the blue arrows are the alternative motion vectors; the red arrows are the motion vectors of blocks obtained by MAP), (d) The interpolated frame without the alternative motion vector, (e) The interpolated frame with the alternative motion vector

Fig. 3-5 presents an example of the alternative motion vector for the detected high cost pixels. Figs. 3-5 (a) and (b) show two original frames. In Fig. 3-5 (c), each block represents a top level block at the top pyramid level, it is scaled to a corresponding 64x64 block at the bottom pyramid level. The blue and red arrows represent the motion vectors obtained by the conventional BMA and MAP, respectively. The blue dots denote high-cost pixels. The yellow arrows show the alternative vectors represent the movement of the detected high cost pixels. The small tennis ball moves to the upper right corner. However, this movement is dismissed by the dominance of the background (grass) in the block. If only the motion vector obtained by the BMA or MAP is propagated, the true motion of the tennis ball cannot be found at the bottom layer. Then, the tennis ball can be missed or exist with the deformed shape in the interpolated frame as shown in Fig. 3-5 (d). With the proposed alternative vector, the true motion vector of the tennis ball is persevered and propagated to the bottom pyramid level. Therefore, it guarantees that the correct motion vector of the tennis ball can be used for frame interpolation that generates the intermediated frame as shown in Fig. 3-5 (e).

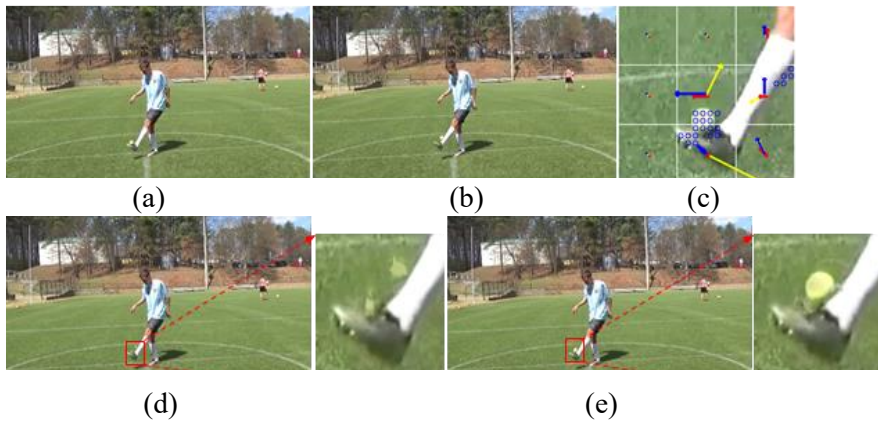


Figure 3-5 Effect of down-sampling and the alternative motion vector with the detected high cost pixels.

(a) The first frame, (b) The second frame, (c) The scaled motion fields with the alternative motion vectors obtained by the second motion estimation for the high cost pixels (the yellow arrows are the alternative motion vectors, the blue arrows are the motion vectors of blocks obtained by BMA, the red arrows are the motion vectors of blocks obtained by MAP), (d) The interpolated frame without the alternative motion vector, (e) The interpolated frame with the alternative motion vector

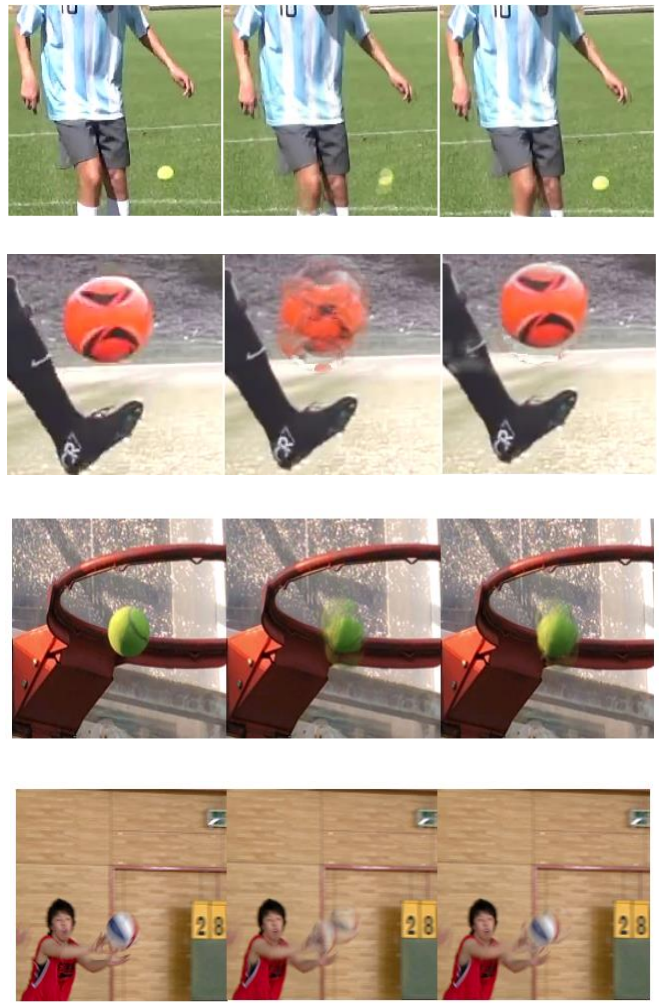
3.5.2. Performance Evaluation

The objective quality of the proposed algorithm is compared to that of the MAP algorithm in [9] in Table 3.1 which shows the comparison of the PSNR. The improvement achieved by the proposed algorithm is about 0.42 dB on an average. Fig. 3-6 presents the comparison of the subjective image qualities of the previous work in [9] and the proposed method. The first column represents the ground-truth

frames, the second column shows the interpolated frames of the previous work in [9], and the third one shows the frames generated by using the proposed algorithm with the alternative vector. In the MAP algorithm, there are broken artifacts in the interpolated frames because the motion vectors of some parts of the small balls are lost. The proposed algorithm reduces the broken artifacts significantly in comparison with the MAP algorithm. The proposed algorithm preserves the shapes of the small object because the motion vectors of the whole parts of the small objects are estimated and preserved by the alternative vectors.

Table 3.1 PSNR comparisons between the MAP algorithm [9] and the proposed method

Test sequence	PSNR (dB)		
	The MAP algorithm [9]	Proposed method	Improvement
Rim ball	35.30	35.83	0.53
Tennis ball	34.71	34.96	0.25
Basketball	29.06	29.46	0.40
Soccer ball	38.97	39.48	0.51
Average			0.42



(a) (b) (c)

Figure 3-6. Visual comparison between the previous MAP algorithm [9] and the proposed method

(a) Ground truth, (b) MAP [9], (c) The proposed

Chapter 4. Semi-Global Accurate Motion Estimation for a Repetition Pattern Region

4.1. Problem Statement

The previous methods for motion estimation of a repetition pattern region, use a local approach that estimates or refines the motion vector of a repetition pattern block by using only the information from the block itself and its neighbors. These methods sometimes miss the corrected motion vector because multiple similar local minima exist in the search space of the matching cost for motion estimation. This thesis tackles the multiple local minima problem by using a semi-global approach that obtains an accurate motion vector for a repetition pattern region. The idea of the proposed algorithm comes from the following observation. Repetition pattern blocks share the same motion vector that is the motion vector of the whole repetition pattern region. Therefore, the blocks in a repetition region can be merged to form a repetition pattern region and a single motion vector is derived for the merged region. The larger the repetition region is, the more accurate the estimated motion vector is. This merging based method obtains a very accurate motion vector at the cost of the increased memory bandwidth and large memory buffers to store pixels of the region and save the pixel differences. Therefore, the proposed algorithm uses a semi-global approach in order to replace the global approach that estimates the motion vector of the whole region. As a result, the semi-global approach reduces the computational complexity while maintaining the accuracy of motion estimation. The proposed algorithm is the first attempt to adopt the semi-global approach to estimate the

motion vector of the repetition pattern blocks. It efficiently handles multiple local minima problem of repetition pattern blocks.

4.2. Objective Function and Constrains

Because a local approach cannot handle multiple local minima problem of repetition pattern blocks, the motion vector field obtained by previous correction methods still include many noisy and unreliable motion vectors as shown in Fig. 4-1(b). Observation of the example image shows that the repetition pattern blocks share the same motion vector that is the motion vector of the whole repetition pattern region. Therefore, the repetition pattern blocks can be merged together and the motion vector is derived for the whole region. Motion estimation for the whole region can obtain an accurate motion vector because it exploits the global property of the movement of the repetition pattern region. However, the motion estimation for a large region consumes very large hardware resources, i.e. a repetition region size is 256x128, at each search position i, it has to fetch 32 768 reference pixels to compute a SAD(i) value. Consequently, it takes many cycles to load pixels from memory. Therefore, it is necessary to find another way to compute the motion vector for a repetition pattern region without additional search operation. In other words, the motion vector of the whole region is derived by exploiting the global property to achieve good accuracy but using local approach to reduce computational complexity.

Objective function is to find a motion vector for the repetition region

$$mv_{region} = argmin\{SAD(u)\} \text{ for pixels in the region, } u \in \boldsymbol{seach\ range} \quad (4.1)$$

Constrains: Complexity is similar to Block Matching Algorithm (BMA) for a small 8x8 block.

4.3. Elector based Voting System

The most frequent motion vector among estimated motion vectors of repetition pattern blocks may be the representative motion vector for the whole region. However, the motion vectors of repetition pattern blocks are unreliable, and consequently, the derivation of the motion vectors from unreliable ones may also be unreliable. Fig. 4-1 shows an example of the estimated motion vector field obtained by full search block matching algorithm. Fig 4-1(b) shows that the motion vector field includes many wrong noisy motion vectors in the repetition pattern regions. The most frequent motion vector is (6, 15) but it is not the correct motion vector of the repetition pattern region. In fact, the ground-truth motion vector is equal to (6, 0). The derivation of the motion vectors by observing their histogram among the blocks in the repetition region may generate an accurate motion vector without an addition search operation. The proposed algorithm that combines the multiple similar local minima characteristic of the repetition pattern blocks and global property of the movement of the repetition pattern regions together to make the reliable derivation of the correct motion vector.

(a) (b)

Figure 4-1 Example of motion vector field for repetition pattern estimated by a local approach.

(a) Overlaid original frames, (b) Estimated motion vector field

This conventional method to build a histogram of MVs is the same as an elector based voting system

Given a repetition region, contain n blocks

block 1: motion vector $mv_1 = \text{argmin}\{SAD_1(u)\}$ for u ∈search range, becomes the elector of block 1

block 2: motion vector $mv_2 = \text{argmin}\{SAD_2(u)\}$ for u ∈search range, becomes the elector of block 2

and so on, block n: motion vector $mv_n = \text{argmin}\{SAD_n(u)\}$ for u ∈search range, becomes the elector of block n

And then, mv_1, mv_2…, mv_n are used to build a histogram of MVs. The drawback of this elector based voting system is the sub-optimization is decided for each block separately. Consequently, the global optimal cannot estimated accurately from the solutions obtained by the sub-problems for component blocks. In addition, these sub-optimizations are sensitive because many ambiguous values, owing to block size is

small, number of pixels are not large enough in order to verify the estimated motion vector reliable or not. Consequently, the motion vector field are noisy, contains many inaccurate motion vectors as shown in Fig. 4-1b.

4.4. Voter based Voting System

The proposed algorithm assumes that the global minimum of the matching cost of the entire repetition pattern region corresponds to one of the local minima of the blocks in the repetition region. Therefore, the motion vector of the repetition region can be obtained from the motion vector candidates of the blocks in the repetition region. Based on the above assumption, this thesis proposes a novel algorithm that builds a histogram of motion vectors from reliable ones obtained from a voter based voting system algorithm. The proposed algorithm consists of two steps. Step 1 makes a histogram of the motion vector candidates that are obtained during the motion estimation for individual blocks in the repetition region. Step 2 selects the most frequent motion vector candidate in the histogram to be the final motion vector of the entire repetition region.

In the proposed voter based voting system, the algorithm delays the sub-optimization of each block to the second stage that preserves the optimal solution of the optimization for the whole repetition region. The below is a description about the proposed voter based voting system.

Given a repetition region, contain n blocks

block 1: list of motion vector candidates [mv^1_1 , mv^2_1 , mv^3_1 , …, mv^{10}_1] = *multiple argument local min*{$SAD_1(u)$} for u ∈search range, becomes the reliable voters of

block 1

block 2: list of motion vector candidates [mv^1_2 , mv^2_2 , mv^3_2 , …, mv^{10}_2] = *multiple argument local min*{$SAD_2(u)$} for u ∈search range, becomes the reliable voters of block 2

and so on, block n: list of motion vector candidates [mv^1_n , mv^2_n , mv^3_n , …, mv^{10}_n] = *multiple argument local min*{$SAD_n(u)$} for u ∈search range, becomes the reliable voters of block n

In this manner, all reliable local MV candidates are preserved and used to build a histogram of MVs. Consequently, it removes sensitive sub-optimizations of each repetition block.

The details of the proposed algorithm are presented in Fig. 4-2. At the beginning of Step 1, the proposed algorithm estimates multiple smallest local minima, or a Motion Vector (MV) candidate set for each block. In order to avoid a local minimum with a relatively large value, the algorithm limits the maximum number of local minima to 10. If the number of the MV candidates is smaller than 10, the algorithm continues searching a local minimum pushing it to the MV candidate set. If 10 MV candidates are in the set and a new local minimum is found, the proposed algorithm compares the local minimum value to the maximum value among the local minima in the MV candidate set, denoted by MAX_VALUE. If the local minimum value is smaller than MAX_VALUE, then the new local minimum is pushed to the MV candidate set. In this manner, all local minima inside the MV candidate set are guaranteed to be the smallest ones. The next step detects whether the block belongs to a repetition pattern region or not, by using the integral projection method

Step 1: Make MV Histogram of MV candidates

1.1. Initialization: All bins in the MV histogram are empty

1.2. Build an MV set (or Top 10 (if enough) smallest local minima) for each block

Core Algorithm

For (each block k)

Initialization: MV Set_k = {Empty}

Loop over search range

1.2.1. Find a local minimum

1.2.2. Push the local minimum into the MV Set_k or Not

```
if (size_of(MV Set_k) < 10)
{
   push the local minimum into the MV Set_k
}
else
{
   find MAX_VALUE = max(local minima in Mv Set_k)
   if (the local minimum < MAX_VALUE)
   {
      remove MAX_VALUE out of MV Set_k
      push the local minimum into the MV Set_k
   }
}
```

Check the block is in a repetition region or not

If (block k is a repetition block)

Push the motion vectors in the MV Set_k into the corresponding bins

Step 2: Choose the representative of the region

- The most frequent MV candidate in the MV Histogram

Figure 4-2 The proposed algorithm

presented in [24]. Finally, a motion vector histogram is generated and then the most frequent motion vector is selected as the representative motion vector for the whole repetition region.

Step 1 exploits the property of multiple local minima in a repetition pattern block while Step 2 represents the global property of a repetition region. In other words, Step 1 improves the reliability of the voting process in Step 2, and consequently, increases the accuracy of the most frequent MV candidate obtained by Step 2. For illustration of the proposed algorithm, an example with five blocks (N = 5) is presented next. Suppose that the MV candidate set for five blocks are obtained as follows:

MV Set_1 = {[-2,-4], [-2, 0]}

MV Set_2 = {[-6,-4], [-2, 0]}

MV Set_3 = {[-2,-4], [-2, 0], [2, 0], [8, 0]}

MV Set_4 = {[-6,-4], [-2, 0], [-2, 2], [4, 2]}

MV Set_5 = {[-2,-2], [-2, 0], [8, 0]}

Then, the histogram of MV candidates are as follows:

MV histogram = {[-6,-4], [-2,-4], [-2,-2], [-2, 0], [-2, 2], [2, 0], [4, 2], [8, 0]}

Corresponding counts: {2, 2, 1, 5, 1, 1, 2}

In this example, the most frequent motion vector is [-2, 0] derived five times in the motion estimations of all the blocks in the repetition region

Additional memory buffers for the proposed algorithm

In step 1.2.1, the proposed algorithm saves eight neighboring SAD values to find a local minimum. In addition, maximum ten motion vector candidates are stored for each block. For the motion vector histogram, the maximum of the number of the MVs in the MV histogram is equal to the size of the search range. In other words, the size of the MV histogram is 33x33 = 1089 MVs. Assuming the raster scan search, the proposed algorithm does not need to save MVs. Instead, it just saves the indices (or positions) of the MVs in the search range because the algorithm can derive the MVs from their indices. The only information that needs to be saved is the frequency count values for the MV candidates in the MV histogram. For a full HD frame with the block size of 8x8, the number of the blocks is 32,400. In the worst case when all blocks belong to a repetition region, the frequency count value can take the value of 32,400, and therefore it requires 15 bits to save each frequency count value, or 2 Bytes. Totally, for all MV candidates, the proposed algorithm requires 2 * 1089 ~ 2 KB, which is relatively small when compared with the memory size to store the original image. Therefore, the proposed method obtains the objective function and satisfy the hardware resource constraints.

4.5. Experimental Results

The performance of the proposed algorithm is shown in Fig. 4-3. From two original frames in Fig. 4-3(a), the algorithm estimates the initial motion vectors of the blocks by using exhaustive full search-based block matching, shown in Fig. 4-3(c) and detects repetition pattern blocks presented in Fig. 4-3(b). The corrected motion vector field by the proposed algorithm is shown in Fig. 4-3(d) and the whole repetition region shares the same motion vector that is the representative motion

vector of the region. It is accurate and equal to the ground truth-value. Therefore, the interpolated frame generated by the corrected motion vector field (see Fig. 4-3 (f)) is clearer than the interpolated frame generated by the noisy motion vector field before correction (see Fig. 4-3(e)).

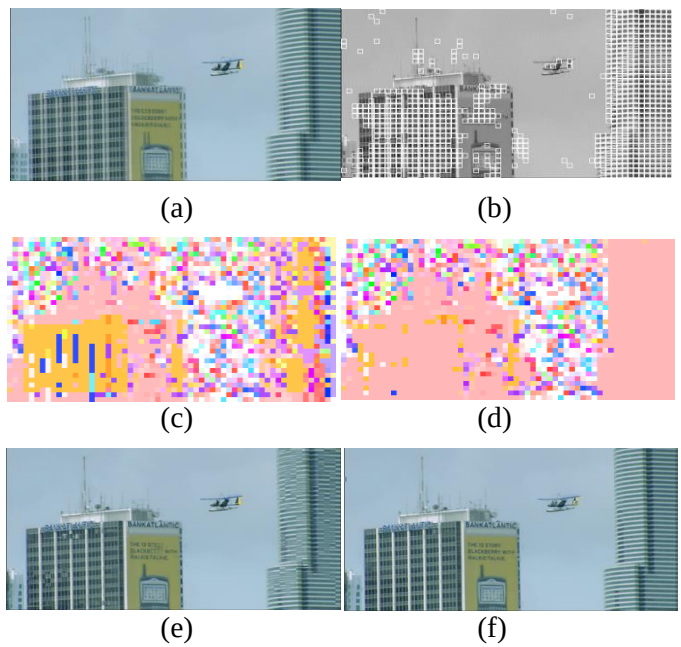


Figure 4-3 An example result of the proposed method

(a) Overlaid original frames, (b) Detected repetition pattern blocks, (c) Initial motion vector field by full search, (d) Corrected motion vector field by the proposed algorithm, (e) Interpolated frame by using (c), (f) Interpolated frame by using (d)

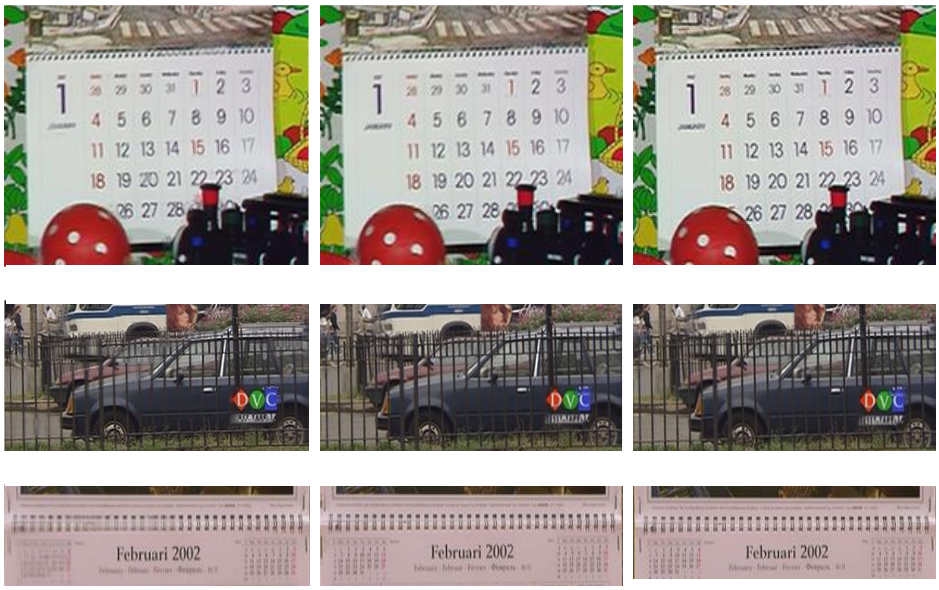
The proposed algorithm is compared with the previous local-based method in [25]. Simulation is conducted with three standard datasets, Bus, Mobile and Calendar sequences which include repetition regions. The PSNR is used as the measurement metric for objective comparison. The simulation result is shown in Table 4.1 which shows the proposed algorithm outperforms the previous method significantly by

around 2.59 dB.

Table 4.1 PSNR comparison.

Test sequence	Local based algorithm [25]		Proposed algorithm
	PSNR (dB)	Δ (dB)	PSNR (dB)
Bus	24.72	2.23	26.95
Mobile	26.16	0.67	26.83
Calendar	28.80	4.86	33.66
Average	26.56	2.59	29.15

Subjective comparisons are presented in Fig. 4-4 in which the left column presents the interpolated frames by the previous method in [25]. The interpolated frames generated by the proposed algorithm are shown in the middle column. The last column corresponds to the ground truth frame. In the previous method in [25], the interpolated frame is blurred and unclear. On the other hand, the proposed algorithm estimates the motion vector of repetition regions accurately, and consequently, generates the output clearer than the previous method does



(a) (b) (c)

Figure 4-4 Subjective comparison between the previous and the proposed algorithms

(a) The previous method [25], (b) The proposed algorithm, (c) Ground truth

First row: Mobile sequence – frame 11, Second row: Bus sequence - frame 33,

Last row: Calendar sequence – frame 301

Chapter 5. Multiple Motion Vectors based Motion Compensation

5.1. Problem Statement

In previous motion compensated frame interpolation algorithms, a Sum of Bilateral Absolute Difference (SBAD) is usually applied as the metric for the selection of motion vector candidates that cross over the same block in the intermediate frame, the motion vector with the minimum SBAD is selected as the final one of the interpolated block. A problem with this method is that it will select wrong motion vector even with the smallest value of SBAD because the minimum SBAD doesn't guarantee that it represents the truth motion vector of the interpolated block. For example, at smoothness areas in images where there exist multiple local minima, it causes many ambiguous motion vectors. In order to reduce the risk of the wrong selection from the SBAD metric, this thesis proposes a new bidirectional motion compensation frame interpolation (MCFI) algorithm that contains a new metric and a novel non-selective approach that preserves motion vector information from all overlapped projected blocks. Firstly, forward and backward motion vector fields are projected into the interpolated frame in order to generate bi-directional motion vectors of the interpolated blocks. The proposed method preserves all motion vectors of overlapped projected blocks for each interpolated block. And an adaptive weighted motion compensation is done for interpolated blocks correspond to their own preserved motion vectors. The weighted coefficients are computed by using a comprehensive metric that composes of distance or overlap area, matching cost and

smoothness cost correspond to the preserved motion vectors. Holes are filled by vector median filter of the motion vector of non-hole neighbor blocks.

5.2. Adaptive Weighted Multiple Motion Vectors based Motion Compensation

5.2.1. One-to-Multiple Motion Vector Projection

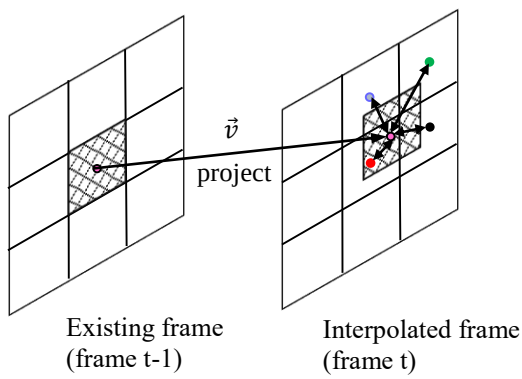
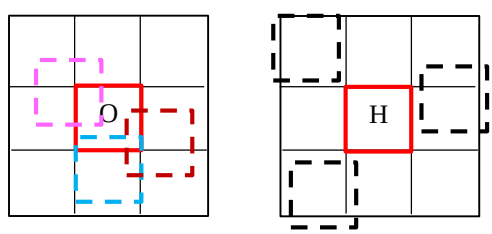


Figure 5-1 Motion Vector Projection

Fig. 5-1 shows an example of the forward motion vector projection. The projected block is, in general, not aligned with the interpolated block as shown in the figure. The shaded block in the center of the frame t – 1 represents the original block to be projected whereas the shaded block at frame t represents the projected block. Note that this projected block is overlapped with up to four blocks in the interpolated frame. In conventional methods, only the interpolated block that is nearest to (or most overlapped with) the projected block will takes the motion vector of the projected block, there is no motion vectors for the other overlapped interpolated blocks. In other words, the conventional methods do winner take it all. In this context,

that is called as a one-to-one projection. The proposed method in this thesis will do one-to-multiple projection. In other words, all overlapped interpolated blocks will share the same motion vector of the projected block. Consequently, it alleviates blocking artifact between the interpolated blocks. In addition, it also reduces the possibility of hole blocks that are blocks without motion vector in the interpolated frame.

After accumulating all projected blocks, in the case when an interpolated block (block in interpolated frame) has multiple projected blocks, conventional methods usually select one block (the best according to a selection metric) among multiple candidates. This selection loses motion vector information of un-selected ones that may have a chance to make a better motion compensation in their overlap areas. In addition, the selected projected block just covers its overlap area within the interpolated block, and consequently, the non-overlapped area that belongs to other projected blocks may be interpolated with wrong interpolated pixels, as shown in Fig. 5-2 (a).



(a) (b)

Figure 5-2 Projected blocks of a interpolated block.

(a) An example of multiple-projected blocks of the interpolated block, O

(Overllaped block)

(b) An example of non-projected block of the interpolated block, H (Hole block)

In Fig. 5-2(a), the red center block is the interpolated block with multiple-projected blocks (other color blocks). In Fig. 5-2(b), the red center block (or Hole block) is the interpolated block without any projected blocks (surrounding black blocks).

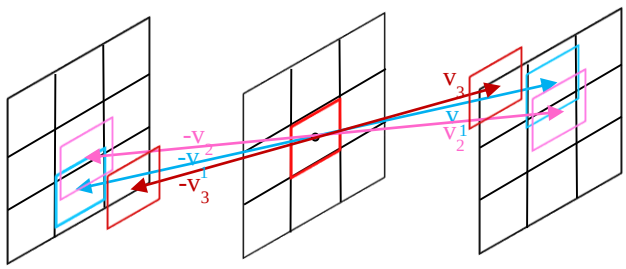


Figure 5-3 An example of the adaptive weighted mutiple motion vector based MC.

This thesis proposes a new motion compensation that uses all the motion vectors of the projected blocks that are overlapped with the target interpolated block. The contribution of each motion vector depends on the relevance to the interpolated block; that is evaluated by the measure metric to be discussed in the next subsection. As a result, a weighted sum of the interpolated pixels is the final interpolated pixel as follows:

$$f_t(x) = \frac{1}{2} \sum_{i=1}^{N} \text{Normalized weight}(\vec{v}_i) * \{f_{t-1}(x - \vec{v}_i) + f_{t+1}(x + \vec{v}_i)\} \quad (5.1)$$

where $\vec{v}_i$ is the bi-directional motion vector of the interpolated block, N is number of bi-directional motion vectors of the interpolated block. Normalized weight($\vec{v}_i$) is computed as Eq. 5.6 explained in the next subsection. Fig. 5-3 shows an example of the proposed MC. In this example, the interpolated block has three projected blocks, corresponding to three motion vector candidates v_1, v_2 and v_3. With each candidate, the motion compensation step generates individual interpolated pixels. The final

interpolated pixels are a weighted sum of three above interpolated pixels.

5.2.2. A Comprehensive Metric as the Extension of Distance

For each overlapped projected block, its distance to the interpolated block is defined as the displacement between their center points:

$$\mathbf{distance}(\vec{v}) = \left\| \boldsymbol{pos}_{\boldsymbol{projected-block}} - \boldsymbol{pos}_{\boldsymbol{interpolated-block}} \right\| \quad (5.2)$$

where $pos_{projected\text{-}block}$, and $pos_{interpolated\text{-}block}$ are the center positions of the projected block, and the interpolated block, respectively. This distance decreases as the size of the overlapped area increases. The motion vector $\vec{v}$ is not always correct, it depends on the accuracy of motion estimation algorithm, in case of a wrong motion vector $\vec{v}$, it can cause the wrong projected block even if the distance value is small. Therefore, only the distance information is not enough to find the best-projected block for the corresponding interpolated block. We need a more comprehensive metric that contains distance as the key component as well as other terms to cover reliability, smoothness of the motion vector fields.

A comprehensive metric consists of distance, matching cost and smoothness cost as follows:

$$\text{METRIC}(\vec{\boldsymbol{v}}) = \text{distance}(\vec{\boldsymbol{v}}) + k_1 * \text{cost}(\vec{\boldsymbol{v}}) \quad (5.3)$$

$$\text{where cost}(\vec{v}) = \text{SAD}(\vec{v}) + k_2 * smoothness(\vec{v})$$

$$= \mathbf{SAD}(\vec{\boldsymbol{v}}) + \boldsymbol{k_2} * \frac{\mathbf{1}}{\mathbf{4}} \sum_{\boldsymbol{i=1}}^{\mathbf{4}} \|\vec{\boldsymbol{v}} - \vec{\boldsymbol{v}}_{\boldsymbol{i}}\| \quad (5.4)$$

$\text{cost}(\vec{v})$ is the total cost that presents the reliability and smoothness of the estimated

motion vector. SAD($\vec{v}$) is Sum of Absolute Difference between each pixel in the original block and the corresponding pixel in the matching block that corresponds to the motion vector $\vec{v}$, $\vec{v}_i$ is neighbor motion vectors of $\vec{v}$, $\vec{v}$ is the estimated motion vector of original block, k_1 and k_2 are normalization parameters. The proposed metric is a comprehensive one that contains distance that presents accuracy of block tracking, matching cost (SAD) shows the reliability of the block tracking and smoothness cost preserves smoothness constraint for true motion vector field. The smaller the metric is the better estimated motion vector is. In other words, the motion vector candidate with a small value of the metric will take more contribution than the one with a large value of the metric. The contribution of each candidate is represented by a weighted coefficient that is an inversion of the metric as shown in Eq. 5.5. Due to the coefficients computed by Eq. 5.5 can get the value outside the interval of [0, 1] therefore they should be normalized to take a value inside the range between 0 and 1 as shown in Eq. 5.6.

$$\text{weight}(\vec{v}_i) = 1 / \text{METRIC}(\vec{v}_i) \qquad (5.5)$$

$$\text{Normalized weight}(\vec{v}_i) = \frac{\text{weight}(\vec{v}_i)}{\sum_i \text{weight}(\vec{v}_i)} \qquad (5.6)$$

5.3. Handling Hole Blocks

The hole block, the interpolated block has no projected block as shown in Fig. 5-2(b), will be filled by a vector median filter of non-hole neighboring blocks as shown in Fig. 5-4, where v_1, v_2, v_3 and v_4 are available up, left, right and down neighbor non-hole blocks of the current hole block. The number of the available neighbor non-hole blocks is up to four because, in large hole areas, some of them are not always

available.

$$v_{hole} = arg \min_{v_j} \sum_{i=1}^{N} \| v_j - v_i \| \qquad (5.7)$$

where N is number of available non-hole blocks.

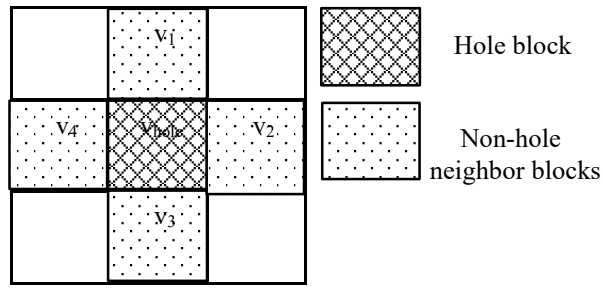


Figure 5-4 Hole blocks handling.

5.4. Framework of the Proposed Motion Compensated Frame Interpolation

The proposed motion compensation method shown in Fig. 5-5 consists of following steps. In the first step, the proposed method projects forward and backward motion vector fields obtained from unidirectional motion estimations into the interpolated frame. Next, for each projected block, compute a distance between it and its overlapped interpolated blocks, and then a comprehensive metric as the extension of the distance shown in section B is computed by the combination of distance, matching cost and smoothness cost. If an interpolated block has multiple overlapped projected blocks, a non-selection adaptive-weighted multiple motion vector - motion compensation is implemented with the weighted coefficients are computed as the inversion of the comprehensive metric obtained in the previous step. In case an interpolated block has no overlapped projected block, a vector median

filter is applied to its available non-hole neighbor blocks to generate the motion vector of the hole block and do conventional bi-directional motion compensation.

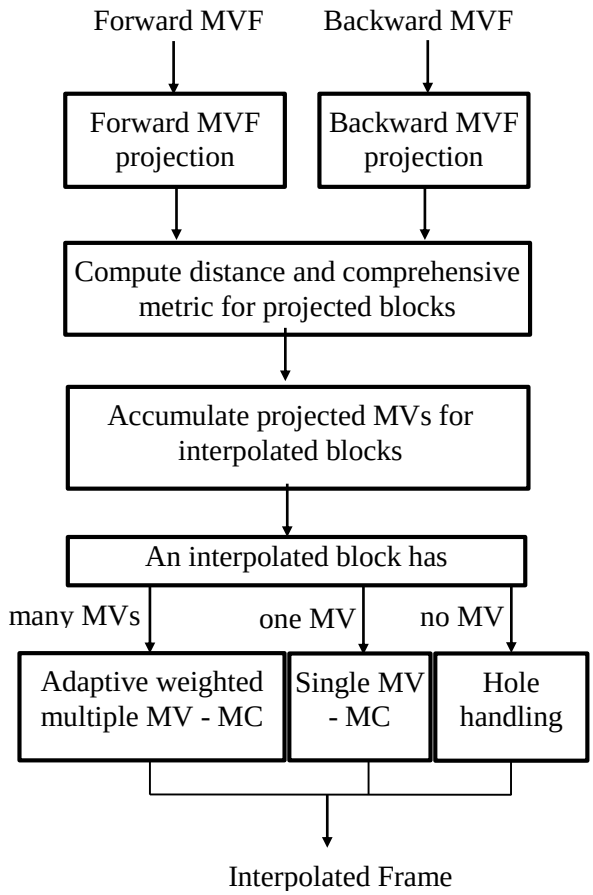


Figure 5-5 The proposed motion compensation method diagram

5.5. Experimental Results

To demonstrate the performance of the proposed method, we use nine test sequences, which are in the standard CIF (325x288) format and 30 frame/s. They are Bus, City, Flower, Football, Foreman, Mobile, Mother & daughter, Soccer and Stefan. The odd frames are removed and the new odd frames are generated from the

even frames using MC-FRUC algorithms. We compare our method with two conventional MC-FRUC methods, they are Dual ME [35] and Yoo [38]. Dual ME is representative for bi-directional ME approach, meanwhile Yoo method is representative of hybrid ME approach. In all experiments, the block size is set to 8x8, the search range is ± 16 for both horizontal and vertical directions.

From table 5.1, the proposed method provides about 0.60 - 1.03 dB higher average PSNR performance than the conventional algorithms. Specially. The performance gaps between the proposed algorithm and the conventional algorithms are high especially on the Bus, City, and Mobile sequences, which have smoothness areas that motion vectors obtained ME step exist some ambiguous ones, therefore when those wrong motion vectors are chosen, visual artifacts are produced. Meanwhile, the proposed method uses multiple motion vectors for non-selective adaptive weighted motion compensation with the weighted coefficients are computed by the comprehensive metric, therefore it mitigates the effect of the wrong motion vectors.

Table 5.1 PSNR comparisons between the proposed method and conventional algorithms

Sequence	Dual ME		Yoo method		Proposed
	PSNR (dB)	△ (dB)	PSNR (dB)	△ (dB)	PSNR (dB)
Bus	25.02	2.19	25.71	1.5	27.21
City	31.94	1.71	32.64	1.01	33.65
Football	22.78	0.23	22.79	0.22	23.01
Foreman	32.37	0.16	32.29	0.24	32.53
Garden	28.75	1.23	29.58	0.4	29.98
Mobile	25.33	1.62	25.68	1.27	26.95
Mother	40.81	0.06	40.92	-0.05	40.87
Soccer	25.96	0.56	26.26	0.26	26.52
Stefan	27.03	1.5	27.99	0.54	28.53
Average	28.89	1.03	29.32	0.60	29.92

Fig. 5-6 and Fig. 5-7 show subjective visual comparisons between the proposed method with previous ones. In the first row, the first column presents the interpolated frames by Dual ME [35] method, the second column shows the interpolated frames by Yoo [38] method and in the second row, the first column shows the ones generated by proposed method, the second column presents the ground truth frames. Previous methods fail at smoothness regions such as texts and numbers in the advertisement board in Fig. 5-6 and in calendar area in Fig. 5-7, meanwhile, the proposed method gives clean interpolated frames, owing to its adaptive weighted multiple motion vectors based MC.

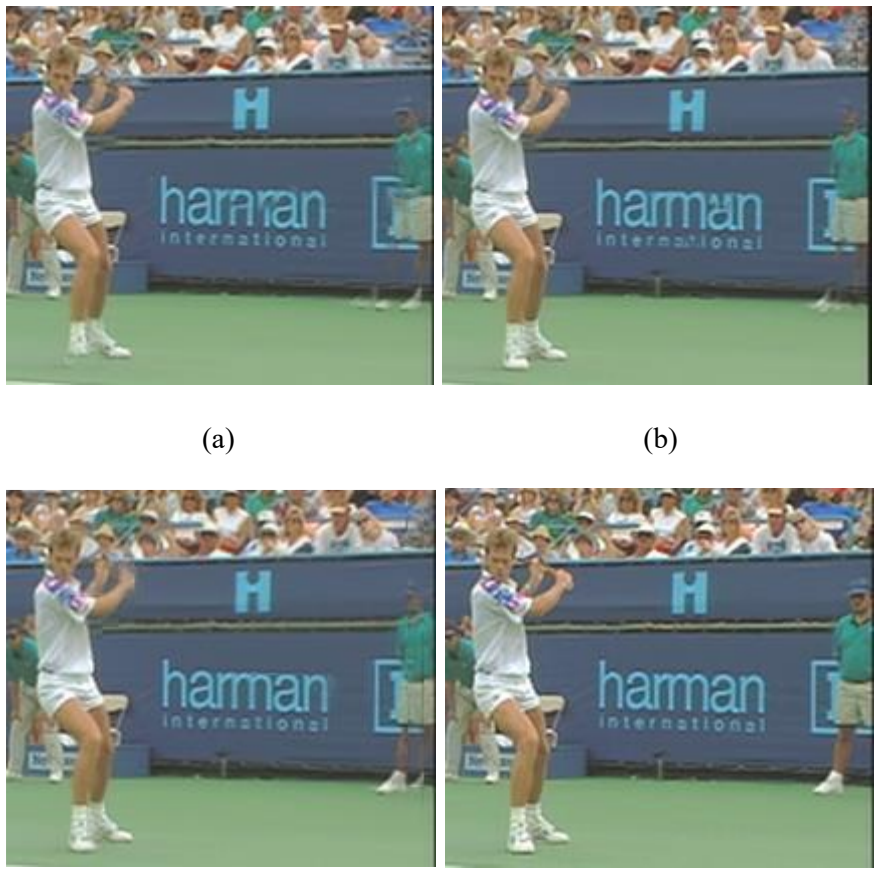


Figure 5-6 Interpolated frames by previous methods and the proposed method on Stefan dataset

(a) Dual ME [35], (b) Yoo [38], (c) The proposed, (d) Ground truth

(a) (b)

(c) (d)

Figure 5-7 Interpolated frames by previous methods and the proposed method on Mobile dataset

(a) Dual ME [35], (b) Yoo [38], (c) The proposed, (d) Ground truth

Chapter 6. Video Frame Interpolation with a Stack of Deep CNN

6.1. Problem Statement

In previous CNN based methods, the objective function or loss function only focuses on pixel difference. Consequently, it usually fails in the estimation of fast and/or complex movement which requires a critical role of motion estimation for high-quality frame interpolation. This thesis presents a comprehensive framework that glue two above previous approaches into a single stacked network such that an analysis-by-synthesis technique is used to estimate bidirectional intermediate optical flows and later a synthesis network glues intermediate results generated by component branches (an optical flow based branch and a CNN kernel based synthesis branch) to synthesize the very end intermediate frame. The primary contributions of the proposed method are summarized as follows. Firstly, the proposed network is a bridge between two branches of approaches: optical flow based frame interpolation and CNN kernels based frame synthesis. Secondly, the thesis introduces a method to derive directly Intermediate optical flows that are the flows from the intermediate frame to two original frames. This module contributes to learning processes for both frame synthesis networks. It glues a motion-ness into the pixel matching loss for the first CNN kernels based synthesis network and it drives the second synthesis network with estimated optical flows. Thirdly, the proposed network is a back-to-back stack of two network layers such that the first network layer generates three input components for the second network layer that is

an extended version of SepConv network [65]. Lastly, the proposed method outperforms the previous algorithms for various datasets.

6.2. The Proposed Network for Video Frame Interpolation

6.2.1. A Stack of Synthesis Networks

Analysis by a synthesis technique is the key component of the proposed network that stacks two synthesis networks together, a back-to-back stack to help each other in learning operation. Consequently, it covers both the spatial property of CNN kernels based synthesis and the temporal property of optical flow based synthesis. In addition, it also narrows down the displacement between input frames and the final intermediate frame for more condense synthesis.

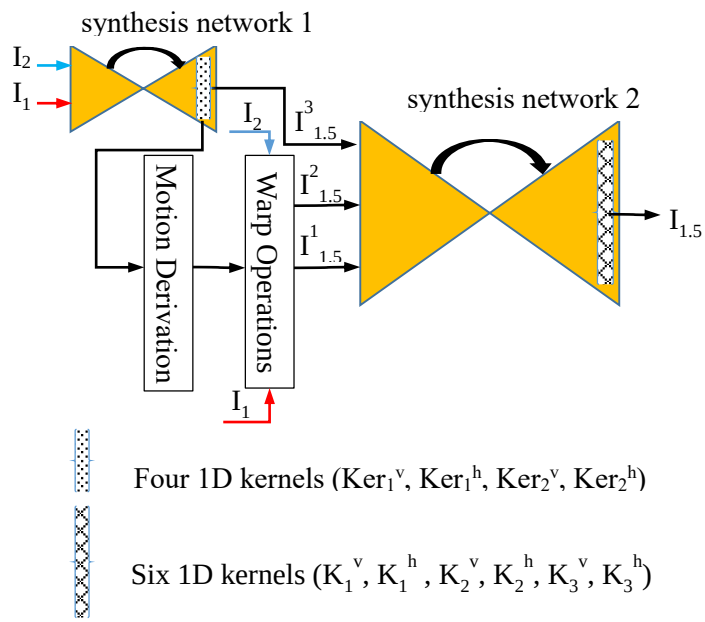


Figure 6-1 Architecture of the proposed network

The proposed network shown in Figure 6-1, is a back-to-back stack of two network layers. In the first layer, a frame synthesis network generates the very first intermediate frame. In addition, as a byproduct of the first synthesis network, four 1D kernels that encode implicitly the motion information are used to derive intermediate optical flows by Motion Derivation module. Then, two original input frames are warped to the intermediate time scale using the estimated intermediate optical flows. Finally, three intermediate interpolated frames are stacked together to feed into the second synthesis network that is a variant of the first one. The stack of networks is used to narrow down the distance between input frames to estimate condensed interpolation kernels. As shown in Figure 6-2, among three intermediate interpolated frames, in term of time scale, the output of the first synthesis network, denoted as $I^3_{1.5}$ is the nearest to the real output target frame, denoted as $I_{1.5}$. On the other hand, the frame, denoted as $I^1_{1.5}$ that is the warped frame from the first original frame (I_1), is slightly offset to the left side of the real output target frame, and the frame, denoted as $I^2_{1.5}$ that is the warped frame from the second original frame, (I_2), is slightly offset to the right side of the real output target frame. For illustration of the timescale of intermediate frames, Figure 6-3 shows an example of a time scale of frames, at the top row, from left to right respectively are the first original frame, the ground truth intermediate frame, and the second original frame.

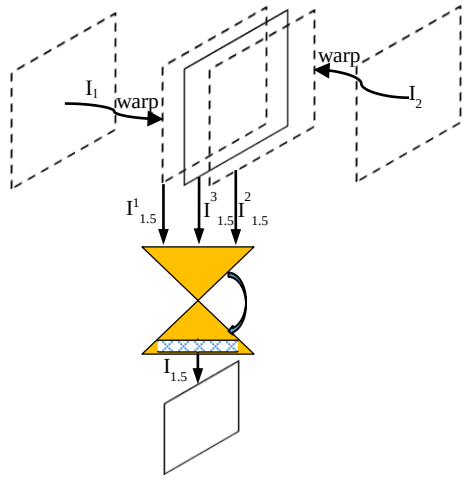


Figure 6-2 The structure of the second synthesis network

In the second row, from left to right respectively are the corresponding intermediate interpolated frames, $I^1_{1.5}$ the warped frame from frame 1, $I^3_{1.5}$ the very first intermediate frame generated by the first synthesis network, and $I^2_{1.5}$ the warped frame from frame 2. The last rows show the corresponding displacements between three above intermediate results and the very end interpolated frame. From images, we can see that among three intermediate interpolated frames, frame $I^3_{1.5}$ is the closest to the target ground-truth frame meanwhile $I^1_{1.5}$ and $I^2_{1.5}$ frames still exist short displacements to the target ground-truth frame. This re-assures our observation.

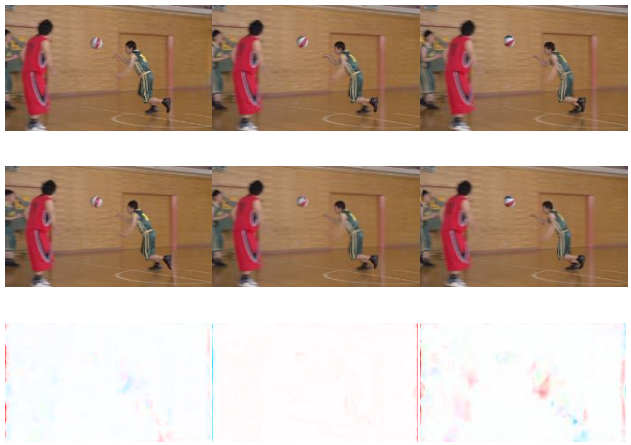


Figure 6-3 An example for the time scales of intermediate interpolated frames

As shown in Figure 6-2, the second layer of the stack is the extended version of the first synthesis network. Originally, the first synthesis network has two input frames and four 1D kernels to convolve with the two original frames. The extended network has three input intermediate frames, and therefore, it has six corresponding 1D kernels. The second synthesis network learns from the closest frames to synthesize the final intermediate frame, and it also embraces both optical flow based results and a CNN based synthesized frame. Consequently, it can cover challenging motion scenarios, such as fast and complex movements.

6.2.2. Intermediate Optical Flow Derivation Module

In the first layer of the stack, the motion derivation module is the glue between two branches of approaches, the optical flow based frame interpolation and the CNN kernels based frame synthesis. This makes a chicken-egg problem solved by training both blended tasks such that the intermediate optical flows, as denoted in Figure 6-4, are estimated by the analysis-by-synthesis technique through convolution kernels

of the first synthesis network. Meanwhile the estimated optical flows role as motion-ness in the loss function of the first synthesis network makes the network learn only pixel matching also motion constraints and scenarios. In addition, estimating the optical flows from the synthesized intermediate frame is a target-based estimation that can fix estimation errors from the previous methods when the intermediate frame is unavailable to verify the accuracy of analysis. In other direction, the estimated intermediate flows are derived from 1D kernels of the synthesis network 1. Consequently, it glues the motion constrains into network 1. Therefore, network 1 learns not only pixel matching but also motion information.

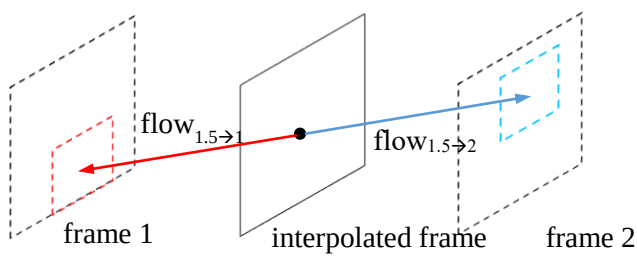


Figure 6-4 Bi-directional intermediate flows

The coefficients of 1D kernels implicate motion information and they are exploited to derive the flow information. The motions are encoded as the offsets of the non-zero kernel values to the kernel center. The motion vector is the weighted sum of the offsets. Therefore, the values of the coefficients and the offsets are used in order to compute the motions. There are four 1D kernels, two corresponding to the displacement of frame 1 to the interpolated frame, and the others corresponding to the displacement of frame 2 to the interpolated frame. The optical flows for both the forward and backward directions with a point of view from the intermediate

frame are computed directly. The formulations of the motion derivation module are represented by the set of equations (6.1), (6.2), (6.3) and (6.4).

$$u_{1.5\rightarrow1} = \frac{\sum_{i=1}^{51} weight_i^{h_1} * offset_i^{h_1}}{\sum_{i=1}^{51} weight_i^{h_1}} \quad (6.1)$$

$$v_{1.5\rightarrow1} = \frac{\sum_{i=1}^{51} weight_i^{v_1} * offset_i^{v_1}}{\sum_{i=1}^{51} weight_i^{v_1}} \quad (6.2)$$

$$u_{1.5\rightarrow2} = \frac{\sum_{i=1}^{51} weight_i^{h_2} * offset_i^{h_2}}{\sum_{i=1}^{51} weight_i^{h_2}} \quad (6.3)$$

$$v_{1.5\rightarrow2} = \frac{\sum_{i=1}^{51} weight_i^{v} * offset_i^{v_2}}{\sum_{i=1}^{51} weight_i^{v_2}} \quad (6.4)$$

where $u_{1.5\rightarrow1}$ and $v_{1.5\rightarrow1}$ are the horizontal and vertical components of the flow from the intermediate frame to frame 1, $u_{1.5\rightarrow2}$ and $v_{1.5\rightarrow2}$ are the horizontal and vertical components of the flow from the intermediate frame to frame 2. $offset_i^{h1}$, $offset_i^{v1}$, $offset_i^{h2}$, $offset_i^{v2}$ are the displacements of the coefficients to the center position in the corresponding 1D kernels.

6.2.3. Warping Operations

Guided by the estimated optical flow, the proposed method warps the input frames into the intermediate timescale. Both forward and backward warping functions, which can be implemented using bilinear interpolation are differentiable. Specifically, the proposed method employs forward warping that uses the estimated optical flow to warp the input frame I_1 to the target locations in the intermediate frame and obtains a warped frame $I^1_{1.5}$. The proposed method warps the input frame I_2 and generates a warped frame $I^2_{1.5}$ in the same way by using backward warping.

Two warped frames are very close to the true interpolated frame. Therefore, they are very suitable for the inputs of the synthesis network 2 that works as a frame refinement to generate the final intermediate frame. This step narrows down the distances between two consecutive input frames and the intermediate one. In addition, it is easier for the network to learn kernels when two inputs are closer.

6.2.4. Training and Loss Function

The proposed network is a stack of component subnets, as suggested by [45], [69], in order to avoid over-fitting, the proposed network is trained end-to-end with a loss function that contain the final loss and two intermediate losses, respectively, as the order given in equation (6.5).

$$\textit{Loss function} = \| I_{1.5} - I_{gt} \|_1 + \| I^3_{1.5} - I_{gt} \|_1 + \| I^w_{1.5} - I_{gt} \|_1 \qquad (6.5)$$

where I_{gt} is the ground truth frame, $I^w_{1.5} = (I^1_{1.5} + I^2_{1.5}) / 2.0$ represents for the warped intermediate frames, $I^1_{1.5}$ and $I^2_{1.5}$ obtained by both forward and backward warping operations. Following [55], [65] the proposed neural network parameters are initialized by a convolution aware initialization [61] and trained by using AdaMax [56] with $\beta 1 = 0.9$, $\beta 2 = 0.999$, a learning rate of 0.001 and a mini-batch size of 12 samples.

The training dataset provided by [53] is used to train the proposed network because this dataset contains high-quality frames extracted from high-resolution videos downloaded from vimeo.com. The resolution of training videos is 448x256. For data augmentation during the training process, the trainer randomly swaps the temporal order between input frames, frame1 becomes frame2 and vice versa. This makes

dataset larger and eliminates potential priors. Pytorch library is used to train the proposed network with two NVIDIA GTX 1080 GPUs.

6.2.5. Network Architecture

The first synthesis network is the same as Sepconv network in [55] in which two inputs are the original frames and outputs are four 1D kernels convoluted with the original frames to generate the very first intermediate frame. The second synthesis network is an extended version of the first synthesis network with the inputs are three intermediate interpolated frames therefore, six 1D kernels are trained to generate the output pixel of the final intermediate frame as the following equation.

$$I_{1.5}(x,y) = K_1^v(x,y) * K_1^h(x,y) * P_{1.5}^1(x,y) + K_2^v(x,y) * K_2^h(x,y) * P_{1.5}^2(x,y) + K_3^v(x,y) * K_3^h(x,y) * P_{1.5}^3(x,y) \quad (6.6)$$

where $P^1_{1.5}(x, y)$, $P^2_{1.5}(x, y)$, and $P^3_{1.5}(x, y)$ respectively are the patches centered at (x, y) position in intermediate interpolated frames $I^1_{1.5}$, $I^2_{1.5}$ and $I^3_{1.5}$. K^v_1, K^h_1, K^v_2, K^h_2, K^v_3, and K^h_3 are the learned pixel-dependent 1D kernels of the second synthesis network.

6.2.6. Experimental Results

6.2.6.1. Frame Interpolation Evaluation

To evaluate the proposed network, quantitative and qualitative comparisons with several representative state-of-the-art video frame interpolation and optical flow

methods are made. Firstly, methods are evaluated with the interpolation category of Middlebury optical flow benchmark that is typically used for assessing frame interpolation methods [53]. The proposed approach is compared with the methods that rank high with this interpolation benchmark. The first one is MDP-Flow2 [59], an accurate optical flow method, as it still remains the highest rank among all classic optical flow methods with the Middlebury benchmark. In addition, PWC method [67] that is a state-of-the-art CNN based optical flow algorithm that performs top among CNN based methods ranked with well-known Sintel optical flow benchmark [43]. To synthesize interpolated frames from the computed optical flows, the same algorithm in [53] is used. For a CNN based frame synthesis algorithm without optical flow estimation, recent SepConv [55] method is chosen owing to its high performance among CNN based algorithms. The optical flow that is the byproduct of the proposed network is also compared to state-of-the-art methods.

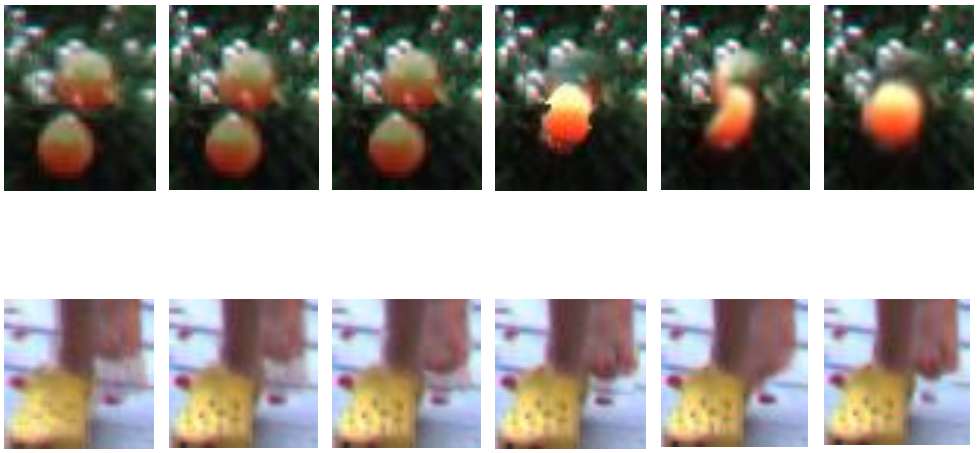
Table 6.1 shows the average interpolation error (AIE) used in [53] where the interpolation error is the root-mean-square (RMS) difference between the ground-truth image and the estimated interpolated image. The proposed network outperforms state-of-art methods and improves the best previous method by a significant margin (9.5%). Especially with *Backyard*, *Basketball*, *Dumptruck* and *Evergreen* datasets which show real-world scenes, captured with a real camera and containing real sources of noise, the proposed network is consistently the best by notable margins. The proposed interpolation method, denoted as InterpCNN, is ranked 3nd in Interpolation Error (with Average statistic) and ranked 1st in Interpolation Error (with Standard Derivation (SD) statistic) among over 150 algorithms listed in the benchmark website at the submission time. For visual

evaluation, Figure 6-5 shows the proposed interpolated frame that is a clear result and alleviates ghost and distorted artifacts whereas they still appear in the interpolated frames generated by the previous algorithms

Table 6.1 Objective comparisons on Middlebury benchmark

	Ave.	Meq.	Sch.	Urb.	Ted.	Bac.	Bas.	Du.	Eve.
Proposed	**4.78**	2.61	3.30	3.14	4.74	**8.11**	**4.48**	**5.78**	**6.06**
CtxSyn	5.28	**2.24**	**2.96**	4.32	**4.21**	9.59	5.22	7.02	6.66
MDP-Flow2	5.83	2.89	3.47	3.66	5.20	10.2	6.13	7.36	7.75
SuperSlomo	5.31	2.51	3.66	**2.91**	5.05	9.56	5.37	6.69	6.73
SepConv	5.61	2.52	3.56	4.17	5.41	10.2	5.47	6.88	6.63
DeepFlow	5.97	2.98	3.88	3.62	5.39	11.0	5.91	7.14	7.80

Note: Ave. = Average; Meq. = Mequon; Sch. = Schefflera; Urb. = Urban; Ted. = Teddy; Bac. = Backyard; Bas. = Basketball; Du. = Dumptruck; Eve. = Evergreen



(a) (b) (c) (d) (e) (f)

Figure 6-5 Visual comparisons on Backyard sequence on Middlebury benchmark.

(a) Overlaid, (b) SpyNet, (c) PWC-Net, (d) MDP-Flow2, (e) SepConv, (f) The proposed

Table 6.2 Objective comparisons on Vimeo90K dataset among CNN based methods

	PSNR	SSIM
ToFlow	33.53	0.9668
ToFlow+mask	33.73	0.9682
SepConv	33.85	0.9697
Proposed	34.65	0.9737

The next well-known dataset for evaluating video frame interpolation algorithms is Vimeo90K dataset provided by [63]. It contains 3,782 triplets of frames with the image resolution of 448×256 pixels. As shown in Table 6.2, the proposed method outperforms previous CNN based networks, SepConv [55], ToFlow [62] and its

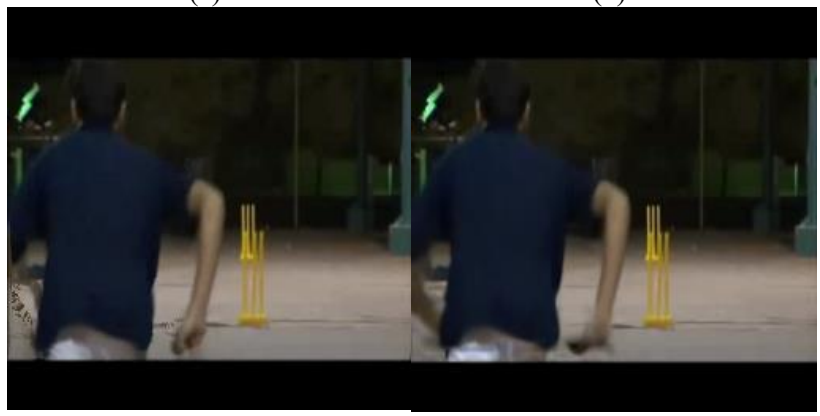
variant, ToFlow with a mask [62] by significant margins in term of both peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) [44].

Table 6.3 Objective comparisons on UCF101 dataset

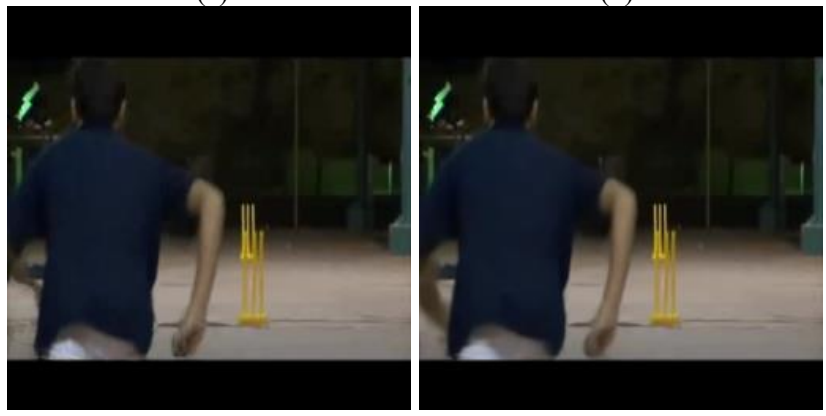
	PSNR	SSIM
Sluper-Slomo [58]	33.14	0.9519
PWC-Net [67]	33.76	0.9618
MDP-Flow2 [59]	34.52	0.9660
DVF [57]	34.12	0.9631
SepConv [55]	34.78	0.9669
Proposed	34.96	0.9683

UCF101 dataset [68] consists of videos with the size of 256x256. This dataset is initially used to evaluate activity recognition and later it is used to evaluate the frame interpolation originated from [57]. UCF101 dataset includes videos with small motion. Therefore, even with a simple interpolation algorithm such as frame average, the video quality of an interpolated frame is sufficiently high as shown in Table 6.3. In this dataset, the proposed network also outperforms other previous methods.

(a) (b)



(c) (d)



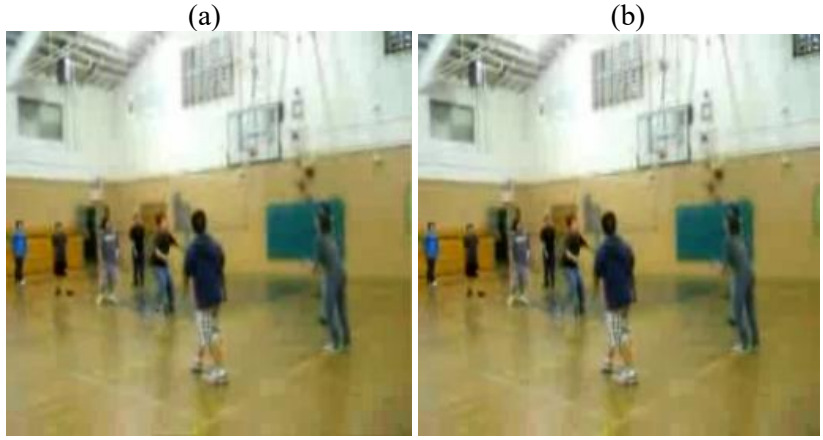
(e) (f)

(g)

Figure 6-6 Subjective visual quality comparison on UCF101 dataset (1)

(a) PWC-Net, (b) DVF, (c) MDP-Flow2 , (d) SepConv, (e) Super-Slomo, (f) The proposed, (g) Ground truth

(a) (b)



(c) (d)

(e) (f)

(g)

Figure 6-7 Subjective visual quality comparison on UCF101 dataset (2)

(a) PWC-Net, (b) DVF, (c) MDP-Flow2, (d) SepConv, (e) Super-Slomo, (f) The proposed, (g) Ground-truth

The last one is a new dataset proposed in this thesis to cover the difficult cases for frame interpolation. These cases include the movement of text objects, occlusion, reveal, and complex movements of small and fast-moving objects. Movement of text objects as a subtitle and logos is difficult for interpolation because the movement often takes place in a background while its motion is in a different direction from the background. Object occlusion and reveal are difficult in a classical computer vision

problem such as optical flow and they are also difficult in frame interpolation. A small object is difficult to estimate its motion and so is fast and complex movement. This new dataset is used to measure the performance of frame interpolation algorithms that focus on enhancement of visual quality. For explanation, this new data set is called Hard Cases for Display (HCD) which consists of four video sequences, each contains 60 frames with the resolution of 864x480, except the Basketball sequence that contains 90 frames with the resolution of 416x240. The dataset covers hard and challenging cases for frame interpolations such as scenes with sub-title, occlusion and reveals, fast complex motions, and the movement of small objects. Table 6.4 shows quantitative comparisons between the proposed methods with representative state-of-the-art methods on HCD dataset. In both PSNR and SSIM, the proposed method outperforms state-of-art methods with notable margins. Figure 6-8, 6-9 and 6-10 show the interpolated frames for visual quality comparisons on HCD dataset. In a fast and complex motion sequence, as shown in the Figure 6-8, the movement of the leg of the soccer player and that of the hand of the goalkeeper is fast and complex. The proposed method improves significantly visual quality in comparison with the previous methods. The Figure 6-9 shows the interpolated frames for the subtitle sequence where the text objects in the subtitle region include artifacts. The previous methods based on optical flow estimations, and CNN kernel based SepConv, suffer from these artifacts whereas the proposed method successfully removes them. For small objects such as balls in the basketball sequence shown in the Figure 6-10, the proposed network alleviates ghost artifacts significantly when compared with the previous methods.

Table 6.4 Objective comparison on HCD dataset

Test sequence	PSNR (dB)			
	MDP-Flow2	PWC-Net	SepConv	Proposed
Subtitle	31.83	24.82	33.83	34.73
Occlusion	29.81	28.35	30.92	32.29
Soccer	29.38	28.01	29.79	31.04
Basketball	34.36	31.11	34.84	36.14
Average	31.35	28.07	32.35	33.55
Test sequence	SSIM			
	MDP-Flow2	PWC-Net	SepConv	Proposed
Subtitle	0.9913	0.9661	0.9924	0.9929
Occlusion	0.9555	0.9476	0.9622	0.9706
Soccer	0.9599	0.9479	0.9636	0.9702
Basketball	0.9867	0.9720	0.9876	0.9902
Average	0.9734	0.9584	0.9765	0.9810

(a)

(b) (c)

(d) (e)

Figure 6-8 Visual comparison of interpolated frames on soccer sequence of HCD dataset

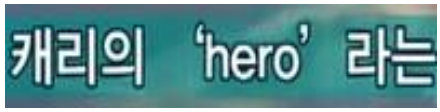
(a) Ground truth, (b) MDP-Flow2, (c) PWC-Net, (d) SepConv, (e) The proposed

(a)

(b)

(c)

(d)



(e)

Figure 6-9 Visual comparison of interpolated frames on subtitle sequence of HCD dataset

(a) Ground truth, (b) MDP-Flow2, (c) PWC-Net, (d) SepConv, (e) The proposed

(a)

(b) (c)

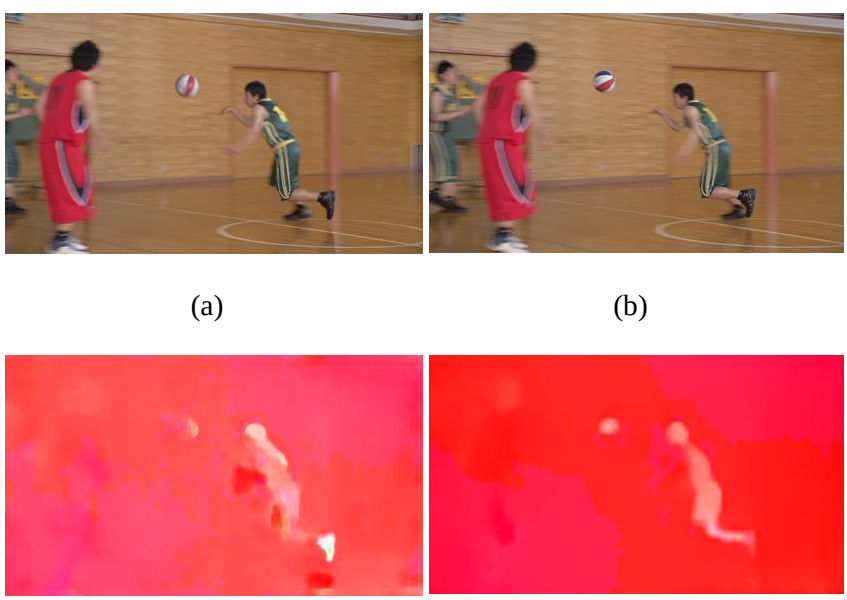
(d) (e)

Figure 6-10 Visual comparison of interpolated frames on basketball sequence of HCD dataset

(a) Ground truth, (b) MDP-Flow2, (c) PWC-Net, (d) SepConv, (e) The proposed

6.2.6.2. Ablation Experiments

Optical flow evaluation. Figure 6-11 shows comparisons between the estimated optical flow by the proposed method and the results obtained by state-of-the-art optical flow methods including MDP-Flow2 [59] (the top-ranked in the Middlebury benchmark) and recent CNN based flow networks, SPyNet [70] and PWC-Net [67]. The top row shows the estimated optical flow results, and the bottom row is the corresponding interpolated frame generated by the above flows by using the same frame interpolation algorithm [53]. The proposed analysis-by-synthesis based motion derivation module estimates the movement of the rotating and moving balls accurately. The results prove that the proposed method preserves the motion of small objects such as balls meanwhile others fail to estimate the movement of the ball. Consequently, either two balls or a distorted ball artifact appears in the interpolated frames generated by the previous methods.



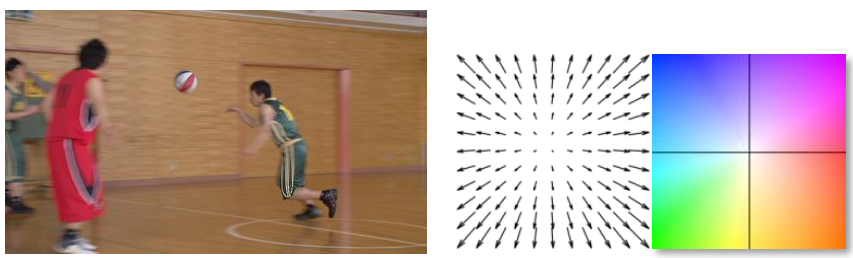
(a) (b)

(c) (d)

(e) (f)

(g) (h)

(i) (k)



(l) (m)

Figure 6-11 Visual optical flow results on basketball sequence.

(a) frame 1, (b) frame 2, (c) color encoded optical flow by SpyNet, (d) color encoded optical flow by PWC-Net, (e) color encoded optical flow by MDP-Flow2, (f) color

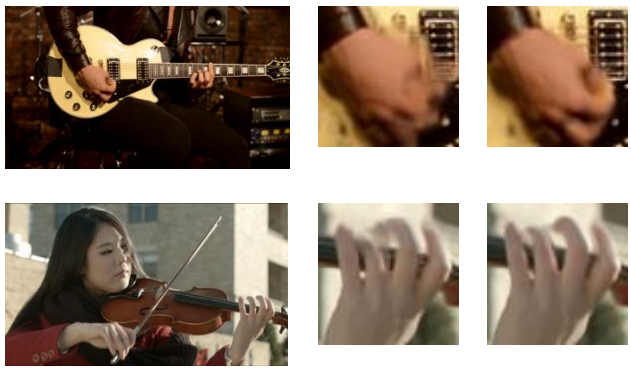
encoded intermediate optical flow by the proposed network, in order from (g) → (h) → (i)→ (k) are the interpolated frames generated by respectively above optical flows (c) → (d) → (e) → (f) with the same frame interpolation algorithm in [53], (l) the ground truth intermediate frame, (m) optical flow-color mapping

Comparison between the first synthesis network and Sepconv.

The effect of the motion-ness on the performance of the first synthesis network is also evaluated. The motion guided warping operations and the first synthesis network are glued together by motion derivation module that adds a motion-ness into the pixel matching loss, this makes the first synthesis network learn not only pixel matching but also motion constrains and scenarios, meanwhile the Sepconv network [55] is similar to a pixel or patch matching that only learns for a pixel loss. Table 6.5 and Figure 6-12 show that the first synthesis network outperforms SepConv in both objective comparisons and subjective visual evaluations.

Table 6.5 Comparison between SepConv and the synthesis network 1

	SepConv		the network 1	
Metric	PSNR	SSIM	PSNR	SSIM
Subtitle	33.83	0.9924	33.93	0.9925
Occlusion	30.92	0.9622	31.53	0.9666
Soccer	29.79	0.9636	30.06	0.9686
Basketball	34.84	0.9876	35.32	0.9895
Average	32.35	0.9765	**32.71**	**0.9793**



(a) (b) (c)

Figure 6-12 Visual comparison between SepConv and the synthesis network 1

(a) Ground truth, (b) SepConv, (c) The network 1

Intermediate results with intermediate optical flows. Finally, a simulation result is investigated to show how the second synthesis learns from intermediate interpolated frames. Figure 6-13 shows the input images, intermediate results and the final output image. Figures 6-13 (k), (l) and (m) show that the second synthesis network encodes short displacements between intermediate networks by bridging the gap between two branches, optical flow based synthesis and learned CNN kernels based interpolated frames and the final one. Therefore, it detects and alleviates the errors from intermediate results in order to generate a better final interpolated frame.

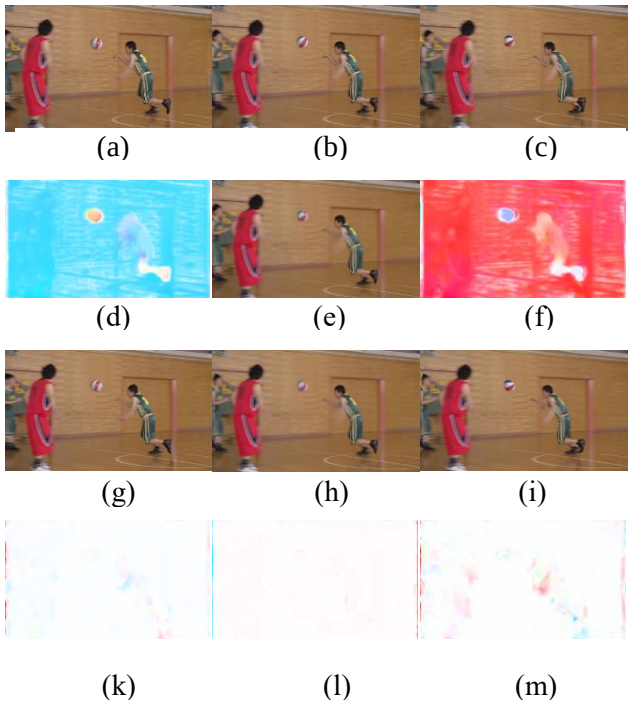


Figure 6-13 Step-by-step analysis of layers

(a) Original frame 1, (b) Ground truth target frame to be interpolated or the original frame 1.5, (c) Original frame 2, (d) The backward intermediate optical flow from frame 1.5 to frame 1, (e) The very first intermediate frame generated by the first synthesis network, (f) The forward intermediate optical flow from frame 1.5 to frame 2, (g) The warped frame from (a) by using (d), (i) The warped frame from (c) by using (f), (h) The final interpolated frame or the output of the second synthesis network, (k) The flow between (g) and (h), (l) The flow between (e) and (h), and (m) The flow between (i) and (h).

Effect of Loss functions: In order to verify the effective of the total loss function as described in section 6.2.4, we dive into the analysis of the effect of intermediate loss components on the total loss function. As described in section 6.2.4, the total

loss function that contains the final loss and two intermediate losses, respectively, as the order given in below equation (6.5).

$$\textit{Loss function} = \| I_{1.5} - I_{gt} \|_1 + \| I^3_{1.5} - I_{gt} \|_1 + \| I^w_{1.5} - I_{gt} \|_1 \qquad (6.7)$$

where I_{gt} is the ground truth frame, the first term is the final loss, the second term is the intermediate loss, corresponds to the first synthesis network, and the third term is the intermediate loss, corresponds to the motion branch, with $I^w_{1.5} = (I^1_{1.5} + I^2_{1.5}) / 2.0$ represents for the warped intermediate frames, $I^1_{1.5}$ and $I^2_{1.5}$ obtained by both forward and backward warping operations.

Now, we dive into three following cases.

Case 1: Total loss function is the same as the equation (6.7).

Case 2: Remove the third term out of the total loss

$$\textit{Loss function} = \| I_{1.5} - I_{gt} \|_1 + \| I^3_{1.5} - I_{gt} \|_1 \qquad (6.8)$$

Case 3: Remove both the second term and the third term out of the total loss, the total loss contains only the final loss.

$$\textit{Loss function} = \| I_{1.5} - I_{gt} \|_1 \qquad (6.9)$$

Table 6.6 compares results obtained by three above cases. From the results shown in Table 6.6, we conclude that training the network with only a single loss at the end of the network is not effective. In addition, intermediate losses can help to avoid over-fitting.

Table 6.6 The effect of loss components

Dataset	PSNR (dB)		
	Case 3	Case 2	Case 1
UCF101	33.03	34.76	34.96
Vimeo90K	33.21	34.55	34.65
HCD	31.68	33.49	33.55

Computational Complexity.

The proposed network is a back-to-back stack of two synthesis networks with total number of parameters are around 43.6 million. On a single Titan X (Pascal) graphics card, it takes 0.86 seconds to generate an interpolated frame with the resolution of 1920 × 1080 and 0.55 seconds to interpolate an intermediate frame with the resolution of 640×480.

6.3. Extension for Quality Enhancement for Compressed Videos Task

Interestingly, the proposed video frame interpolation network is designed for video frame interpolation or frame rate up conversion problem, but it can apply for post-processing task of compressed videos in order to improve quality of the compressed videos. One more time, it again shows the generalization of the proposed network.

In this section, I call the proposed video frame interpolation network described in section 6.1 as a MEMC network that composes of two synthesis networks, synthesis

network 1, synthesis network 2, motion derivation and warp operations modules. The input of the proposed MEMC network are three reconstructed frames, the current reconstructed frame that to be enhanced and two nearest neighbor reconstructed frames, one is the previous frame, the other is the next frame. The first synthesis network and motion derivation module role as a motion estimation network that will derive motion vectors between the current frame and two nearest neighbor frames. In other hand, the warp operations and the second synthesis network role as a motion compensation network that generates the enhanced current frame from three inputs, the current frame, motion compensated previous frame and motion compensated next frame. As described in section 6.1, the proposed motion estimation network is an analysis-by-synthesis technique that estimates motion more accurate than conventional analysis based approaches. In addition, unlike MF-Net [73] for this task, that only work well for low delay configuration, the proposed MEMC network can be applied for any kind of coding configuration, both low delay configuration and random access configuration.

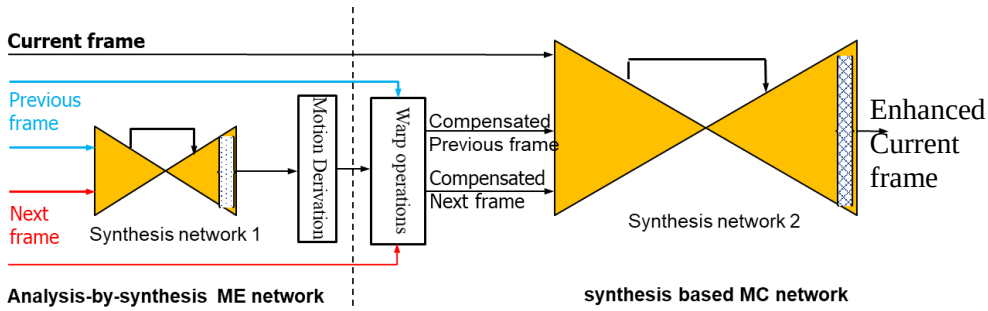


Figure 6-14 Architecture of the MEMC network

Experimental Results.

The proposed MEMC network is compared to the previous state-of-art MF-Net [73] that is designed specifically for this task. With both low delay and random

access configurations, video sequences are compressed by the same HEVC reference software (HM 16.0) with Quantization Parameter (QP) = 37. Table 6.7 shows comparison on the quality of reconstructed sequence, among the baseline HEVC, the enhanced sequence by the previous MF-Net and the enhanced sequence obtained by the proposed MEMC network. In low delay configuration, both MF-Net and the proposed network improve quality of reconstructed frames significantly, 0.41 dB and 0.35 dB respectively meanwhile in random access configuration, only the proposed method enhances the quality of reconstructed frames, even MF-Net degrades the quality of reconstructed frames compare to the baseline HEVC, owing to large displacement between the target frame and its nearest anchor-frames.

Table 6.7 Comparison on the quality of reconstructed frames

PSNR (dB)					
Low delay			Random Access		
HEVC	MF-Net	Ours	HEVC	MF-Net	Ours
31.09	31.50	31.44	31.79	31.72	31.91

In order to have an insight analysis on how each frame is improved by each method, Figure 6-15 and 6-16 show the examples of frame by frame comparison between methods. The blue line denotes the PSNR of reconstructed frames obtained by base line HEVC (or decoded by HEVC), the orange line denotes the PSNR of enhanced frames by applying MF-Net, and the grey line denotes the PSNR of enhanced frames obtained by the proposed MEMC network. The period of each anchor frame is four, that means frames 0, 4, 8, 12, 16 and so on are anchor frames. Lowest quality reconstructed frames are frames just before the corresponding anchor frames, such

as frame 3, 7, 11, 15 and so on enhanced significantly by both post-processing network, MF-Net and the proposed MEMC network, 0.68 and 0.57 dB increase from baseline HEVC, respectively as shown in Table 6.10. As shown in table 6.8, the proposed method outperforms the MF-Net at frames next right after anchor frames such as frame 1, 5, 9, 13, 17 and so on. Because in MF-Net approach, distance from those frames to corresponding anchor frames are asymmetric. For middle frames between anchor frames, such as frame 2, 6, 10, 14, and so on, the MF-Net improves better than the proposed MEMC network, 0.36 dB improvement, compare to 0.22 dB improvement obtained by the proposed MEMC network as shown in Table 6.9.

Table 6.8 Comparison at frames right after anchor frames.

Frame	1	5	9	13	17	21	25	29	33	37	41	45	49	53	Aver.
MF-Net	0	0.14	0.2	0.12	0.16	0.22	0.17	0.24	0.21	0.13	0.22	0.22	0.23	0	0.16
Proposed	0.23	0.25	0.18	0.14	0.3	0.17	0.15	0.14	0.23	0.19	0.26	0.29	0.29	0.31	0.22

Table 6.9 Comparison at middle frames between anchor frames

Frame	2	6	10	14	18	22	26	30	34	38	42	46	50	54	Aver.
MF-Net	0.26	0.36	0.37	0.34	0.42	0.47	0.17	0.41	0.21	0.50	0.36	0.49	0.52	0.19	0.36
Proposed	0.11	0.25	0.09	0.08	0.35	0.33	0.14	0.41	0.23	0.41	0.28	0.17	0.09	0.13	0.22

Table 6.10 Comparison at lowest quality reconstructed frames (frame just before anchor frames)

Frame	3	7	11	15	19	23	27	31	35	39	43	47	51	55	Aver.
MF-Net	0.58	0.57	0.65	0.64	0.76	0.65	0.66	0.66	0.67	0.68	0.69	0.84	0.83	0.63	0.68
Proposed	0.48	0.6	0.4	0.53	0.55	0.52	0.54	0.6	0.57	0.62	0.55	0.67	0.76	0.6	0.57

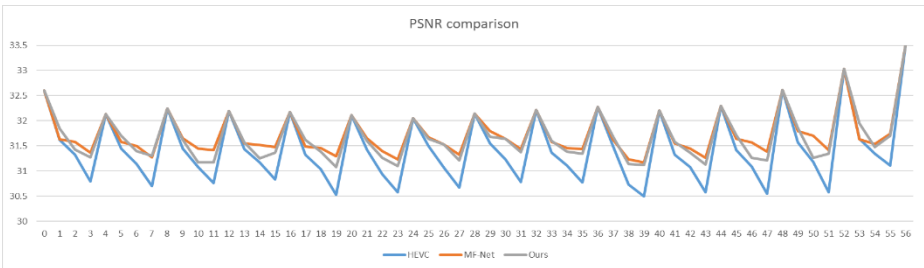


Figure 6-15 An example of frame by frame comparison in low delay configuration

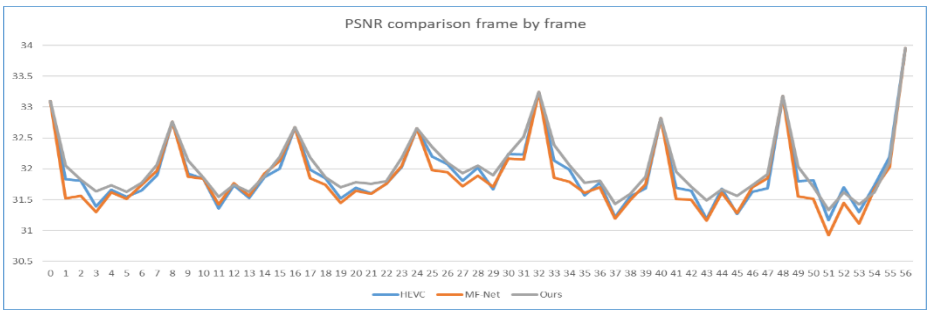
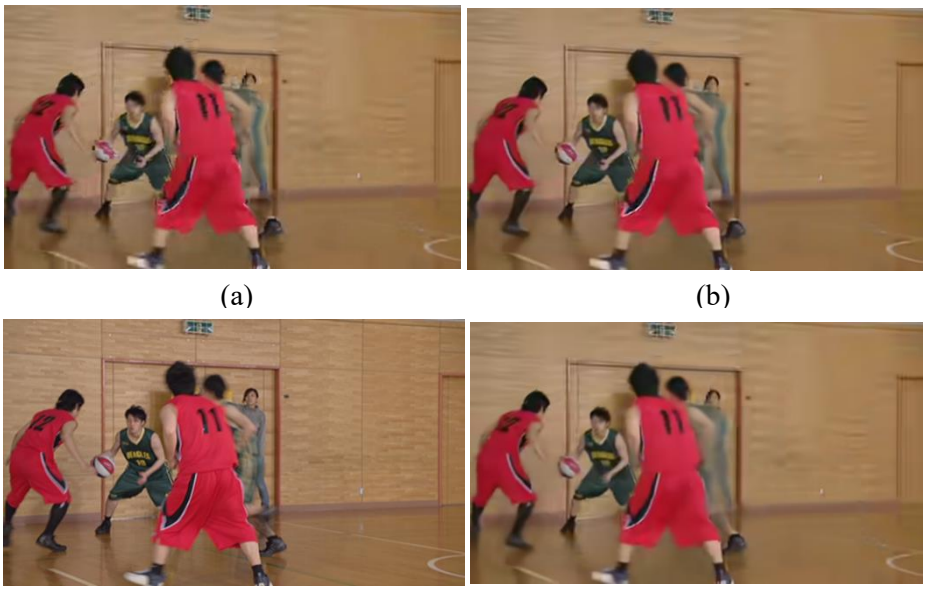


Figure 6-16 An example of frame by frame comparison in random access configuration

Figure 6-17 shows a visual comparison between enhanced reconstructed frame obtained by MF-Net and that obtained the proposed MEMC network. Both enhanced frames alleviate blocking artifacts significantly in comparison to the raw

reconstructed frame decoded by HEVC baseline, especially at regions around the ball.



(a) (b)

(c) (d)

Figure 6-17 Visual comparison between enhanced reconstructed frame obtained by MF-Net and that obtained the proposed MEMC network

(a) Reconstructed frame by HEVC, (b) Enhanced frame by MF-Net, (c) Ground truth, (d) Enhanced frame by Ours

6.4. Extension for Improving the Coding Efficiency of HEVC based Low Bitrate Encoder

As the previous video coding standards, High Efficiency Video Coding (HEVC) or H.265 [3] adopts block based hybrid video coding framework where inter prediction, which aims to remove the temporal redundancy, serves as a critical part of the coding framework. In particular, in B frame with random access configuration,

inter prediction makes use of the temporal correlation between the current to-be-coded picture and neighbor pictures in both directions, previous and future in order to obtain a predicted version of the current picture, and then encode the residual between the predicted values and the original values.

The Hierarchical B Coding Structure is adopted in HEVC, thanks to its coding efficiency. A typical hierarchical B structure with 4 temporal levels in Random Access (RA) configuration is depicted in Figure.1, where frame 0 and frame 1 (denoted as the coding order) belong to temporal level 0, which provide high quality reference for subsequent frames. Once frames in level 0 are reconstructed, level 1 frame 2 can be bi-predicted by frame 0 and frame 8. Regarding level 2 frames 3 and frame 6, both reconstructed frames of level 0 and level 1 can be used as references. Lastly, level 3 contains frame 4, 5, 7 and 8 which reference nearest lower level frames in both forward and backward directions, previous and future. Generally speaking, each B picture can be predicted using the nearest pictures of the lower temporal levels in forward and backward directions. Furthermore, as the temporal distance between two reference frames is getting closer for higher level frames, the prediction of the intermediate frame becomes more reliable.

In random access configuration, a temporal level is assigned to frames in order to apply differently quantization parameters (QP) in order to satisfy the trade-off between bitrate and the quality of the reconstructed frames.

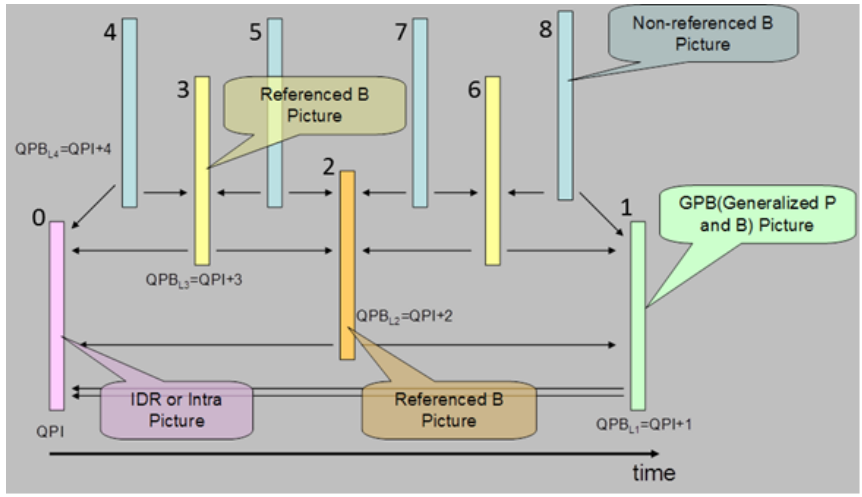


Figure 6-18 Graphical representation of random access configuration [33]

Figure 6-18 shows a graphical representation of random access configuration, where highest temporal levels frames, denoted as blued frames (or non-referenced B picture) in figure 1, they are encoded based on the prediction from reference frames with the highest QP offset (QP_{offset} = 4). Consequently, it takes less bits to encode those non-referenced B pictures. Even those frames are non-referenced frames, they are still stored in reference buffers. Consequently, it consumes memory footprint.

Recently, with the break-through of Convolutional Neural Networks (CNN) in video super resolution, video frame interpolation (or deep frame rate up conversion), the quality of the interpolated frames generated by deep FRUC methods are close to the real original frames. It suggests a combination between deep FRUC and HEVC in order to improve coding efficiency without scarifying the quality of reconstructed frame. In addition, in random access configuration, skip encoding highest temporal level frames have several following benefits.

1. Approximately, reduce encoding time twice

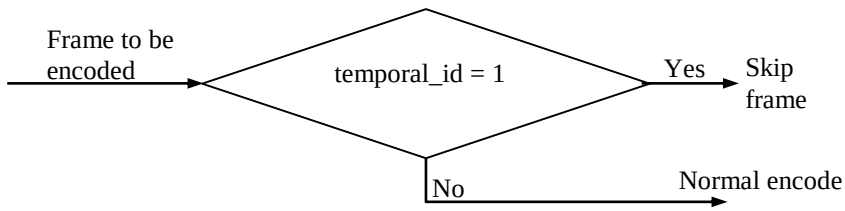
2. Reduce memory buffer for saving non-referenced reconstructed pictures.

3. Improve coding efficiency

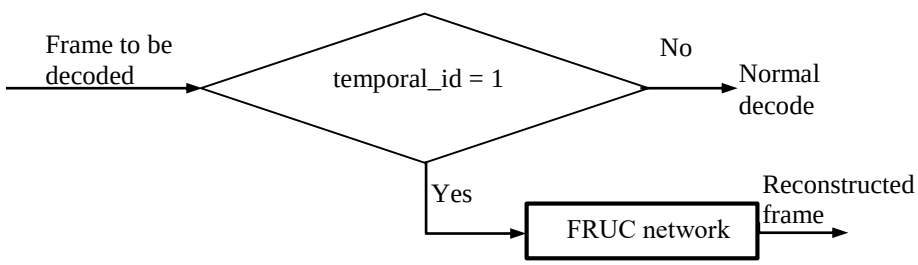
4. Do not change the structures of both encoder and decoder

5. Reconstructed frames are enhanced (i.e. without blocking artifacts) with higher visual quality

In this section, a combination between the proposed FRUC network in section 6.1. and HEVC is described as below figure.



At encoder side



At decoder side

Figure 6-19 Diagram of the integration of FRUC net into encoder/decoder sides of HEVC

Experimental Results.

As shown in Figure 6-20 and figure 6-21, RD curve comparison between HEVC baseline and the proposed method, in low bit rate regions, the proposed method,

denoted as B curve, that integrates FRUC into HEVC improves the coding efficiency of the standard HEVC baseline (HM 16.0), denoted as A curve.

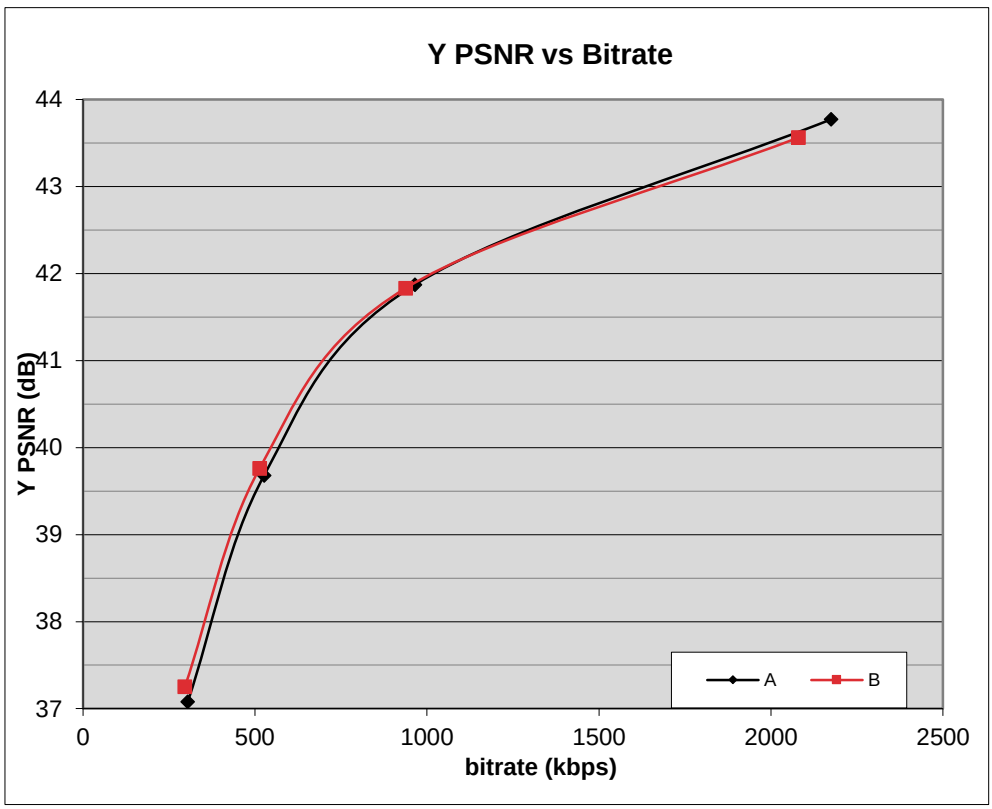


Figure 6-20 RD curve comparison between HEVC baseline and the proposed method (FRUC + HEVC) for vidyo1 sequence

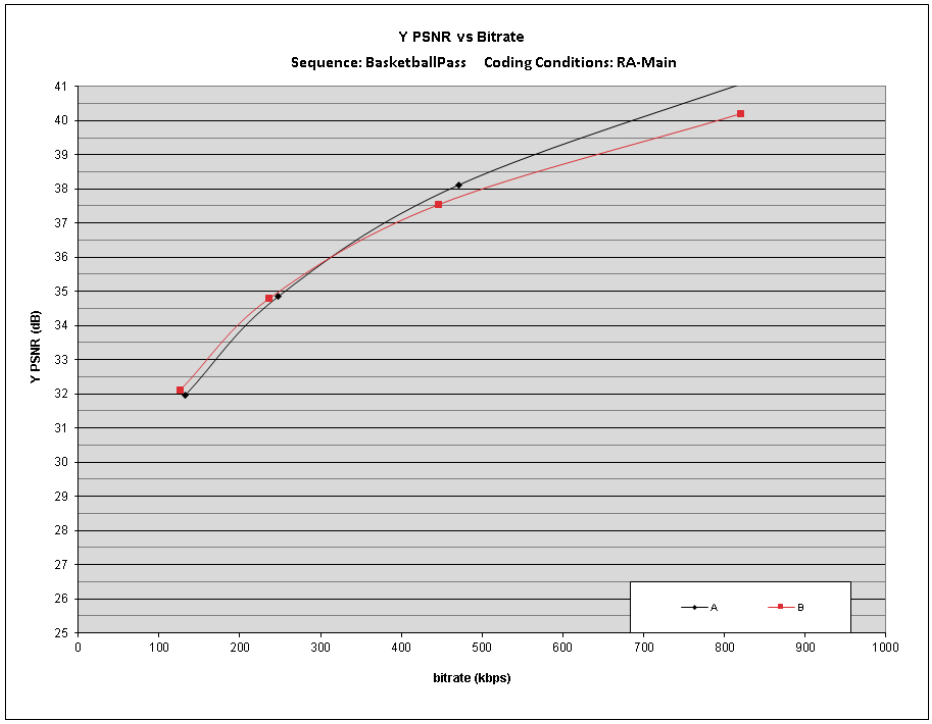


Figure 6-21 RD curve comparison between HEVC baseline and the proposed method (FRUC + HEVC) for basketball pass sequence

Chapter 7. Conclusion

This thesis tackles on various challenging cases for motion estimation of small objects, motion estimation of a repetition pattern region, motion compensated frame interpolation and video frame interpolation with a deep CNN network.

In conventional block-based hierarchical motion estimation, a motion vector of a small object is not detected at the top level, and thereby resulting in object deformation in interpolated frames in MC-FRUC. This thesis proposes a new algorithm for estimating the motion of a small object in a hierarchical motion estimation framework, which improves the image quality of an interpolated frame. The proposed algorithm detects high-cost pixels for each block, and estimates the motion vector of high-cost pixels. This motion vector is used as an additional motion vector candidate in hierarchical motion estimation. The additional motion vector is propagated to the bottom level, and thus enabling a motion vector of a small object to be discovered at the bottom level. Experimental results for MC-FRUC show that the proposed algorithm achieves a better performance than the MAP algorithm in terms of both subjective image quality and objective measurements. The PSNR is improved by 0.42 dB on average by using the proposed algorithm.

This thesis presents a novel algorithm to estimate the motion information in repetition pattern regions. The algorithm is the first to adopt a semi-global approach that exploits both local and global properties of repetition pattern regions. It merges the repetition pattern blocks into a large region and makes the histogram of the smallest local minima of all blocks in the region. The merging represents a large region and makes the histogram of motion vector candidates that correspond to the

smallest local minima of a SAD surface. The histogram of the motion vector candidates is built by using a voter based voting system that more reliable than an elector based voting system. It improves the accuracy of the motion vectors in the repetition pattern region. The proposed algorithm is simple but effective in the estimation of the motion vectors for repetition pattern blocks. In other word, it obtains the objective function that is to estimate correctly the motion vector of a large repetition pattern region with low complexity

This thesis proposes a non-selective adaptive weighted motion compensation for frame rate up conversion algorithm. It projects both forward and backward motion vectors into interpolated frame in order to generate bi-directional motion vectors. The algorithm preserves completely all motion vectors of overlapped projected blocks. And an adaptive weighted motion compensation is done for interpolated blocks correspond to their own preserved motion vectors. The weighted coefficients are computed by using a comprehensive metric that composes of distance or overlap area, matching cost and smoothness cost correspond to the preserved motion vectors. Holes are filled by vector median filter of the motion vector of non-hole neighbor blocks. The proposed algorithm outperforms previous methods and reduce block artifact significantly.

This thesis proposes a back-to-back stack of synthesis networks by bridging the gap between two branches, optical flow based synthesis and learned CNN kernels based interpolation together into a comprehensive joint framework. Intermediate optical flows are introduced and estimated directly from learned CNN kernels by using analysis-by-synthesis technique and vice versa the synthesis network learns not only a pixel matching loss but also motion-ness criterion. Consequently, the

proposed method handles fast, complex motions of small objects effectively. The proposed network is also the first attempt to bridge two branches of previous approaches, optical flow based synthesis and CNN kernels based synthesis into a comprehensive network. The proposed method is evaluated with various datasets and outperforms previous methods in both objective metrics and subjective visual evaluations.

References

[1] Y. Liu, Y. Wang, and S. Chien, “Motion blur reduction of liquid crystal displays using perception-aware motion compensated frame rate up-conversion.” *IEEE Works. Signal Process. Syst.* pp. 84-89, 2011

[2] T. Tsai, and H. Lin, “Hybrid frame rate up conversion method based on motion vector mapping.” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no.11, pp.1901-1910, June 2013

[3] G. J. Sullivan, J. R. Ohm, W.J. Han and Thomas Wiegand, " Overview of the High Efficiency Video Coding (HEVC) Standard." *IEEE Trans Circuits Syst. Video Technol.* vol. 22, no. 12, pp. 1649-1668, Dec. 2012

[4] B.D. Choi, J.W. Han, C.S. Kim, and S.J. Ko, "Motion-compensated frame interpolation using bilateral motion estimation and adaptive overlapped block motion compensation." *IEEE Trans. Circuits Syst. Video Technol.* vol. 17, no. 4, pp. 407-416, April 2007.

[5] T. Tran, and C.T.L. Dinh, "Frame rate converter with pixel-based motion vectors selection and halo reduction using preliminary interpolation." *IEEE J. Select. Topics Signal Process.* vol. 5, no. 2, pp. 252-261, April 2011

[6] S.G. Jeong, C. Lee, and C.S. Kim, "Motion-compensated frame interpolation based on multi hypothesis motion estimation and texture optimization." *IEEE Trans. Image Process.* vol. 22, no.11, pp. 4497-4509, Nov. 2013

[7] G. Lee, C. Chen, C. Hsiao, and J. Wu, "Bi-directional trajectory tracking with variable block-size motion estimation for frame rate Up-convertor." *IEEE J. Emerging Select. Topics Circuits Syst.* vol. 4, no. 1, pp. 29-42, March 2014

[8] J.G. Kim, and D.H. Lee. "Frame rate up conversion using pyramid structure and dense motion vector fields." *J. Electro. Imaging* vol. 25, no. 3, May 2016.

[9] D. Wang, L. Zhang, and A. Vincent "Motion-compensated frame rate up-conversion—Part I: Fast multi-frame motion estimation.″ *IEEE Trans. Broadcasting.* vol. 56, no. 2, pp.133-141, June 2010.

[10] A. Heinrich, R. J. Vleuten, and G. De Haan, “Perception-Oriented Methodology for Robust Motion Estimation Design." *IEEE J. Select. Topics Signal Process.* vol. 8, no. 3, pp. 463-474, June 2014.

[11] D. Salih, and Y. Altunbasak. "Novel true-motion estimation algorithm and its application to motion-compensated temporal frame interpolation." *IEEE Trans. Image Process.* vol. 22, no. 8, pp. 2931-2945, Aug. 2013

[12] V. T. Nguyen, H.J. Lee, “An Efficient Non-Selective Adaptive Motion Compensated Frame Rate Up Conversion,” *IEEE Inter. Sym. Circuits Syst.* May. 2017

[13] D. Choi, W.S. Song, H. Choi, and T.J. Kim "MAP-Based Motion Refinement Algorithm for Block-Based Motion-Compensated Frame Interpolation." *IEEE Trans. Circuits Syst. Video Technol.* vol. 26, no. 10, pp. 1789-1804, Oct. 2016

[14] N. Jacobson, Y.L. Lee, V. Mahadevan, N. Vasconcelos, and T. Q. Nguyen, “A novel approach to FRUC using discriminant saliency and frame segmentation,” *IEEE Trans. Image Process*., vol. 19, no. 11, pp. 2924–2934, Nov. 2010.

[15] A.M. Huang and T. Q. Nguyen, “A multistage motion vector processing method for motion-compensated frame interpolation,” *IEEE Trans. Image Process*., vol. 17, no. 5, pp. 694–708, May 2008

[16] H. Liu, R. Xiong, D. Zhao, S. Ma, and W. Gao, "Multiple Hypotheses Bayesian Frame Rate Up-Conversion by Adaptive Fusion of Motion-Compensated Interpolations," *IEEE Trans. Circuits Syst. Video Technol.* vol. 22, no. 8, pp. 1188–1198, Aug. 2012

[17] V. T. Nguyen, H.J. Lee, "A Semi-global Motion Estimation of a Repetition Pattern Region for Frame," *IEEE Inter. Conf. Image Process.* Sep. 2017

[18] S.H. Lee, O.J. Kwon, and R.H. Park, "Motion Vector Correction based on the Pattern-like Image Analysis," *IEEE Trans. Consumer Electron.* vol. 49, no. 3, pp.479-484, Aug. 2003

[19] S.G. Kim, T.G. Ahn, and S.H. Park, "Motion Estimation Algorithm for Periodic Pattern Objects based on Spectral Image Analysis," *Proc. IEEE Inter. Conf. Consumer Electron.* 2013

[20] H. K. Horn and B. G. Schunck, "Determining optical flow," Artificial Intelligence, 1981, pp. 185-203

[21] C. Zach, T. Pock, and H. Bischof, "A duality based approach for real time TV-L 1 optical flow". *Pattern Recognition*, 2007, pp. 214-223.

[22] Bartels, C. & De Haan, G, "Smoothness constraints in recursive search motion estimation for picture rate conversion". *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 10, pp. 1310–1319, Oct. 2010

[23] D. Wang, A. Vincent, P. Blanchfield, and R. Klepko, "Motion-compensated frame rate up-conversion—Part II: New algorithms for frame interpolation". *IEEE Trans. Broadcasting*, vol. 56, no. 2, pp. 142-149, June 2010

[24] S.H. Lee, O.J. Kwon, and R.H. Park, “Motion Vector Correction based on the Pattern-like Image Analysis,” *IEEE Trans. Consumer Electron.*, vol. 49, no. 3, pp. 479-484, Aug. 2003.

[25] Y.W. Sohn and M.J. Kang, “Block based Motion Vector Smoothing for Periodic Pattern Region,” in *Proc. ICIAR,* 2007.

[26] S.G. Kim, T.G. Ahn, and S.H. Park, “Motion Estimation Algorithm for Periodic Pattern Objects based on Spectral Image Analysis,” in *Proc. IEEE ICCE,* 2013.

[27] A. Heinrich, C. Bartels, R.J. Vleuten and G. de Haan, “Robust Motion Estimation Design Methodology,” in *Proc. CVMP,* 2010.

[28] D. Wang, L. Zhang and A. Vincent, “Motion-Compensated Frame Rate Up-Conversion—Part I: Fast Multi-Frame Motion Estimation,” *IEEE Trans. Broadcasting*, vol. 56, no. 2, pp. 133-141, Jun. 2010

[29] Nguyen Van Thang and Huyk-Jae Lee, “An Efficient Non-Selective Adaptive Motion Compensated Frame Rate Up Conversion” in *Proc. IEEE ISCAS*, 2017.

[30] Sang-Heon Lee and Hyuk-Jae Lee, “Motion- Compensated Frame Interpolation for Intra-mode Block,” *IEICE Trans. Inform. Syst.,* vol. E91-D, no. 4, pp. 1117-1127, Oct. 2008.

[31] D. Choi, W. Song, H. Choi, and T..J. Kim, “MAP-Based Motion Refinement Algorithm for Block-Based Motion-Compensated Frame Interpolation,” *IEEE Trans. Circuits Syst. Video Technol.,* vol. 26, no. 10, pp. 1789 – 1804, Oct. 2016.

[32]Orchard, M. T., & Sullivan, G. J. “Overlapped block motion compensation: An estimation-theoretic approach”. *Image Processing, IEEE Transactions* on, vol 3, no 5, September, 1994, pp. 693-699.

[33]ITU-T H.265, “SERIES H: AUDIOVISUAL AND MULTIMEDIA SYSTEMS

Infrastructure of audiovisual services – Coding of moving video High efficiency video coding”. *International Telecommunication Union,* April, 2013.

[34]Kim U. & M. Sunwoo, “New frame rate up-conversion algorithms with low computational complexity,” *IEEE Transaction on Circuits System on Video Technology*, vol. 24, no. 3, March. 2014, pp. 384–393.

[35]Kang, S. J., Yoo, S., & Kim, Y. H, “Dual motion estimation for frame rate up-conversion”. *IEEE Transactions on Circuits and Systems for Video Technology*, 20(12), 2010.

[36]Choi, B. D., Han, J. W., Kim, C. S., & Ko, S. J, “Motion-compensated frame interpolation using bilateral motion estimation and adaptive overlapped block motion compensation”. *IEEE Transactions on Circuits and Systems for Video Technology*, 17(4), 2007.

[37]Kang, S. J., Cho, K. R., & Kim, Y. H, “Motion compensated frame rate up-conversion using extended bilateral motion estimation”. *IEEE Transactions on Consumer Electronics*, 53(4), 2007.

[38]Yoo, D.G., Kang, S.J. & Kim, Y.H., “Direction-select motion estimation for motion-compensated frame rate up-conversion”. *Journal of Display Technology,* 9(10), 2013. pp.840-850.

[39]Y. Guo, L. Chen, and X. Chang. “Frame Rate Up-Conversion Method for Video Processing Applications”, *IEEE Trans on Broadcasting* vol. 60, no. 4, Dec, 2014.

[40] Tsai, T.H. & Lin, H.Y., “Hybrid Frame Rate Up Conversion Method Based on Motion Vector Mapping.”, *IEEE Transactions on Circuits and Systems for Video Technology*, 23(11), 2013, pp.1901-1910.

[41]Hei Law, Jia Deng. CornerNet: Detecting Objects as Paired Keypoints. *Proc. European Conf. Computer Vision*, Sep. 2018.

[42]K. Ma , Z. Shu, X. Bai, J. Wang, and D. Samaras, DocUNet: Document Image Unwarping via A Stacked U-Net. *Proc. IEEE Conf. Computer Vision Pattern Recognition*, June 2018.

[43]D. J. Butler, J. Wulff, G. B. Stanley, M. J. Black. A naturalistic open source movie for optical flow evaluation. *Proc. European Conf. Computer Vision*, Oct. 2012.

[44]Z. Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity, *IEEE Trans. Image Processing*, vol. 13, no. 4, pp. 600-612, April 2004.

[45]A. Newell, K. Yang, and J. Deng. Stacked Hourglass Networks for Human Pose Estimation. *Proc. European Conf. Computer Vision*, Oct. 2016.

[46]A. Dosovitskiy, P. Fischer, E. Ilg, P. H¨ausser, C. Hazirbas, V. Golkov, P. V. D. Smagt, D. Cremers, and T. Brox. FlowNet: Learning optical flow with convolutional networks. *IEEE Int. Conf. Computer Vision*, pages 2758–2766, 2015

[47]L. L. Raket, L. Roholm, A. Bruhn, and J. Weickert. Motion compensated frame interpolation with a symmetric optical flow constraint. *Advances in Visual Computing*, vol. 7431, pages 447–457, 2012

[48]Z. Yu, H. Li, Z. Wang, Z. Hu, and C. W. Chen. Multi-level video frame interpolation: Exploiting the interaction among different levels. *IEEE Trans. Circuits Syst. Video Techn*., vol. 23, no. 7, pages 1235–1248, 2013

[49]A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Proc. Advances Neural Inform. Process. Syst*., 2012.

[50]R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *IEEE Conf. Computer Vision Pattern Recognition*, pages 580-587, June 2014

[51]J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. *IEEE Conf. Computer Vision Pattern Recognition*, pages 3431– 3440, 2015

[52]G. Long, L. Kneip, J. M. Alvarez, H. Li, X. Zhang, and
Q. Yu. Learning image matching by simply watching video. *Proc. European Conf. Computer Vision*, vol. 9910, pp.434–450, Oct. 2016.

[53] S. Baker, D. Scharstein, J. P. Lewis, S. Roth, M. J. Black, and R. Szeliski. A database and evaluation methodology for optical flow. *Int. J. Computer Vision*, vol 92 no. 1, pp.1–31, 2011.

[54] S. Niklaus, L. Mai, and F. Liu. Video frame interpolation via adaptive convolution. Proc. *IEEE Conf. Computer Vision Pattern Recognition*, July 2017

[55] S. Niklaus, L. Mai, and F. Liu. Video frame interpolation via adaptive separable convolution. *Proc. IEEE Int. Conf. Computer Vision*, Oct. 2017

[56] S. Niklaus, and F. Liu. Context-aware Synthesis for Video Frame Interpolation. *Proc. IEEE Conf. Computer Vision Pattern Recognition*, June 2018

[57] Z. Liu, R. Yeh, X. Tang, Y. Liu, and A. Agarwala. Video frame synthesis using deep voxel flow. *Proc. IEEE Int. Conf. Computer Vision*, Oct. 2017

[58] Jiang, H., Sun, D., Jampani, V., Yang, M. H., Learned-Miller, E., and Kautz, J. Super SloMo: High Quality Estimation of Multiple Intermediate Frames for Video Interpolation. *Proc. IEEE Conf. Computer Vision Pattern Recognition*, June 2018

[59] L. Xu, J. Jia, and Y. Matsushita. Motion detail preserving optical flow estimation. *IEEE Trans. Pattern Analys. Machine Intel.,* vol. 34 no. 9, pp.1744–1757, 2012

[60] S. Meyer, O. Wang, H. Zimmer, M. Grosse, and A. Sorkine Hornung. Phase-based frame interpolation for video. *Proc. IEEE Conf. Computer Vision Pattern Recognition*, June 2015

[61] S. Meyer, A. Djelouah, B. McWilliams, A. S. Hornung, M. Gross, and C. Schroers. PhaseNet for Video Frame Interpolation. *Proc. IEEE Conf. Computer Vision Pattern Recognition*, June 2018

[62] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid. DeepFlow: large displacement optical flow with deep matching. *Proc. IEEE Int. Conf. Computer Vision*, Dec. 2013.

[63] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, William T. Freeman, Video Enhancement with Task-Oriented Flow.

https://arxiv.org/abs/1711.09078

[64] T. Xue, J. Wu, K. L. Bouman, and B. Freeman. Visual dynamics: Probabilistic future frame synthesis via cross convolutional networks. *Proc. Advances Neural Inform. Process. Syst.*, pages 91–99, 2016

[65] A. Bansal, X. Chen, B. Russell, A. Gupta, and D. Ramanan. PixelNet: Representation of the pixels, by the pixels, and for the pixels.

https://arxiv.org/abs/1702.06506

[66] D. P. Kingma and J. Ba. Adam: A method for stochastic Optimization.

https://arxiv.org/abs/1412.6980

[67] D. Sun, X. Yang, M. Liu, and J. Kautz. PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. *Proc. IEEE Conf. Computer Vision Pattern Recognition*, June 2018

[68] K. Soomro, A. Roshan Zamir, and M. Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. In CRCV-TR-12-01, 2012

[69] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. *Proc. IEEE Conference Computer Vision Pattern Recognition* , Jul. 2017

[70] A. Ranjan and M. J. Black. Optical flow estimation using a spatial pyramid network". *Proc. IEEE Conf. Computer Vision Pattern Recognition*, Jul. 2017

[71] A. Aghajanyan. Convolution aware initialization.

https://arxiv.org/abs/1708.01692

[72] Y. Lu, J. Valmadre, H. Wang, J. Kannala, M. Harandi, and P. H. S. Torr, Devon: Deformable Volume Network for Learning Optical Flow

https://arxiv.org/abs/1802.07351

[73] Ren Yang et. al. “Multi-Frame Quality Enhancement for Compressed Video”, *Proc. IEEE Conf. Computer Vision Pattern Recognition*, June 2018

초록

블록 기반 계층적 움직임 추정은 고화질의 보간 이미지를 생성할 수 있어 폭넓게 사용되고 있다. 하지만, 배경 영역이 움직일 때, 작은 물체에 대한 움직임 추정 성능은 여전히 좋지 않다. 이는 maximum a posterior (MAP) 방식으로 이미지 피라미드의 최상위 레벨에서 down-sampling 과 over-smoothing 으로 인해 작은 물체의 움직임이 무시되기 때문이다. 결과적으로 이미지 피라미드의 최하위 레벨에서 작은 물체의 움직임 벡터는 검출될 수 없어 보간 이미지에서 작은 물체는 종종 변형된 것처럼 보인다. 본 논문에서는 작은 물체의 움직임을 나타내는 2 차 움직임 벡터 후보를 추가하여 작은 물체의 움직임 벡터를 보존하는 새로운 알고리즘을 제안한다. 추가된 움직임 벡터 후보는 항상 이미지 피라미드의 최상위에서 최하위 레벨로 전파된다. 실험 결과는 제안된 알고리즘의 보간 생성 프레임이 기존 MAP 기반 보간 방식으로 생성된 프레임보다 이미지 화질이 상당히 향상됨을 보여준다.

움직임 보상 프레임 보간에서, 이미지 내의 반복 패턴은 움직임 추정을 위한 정합 오차 탐색 시 다수의 유사 local minima 가 존재하기 때문에 정확한 움직임 벡터 유도를 어렵게 한다. 본 논문은 반복 패턴에서의 움직임 추정의 정확도를 향상시키기 위해 반복 영역의 local 한 특성과 global 한 특성을 동시에 활용하는 semi-global 한 접근을 시도한다. 움직임 벡터 후보의 히스토그램은 선거 기반 투표 시스템보다 신뢰할 수 있는 유권자 기반 투표 시스템 기반으로 형성된다.

실험 결과는 제안된 방법이 이전의 local 한 접근법보다 peak signal-to-noise ratio (PSNR)와 주관적 화질 판단 관점에서 상당히 우수함을 보여준다.

비디오 프레임 보간 또는 움직임 보상 프레임율 상향 변환 (MC-FRUC)에서, 단방향 움직임 궤적에 따른 움직임 보상은 overlap 과 hole 문제를 일으킨다. 본 연구에서 이러한 문제를 해결하기 위해 양방향 움직임 보상 프레임 보간을 위한 새로운 알고리즘을 제시한다. 먼저, 제안된 방법은 단방향 움직임 추정으로부터 얻어진 두 개의 단방향 움직임 영역(전방 및 후방)으로부터 양방향 움직임 벡터를 생성한다. 이는 전방 및 후방 움직임 벡터를 보간 프레임에 투영함으로써 수행된다. 보간된 블록에 여러 개의 투영된 블록이 있는 경우, 투영된 블록과 보간된 블록 사이의 거리를 확장하는 기준이 가중 계수를 계산하기 위해 제안된다. Hole 은 hole 이 아닌 이웃 블록의 vector median filter 를 기반으로 처리된다. 제안 방법은 기존의 MC-FRUC 보다 성능이 우수하며, 블록 열화를 상당히 제거한다.

본 논문에서는 CNN 을 이용한 비디오 프레임 보간에 대해서도 다룬다. Optical flow 및 비디오 프레임 보간은 한 가지 문제가 다른 문제에 영향을 미치는 chicken-egg 문제로 간주된다. 본 논문에서는 중간 optical flow 를 계산하는 네트워크와 보간 프레임을 합성 하는 두 가지 네트워크로 이루어진 하나의 네트워크 스택을 구조를 제안한다. The final 보간 프레임을 생성하는 네트워크의 경우 첫 번째 네트워크의 출력인 보간 프레임 와 중간 optical flow based warped frames 을 입력으로 받아서 프레임을 생성한다. 제안된 구조의 가장 큰 특징은 optical flow 계산을 위한 합성에 의한 분석법과 CNN 기반의

분석에 의한 합성법을 모두 이용하여 하나의 종합적인 framework 로 결합하였다는 것이다. 제안된 네트워크는 기존의 두 가지 연구인 optical flow 기반 프레임 합성과 CNN 기반 합성 프레임 합성법을 처음 결합시킨 방식이다. 실험은 다양하고 복잡한 데이터 셋으로 이루어졌으며, 보간 프레임 quality 와 optical flow 계산 정확도 측면에서 기존의 state-of-art 방식에 비해 월등히 높은 성능을 보였다. 본 논문의 후 처리를 위한 심층 비디오 프레임 보간 네트워크는 코딩 효율 향상을 위해 최신 비디오 압축 표준인 HEVC/H.265 에 적용할 수 있으며, 실험 결과는 제안 네트워크의 효율성을 입증한다.

주요어: frame interpolation, MEMC, CNN, small objects, repetition regions, FRUC

Student Number: 2012-31285