Journal of Zhejiang University SCIENCE A
ISSN 1673-565X (Print); ISSN 1862-1775 (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

Frame rate up-conversion using multiresolution critical point filters with occlusion refinement*

Yi-xiong ZHANG[†], Wei-dong WANG[†‡], Peng LIU, Qing-dong YAO

(*Department of Information Science and Electronic Engineering, Zhejiang University, Hangzhou 310027, China*)

[†]E-mail: zjuzyx@isee.zju.edu.cn; wdwang2001@yahoo.com.cn

Received Mar. 19, 2008; revision accepted July 7, 2008; CrossCheck deposited Nov. 10, 2008

Abstract: In this paper, multiresolution critical-point filters (CPFs) are employed to image matching for frame rate up-conversion (FRUC). By CPF matching, the dense motion field can be obtained for representing object motions accurately. However, the elastic motion model does not hold in the areas of occlusion, thus resulting in blur artifacts in the interpolated frame. To tackle this problem, we propose a new FRUC scheme using an occlusion refined CPF matching interpolation (ORCMI). In the proposed approach, the occlusion refinement is based on a bidirectional CPF mapping. And the intermediate frames are generated by the bidirectional interpolation for non-occlusion pixels combined with unidirectional projection for the occlusion pixels. Experimental results show that ORCMI improves the visual quality of the interpolated frames, especially at the occlusion regions. Compared to the block matching based FRUC algorithm, ORCMI can achieve 1~2 dB PSNR gain for standard video sequences.

Key words: Frame rate up-conversion, Multiresolution critical-point filters, Occlusion refinement
doi:10.1631/jzus.A0820200 **Document code:** A **CLC number:** TN919.8

INTRODUCTION

Frame rate up-conversion (FRUC) is necessary for exchange between two display formats with different frame rates. For example, in order to view filmed programs on television, film should be up-converted to video. This is because films are usually produced at a rate of 24 frames per second (fps), while TV sets require much higher frame rates, such as 50 or 60 fps. In addition, FRUC techniques can also be used for video compression and slow motion replay (Yang *et al.*, 2007).

Until now, simple approaches such as frame repetition and frame averaging have been used for FRUC, but these algorithms cause motion jerkiness or motion blurring in moving objects. Motion compensation FRUC (MC-FRUC), taking motion information into account, has been shown to provide better results and reduce blurring and motion jerkiness (Haavisto *et al.*, 1989; Castagno *et al.*, 1996).

In MC-FRUC, two main issues need to be addressed. First, the purpose of motion estimation in FRUC is to find true motion trajectories, but not to minimize the energy of MC residual signals. Most MC-FRUC techniques (Lee *et al.*, 2002; Ha *et al.*, 2004; Zhai *et al.*, 2005; Choi *et al.*, 2006; 2007) utilize the block matching algorithm (BMA) for motion estimation. BMA is simple and easy to implement. But, the motion vectors (MVs) of blocks estimated by BMA are not faithful to true object motions (Ha *et al.*, 2004). Second, dealing with occlusion is another challenge in MC-FRUC (Wittebrood *et al.*, 2003). In covered (or uncovered) area, the correct pixel information is only available in the past (or future) frame. The MVs of the occlusion pixels will become unreliable and temporal interpolation cannot be applied within the occlusion areas. Occlusion detection techniques for object tracking and frame interpolation have been developed by many

[‡] Corresponding author

[*] Project (No. 2004C21052) supported by the Key Program of the Science and Technology Commission Foundation of Zhejiang Province, China

researchers (Altnubasak and Tekalp, 1997; Sanchez, 2003; Wittebrood *et al.*, 2003; Beller *et al*., 2007). But most of them mainly pay attention to detecting occlusion areas in input frames rather than dealing with occlusion pixels in intermediate frames.

In this paper, elastic image matching using multiresolution critical-point filters (CPFs) (Shinagawa and Kunii, 1998; Habuka and Shinagawa, 2004) is applied to FRUC. The dense motion field obtained by CPF matching can represent non-translational motions such as rotation and zooming effectively (Chambers *et al.*, 2003; Durand and Hutchinson, 2003). However, in the areas of occlusion, the elastic model in the CPF matching algorithm does not hold, and can introduce blur artifacts in the interpolated frames. To overcome this drawback, we propose an occlusion refined CPF matching interpolation (ORCMI) FRUC algorithm. In this approach, occlusion regions are detected based on bidirectional CPF mapping. The MVs of the occlusion pixels are refined using spatial or temporal MV prediction. The intermediate frames are produced by bidirectional interpolation for non-occlusion pixels and by unidirectional projection for occlusion pixels.

The rest of this paper is organized as follows. Section 2 briefly introduces the CPF matching algorithm and the approach to apply CPF matching interpolation (CMI) to FRUC. The proposed ORCMI based FRUC is presented in Section 3. Section 4 discusses the experimental results. Finally, Section 5 concludes this paper.

CMI BASED FRUC

CPF matching algorithm

CPF was first introduced by Shinagawa and Kunii (1998). The filters provide a means of accurately matching the pixels of two images. Suppose there are two images, the source and the destination. A set of multiresolution subimages are constructed for both the two images. Then, mappings from the source to the destination subimages are performed at each level, from the coarsest resolution to the finest resolution. The mapping is computed pixel-by-pixel, constrained by the inherited and the bijectivity conditions. The mapping with minimum energy will be selected as final correspondence.

1. Construction of the multiresolution hierarchy

Suppose the width and the height of the original image size are W and H, respectively, and n denotes the hierarchy level of the finest resolution. A multiresolution hierarchy of size $W/2^{(n-m)}\times H/2^{(n-m)}$ ($1\leq m\leq n$) images will be computed. There are four subimages to be calculated, which are formed by extraction of the minimum, the maximum and the saddle points at each level of the hierarchy. Let $p_{(i,j)}^{(m,s)}$ denote the pixel at (i, j) in the subimage, where m is the level of hierarchy and s is the subimage type. The pixels of subimages in the hierarchy are recursively calculated from the pixels of its higher level subimages in the hierarchy as follows:

$$p_{(i,j)}^{(m,0)}=\min(\min(p_{(2i,2j)}^{(m+1,0)},p_{(2i,2j+1)}^{(m+1,0)}),\ \min(p_{(2i+1,2j)}^{(m+1,0)},p_{(2i+1,2j+1)}^{(m+1,0)})), \tag{1}$$

$$p_{(i,j)}^{(m,1)}=\max(\min(p_{(2i,2j)}^{(m+1,1)},p_{(2i,2j+1)}^{(m+1,1)}),\ \min(p_{(2i+1,2j)}^{(m+1,1)},p_{(2i+1,2j+1)}^{(m+1,1)})), \tag{2}$$

$$p_{(i,j)}^{(m,2)}=\min(\max(p_{(2i,2j)}^{(m+1,2)},p_{(2i,2j+1)}^{(m+1,2)}),\ \max(p_{(2i+1,2j)}^{(m+1,2)},p_{(2i+1,2j+1)}^{(m+1,2)})), \tag{3}$$

$$p_{(i,j)}^{(m,3)}=\max(\max(p_{(2i,2j)}^{(m+1,3)},p_{(2i,2j+1)}^{(m+1,3)}),\ \max(p_{(2i+1,2j)}^{(m+1,3)},p_{(2i+1,2j+1)}^{(m+1,3)})), \tag{4}$$

where $p_{(i,j)}^{(n,0)}=p_{(i,j)}^{(n,1)}=p_{(i,j)}^{(n,2)}=p_{(i,j)}^{(n,3)}=p_{(i,j)}$, which are the pixels of the original image.

These multiresolution hierarchical subimages are computed for both the source and the destination images. Meanwhile, the destination subimages should be up-sampled to provide intensity values at half pixel locations where we use bilinear interpolation.

2. Mapping with inherited condition

Once the multiresolution hierarchy is constructed, a top down method is utilized to map pixels from the source image to the destination image. The number of candidate mappings at each level is constrained by the mapping at its upper level. Fig.1 shows an example, where a pixel p at level m of the source image is searching for its corresponding pixel q in the destination image. Suppose the four nearest pixels of pixel p are a, b, c and d. Their parents (A, B, C, D) are mapped to A', B', C', D' at level $m-1$. For each of the parents, one child pixel (if b is the top left of B, then b' will be the top left of B') is selected. The four children pixels define an inherited quadrilateral *a*′*b*′*c*′*d*′, inside which we search for the pixel *q* with a minimum mapping energy.

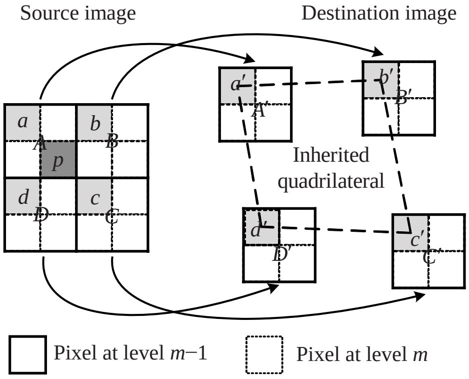


Fig.1 Definition of inherited quadrilateral

3. Bijectivity conditions and mapping energy

The mapping is also constrained by some bijectivity conditions (BCs), which define an elastic motion model. Let $f^{(m,s)}$ be a mapping function of the submapping with type *s* at level *m*. Consider each square *abcd* on the source image plane denoted by an index *R*. The square is mapped by $f^{(m,s)}$ to a quadrilateral *a*′*b*′*c*′*d*′ denoted by $f^{(m,s)}(R)$ on the destination image plane. $f^{(m,s)}(R)$ should satisfy the following conditions (Fig.2):

(1) The edges of $f^{(m,s)}(R)$ should not intersect with each other.

(2) The orientation of the edges of $f^{(m,s)}(R)$ should be the same as that of *R* (clockwise or counterclockwise).

(3) The length of one edge of $f^{(m,s)}(R)$ can be zero to allow retracted mappings, i.e., $f^{(m,s)}(R)$ may be a triangle. It is not allowed, however, to be a pixel or a line segment (figures of area 0).

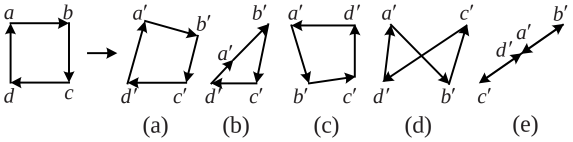


Fig.2 (a) and (b) satisfy the bijectivity conditions, and (c), (d), (e) violate them

Then, the mapping energy of the candidate pixels satisfying the above conditions will be calculated and compared. The energy is defined as $E_{(i,j)}^{(m,s)} = \lambda C_{(i,j)}^{(m,s)} + D_{(i,j)}^{(m,s)}$, where $C_{(i,j)}^{(m,s)}$ is the energy related to pixel intensity, and $D_{(i,j)}^{(m,s)}$ is the energy regarding the distance between the corresponding pixels and the distance between neighboring mappings. The pixel with minimum energy will be determined as the final corresponding pixel. Readers can refer to (Shinagawa and Kunii, 1998; Habuka and Shinagawa, 2004) for more details about the mapping energy in CPF.

CMI in FRUC

For FRUC, CPF mapping is performed between the past frame F_n and the following frame F_{n+1} of the input sequence, denoted by $MAP_{n\to n+1}$. After the mapping is completed, each pixel *p* in F_n has a corresponding pixel $\underset{n\to n+1}{MAP}(p)$ in F_{n+1}, and the MV of pixel *p* can be represented as $\underset{n\to n+1}{MV}(p) = \underset{n\to n+1}{MAP}(p) - p$. With this dense motion field, the intermediate frame F_{n+t} (0<*t*<1) is interpolated as

$$q = (1-t)\cdot p + t\cdot (p + \underset{n\to n+1}{MV}(p)), \tag{5}$$

$$f_{n+t}(q) = (1-t)\cdot f_n(p) + t\cdot f_{n+1}(p + \underset{n\to n+1}{MV}(p)), \tag{6}$$

where *q* and $f_{n+t}(q)$ denote the pixel position and intensity of F_{n+t}, respectively.

An example of CMI based FRUC experiment will be given in Section 4.

ORCMI BASED FRUC

In this section, we analyze the occlusion problem in CPF mapping, and propose an occlusion refinement model based on bidirectional CPF mapping. Then, the ORCMI based FRUC scheme will be described in detail.

Occlusion refinement model in ORCMI

Occlusion regions can be classified as background to be covered (BTBC) and uncovered background (UB) (Altnubasak and Tekalp, 1997). In CPF mapping, each pixel in the source frame has a corresponding pixel in the destination frame. The mapping is not necessarily one-to-one, but can be multi-to-one. A CPF mapping with occlusion is illustrated in Fig.3. For the pixels (p_1, p_5) in the moving object, the mapping procedure can find their correct corresponding pixels (q_1, q_5) in the destination frame. On the contrary, for the pixels (p_2, p_3, p_4) in the BTBC region, they will be mapped to incorrect positions (q_2, q_3, q_4) due to disappearing of the BTBC region. Because of the bijectivity conditions, those incorrect corresponding pixels will converge in the destination frame (Fig.3). Obviously, once the BTBC regions in the source frame are detected, the BTBC regions in the intermediate frame can be obtained through position interpolation like Eq.(5). The roles of BTBC and UB are interchanged in the inverse direction, so the UB region detection can be solved by the backward mapping from the destination frame to the source frame. The occlusion refinement model includes two steps, occlusion detection and MV refinement, which will be discussed individually in the following.

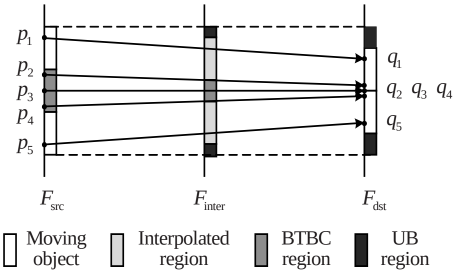
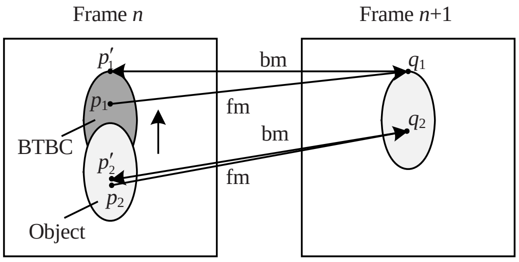


Fig.3 CPF mapping with occlusion

1. Occlusion detection

BTBC region detection is based on two assumptions. First, the intensity difference between the BTBC pixels and their corresponding pixels can be much larger than that of the non-occlusion pixels. The second assumption utilizes bidirectional consistency check, as shown in Fig.4. In the forward mapping, BTBC pixel p_1 in the source frame will be mapped to an inappropriate pixel q_1 in the destination frame. However, in the backward mapping, q_1 is not occluded, and the mapping procedure can find its correct corresponding pixel p_1' in the source frame. The distance between p_1 and p_1' is remarkably larger than that of the non-occlusion pixels, e.g., the distance between p_2 and p_2' in Fig.4. On the basis of these two criterions, BTBC regions in the source frame can be estimated.



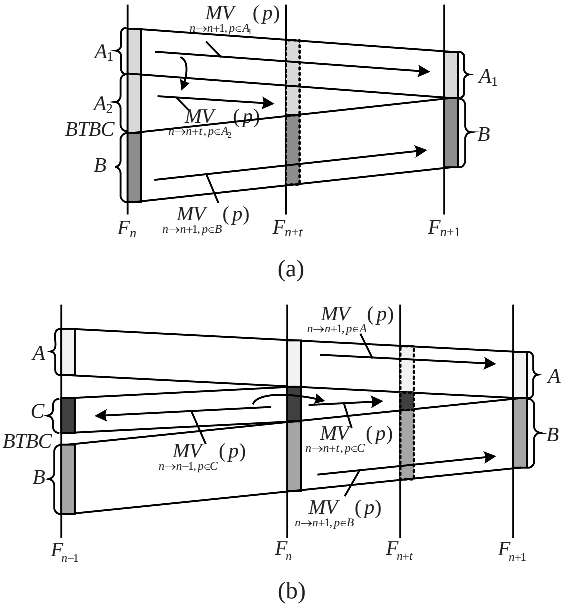
fm: forward mapping; bm: backward mapping

Fig.4 Bidirectional consistency check

2. MV refinement

For the occlusion pixels, their incorrect MVs will be refined using spatial or temporal MV prediction. The two prediction modes are shown in Fig.5, where different gray scales represent different objects.

Fig.5a shows the spatial prediction mode. From F_n to F_{n+1}, objects A and B move with different velocities. A part of object A (A_2) is occluded by object B in F_{n+1}. The pixels in A_2 will be assigned incorrect MVs in the mapping from F_n to F_{n+1}. However, due to the high similarity of MVs from different parts of an object, the MVs of the occlusion pixels can be estimated from their neighboring MVs. For example, in Fig.5a we can use the MVs of A_1 (the non-occluded part of object A) to estimate the MVs of A_2.


$MV_{n\to n+1, p\in A_1}(p)$
A_1
A_2
BTBC
B
$MV_{n\to n+t, p\in A_2}(p)$
A_1
B
$MV_{n\to n+1, p\in B}(p)$
F_n
F_{n+t}
F_{n+1}
$MV_{n\to n+1, p\in A}(p)$
A
C
BTBC
B
A
B
$MV_{n\to n-1, p\in C}(p)$
$MV_{n\to n+t, p\in C}(p)$
$MV_{n\to n+1, p\in B}(p)$
F_{n-1}
F_n
F_{n+t}
F_{n+1}

(a)

(b)

Fig.5 MV refinement of occlusion region. (a) Spatial prediction mode; (b) Temporal prediction mode

Fig.5b shows the temporal prediction mode. The motion of object C in F_n is different from that of its neighboring objects (A or B). C is wholly occluded by A and B in F_{n+1}. In this case, it is unsuitable to estimate the MVs of C from the MVs of A or B. Although C disappears in F_{n+1}, it may be found in the previous frame (F_{n-1}) from the temporal point of view. In general, motion in a short time interval can be considered as linear. Thus, for the pixels in C, we can estimate their forward MVs from F_n to F_{n+t}, using their backward MVs from F_n to F_{n-1}, as shown in Fig.5b.

Implementation of ORCMI based FRUC

Fig.6 shows the overall architecture of the ORCMI based FRUC scheme. It consists of several processing units. First, CPF matching is applied to estimate the MVs of the pixels in input frames. Second, occlusion detection is performed to detect BTBC regions and UB regions based on forward and backward mapping, respectively. Third, the MVs of occlusion pixels are refined using spatial or temporal MV prediction. Finally, bidirectional pixel interpolation for non-occlusion regions and unidirectional pixel projection for occlusion regions are used to generate the intermediate frame.

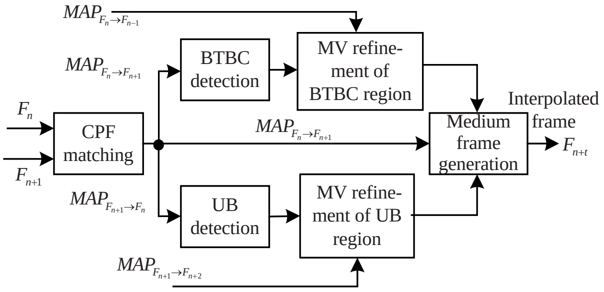


Fig.6 Structure of ORCMI based FRUC scheme

1. CPF matching

Suppose the input frames are F_{n-1}, F_n, F_{n+1}, F_{n+2}, ... and the to-be-interpolated frame is F_{n+t} ($0<t<1$). In the proposed FRUC, we use forward mapping $MAP_{n\to n+1}$ to obtain the initial MVs for all the pixels in F_n. The forward mapping $MAP_{n\to n+1}$ and the backward mapping $MAP_{n+1\to n}$ are used to detect the BTBC regions in F_n and the UB regions in F_{n+1}, respectively. In addition, for temporal MV prediction, the previous mapping $MAP_{n-1\to n}$ may be used to estimate the MVs of the BTBC pixels, and the next mapping $MAP_{n+1\to n+2}$ to estimate the MVs of the UB pixels.

2. Occlusion detection

For each pixel p in F_n, it is mapped to a pixel at $\underset{n\to n+1}{MAP}(p)$ in F_{n+1} in the forward mapping. Then the pixel at $\underset{n\to n+1}{MAP}(p)$ is mapped to another pixel at $\underset{n+1\to n}{MAP}(\underset{n\to n+1}{MAP}(p))$ in F_n in the backward mapping. According to the above two criterions, BTBC region detection consists of the following steps:

Step 1: Compute the predicted frame $\tilde{F}_n$ using $\tilde{f}_n(p)=f_{n+1}(p+\underset{n\to n+1}{MV}(p))$. Then calculate the frame difference (FD), where $FD_n(p)=\left|f_n(p)-\tilde{f}_n(p)\right|$.

Step 2: Calculate the inverse map distance (IMD) frame: $IMD_n(p)=\left\|\underset{n+1\to n}{MAP}(\underset{n\to n+1}{MAP}(p))-p\right\|$.

Step 3: If $FD_n(p)>T_1$ and $IMD_n(p)>T_2$ (T_1 and T_2 are threshold values), mark p as a BTBC pixel, thus the BTBC frame of F_n ($BTBC_n$) is obtained. In this study we use T_1=10 and T_2=1.

Step 4: Apply a morphological opening operation with a 3×3 kernel to $BTBC_n$, followed by a morphological closing operation with the same kernel.

After above operations, the BTBC regions in F_n are detected. Then the BTBC frame of F_{n+t} ($BTBC_{n+t}$) is obtained through position interpolation as Eq.(5). Similarly, the UB frames of F_{n+1} and F_{n+t} (UB_{n+1} and UB_{n+t}) are detected based on the inverse mapping.

3. MV refinement of occlusion pixels

Before MV refinement, the MV prediction mode should be determined for the occlusion pixels. It is natural that different parts of an object have many similar texture properties such as luminance distribution and average intensity. Hence, the MV prediction mode of the occlusion pixel can be determined by measuring the texture similarity between the region surrounding it and the region surrounding its neighboring non-occlusion pixels.

Consider an $N\times N$ (for CIF format, N=5) sized block B_p centered at position p. The average intensity (AVG) and the mean square difference (MSD) of B_p are defined as

$$AVG_p = \frac{1}{N^2}\sum_{j=1}^{N}\sum_{i=1}^{N} f(i,j), \tag{7}$$

$$MSD_p = \frac{1}{N^2}\sum_{j=1}^{N}\sum_{i=1}^{N}\left(f(i,j)-AVG_p\right)^2. \tag{8}$$

The texture similarity between blocks B_p and B_q is defined as

$$\Phi_{pq} = \frac{255}{\left|AVG_p - AVG_q\right|^2 + \left|MSD_p - MSD_q\right|}. \tag{9}$$

For each occlusion pixel p, we search for its eight nearest non-occlusion pixels q_i ($1\le i\le 8$) in eight directions (horizontal, vertical and diagonal), as shown in Fig.7. The texture similarity Φ_{pq_i} ($1\le i\le 8$) between B_p and B_{q_i} will be computed and compared. Suppose q_j is the pixel with the maximum similarity value Φ_{pq_j}. If $\Phi_{pq_j} > \Gamma$ (Γ is threshold value), spatial prediction mode is selected. Otherwise, temporal prediction mode will be selected. In our implementation, we set Γ=1.2 empirically.

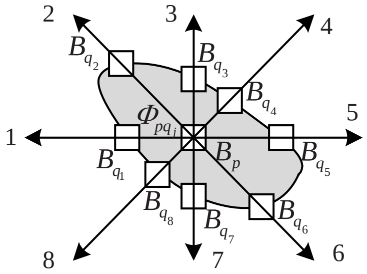


Fig.7 Prediction mode determination

For spatial prediction mode, m (m=3 in this study) pixels with the highest similarity values among the above eight pixels are selected. And the MV of pixel p will be refined as

$$\underset{n\to n+1}{MV}{}'(p) = \sum_{i=1}^{m}\Phi_{pq_i}\underset{n\to n+1}{MV}(q_i)\Bigg/\sum_{i=1}^{m}\Phi_{pq_i}, \tag{10}$$

where Φ_{pq_i} is used as the weight coefficient.

As discussed in the subsection "MV refinement", for temporal MV prediction, the previous and the next mappings are utilized to refine the MVs of the occlusion pixels, which can be formulated as follows:

For $p\in BTBC_n$,

$$\underset{n\to n+1}{MV}{}'(p) = -\underset{n\to n-1}{MV}(p). \tag{11}$$

For $p\in UB_{n+1}$,

$$\underset{n+1\to n}{MV}{}'(p) = -\underset{n+1\to n+2}{MV}(p). \tag{12}$$

4. Intermediate frame generation

Let q denote the pixel position in the to-be-generated frame F_{n+t}, and $f_{n+t}(q)$ denote its intensity. For the non-occlusion pixels in F_{n+t}, their positions and values are interpolated by Eq.(5) and Eq.(6), respectively. For the BTBC pixels in F_n and the UB pixels in F_{n+1}, they are directly projected to the intermediate frame, using the refined MVs as follows:

For $p\in BTBC_n$,

$$q = p + t\cdot\underset{n\to n+1}{MV}{}'(p) = p - t\cdot\underset{n\to n-1}{MV}(p), \tag{13}$$

$$f_{n+t}(q) = f_n(p), \tag{14}$$

where $q\in BTBC_{n+t}$.

For $p\in UB_{n+1}$,

$$q = p + (1-t)\cdot\underset{n+1\to n}{MV}{}'(p) = p - (1-t)\cdot\underset{n+1\to n+2}{MV}(p), \tag{15}$$

$$f_{n+t}(q) = f_{n+1}(p), \tag{16}$$

where $q\in UB_{n+t}$.

EXPERIMENTAL RESULTS

Evaluation tools are also very important for FRUC. In the traditional FRUC evaluation method, the FRUC algorithm is used to interpolate new frames from the original input video at reduced frame rates. Then the original skipped frames can be used for performance evaluation, such as PSNR calculation. However, the purpose of FRUC is to provide output sequences with a higher frame rate. The experiments with input sequences of reduced frame rates cannot truly reflect the performances of FRUC algorithms. In our first experiment, the traditional FRUC evaluation method was adopted. And we compared the performances of CMI and ORCMI with the adaptive overlapped block motion compensation (AOBMC) algorithm (Choi *et al.*, 2007), which is an advanced BMA based FRUC scheme. In the second experiment, a new FRUC evaluation method was developed to compare the performances of CMI and ORCMI. Several standard CIF (352×288) format video test sequences were used, including Foreman, Mobile, Table Tennis, Football, Carphone, and Coastguard. In all of our experiments, t was set as 0.5.

Subjective evaluation of ORCMI algorithm

Fig.8 shows an example of CMI based FRUC experiment. Figs.8a and 8b are the 133rd and 135th frames of Foreman (CIF) sequence, respectively. And their frame difference is shown in Fig.8c. Fig.8d shows the original 134th frame. The interpolated frame by CMI based on the mapping from Fig.8a to Fig.8b is shown in Fig.8e. We can see that blur artifact appears at the hat contour, where occlusion happens. The reason is that the assumption of bijectivity conditions does not hold in the case of occlusion.

Fig.8f shows the interpolated frame by ORCMI in-between Figs.8a and 8b. Comparing Fig.8f with the interpolated frame by CMI (Fig.8e), we can see that the visual quality at the hat contour has been improved after occlusion refinement. Fig.9 shows the interpolated frames of Table Tennis, Football and Mobile sequences. The original frames are shown in the left column. The interpolated frames by CMI and ORCMI are shown in the middle and the right columns, respectively. For Table Tennis and Football sequences, the quality improvement can be easily observed around the table tennis player's hand, the racket, and the football player's body. For Mobile sequence, however, the quality improvement is unobvious because it has much less occlusion than the other two sequences.

Comparison with traditional FRUC algorithm

In this experiment, the traditional evaluation method was adopted. The odd frames in test sequences were skipped. The tested FRUC algorithms were used to double the frame rate. Then the interpolated frames were compared with the skipped

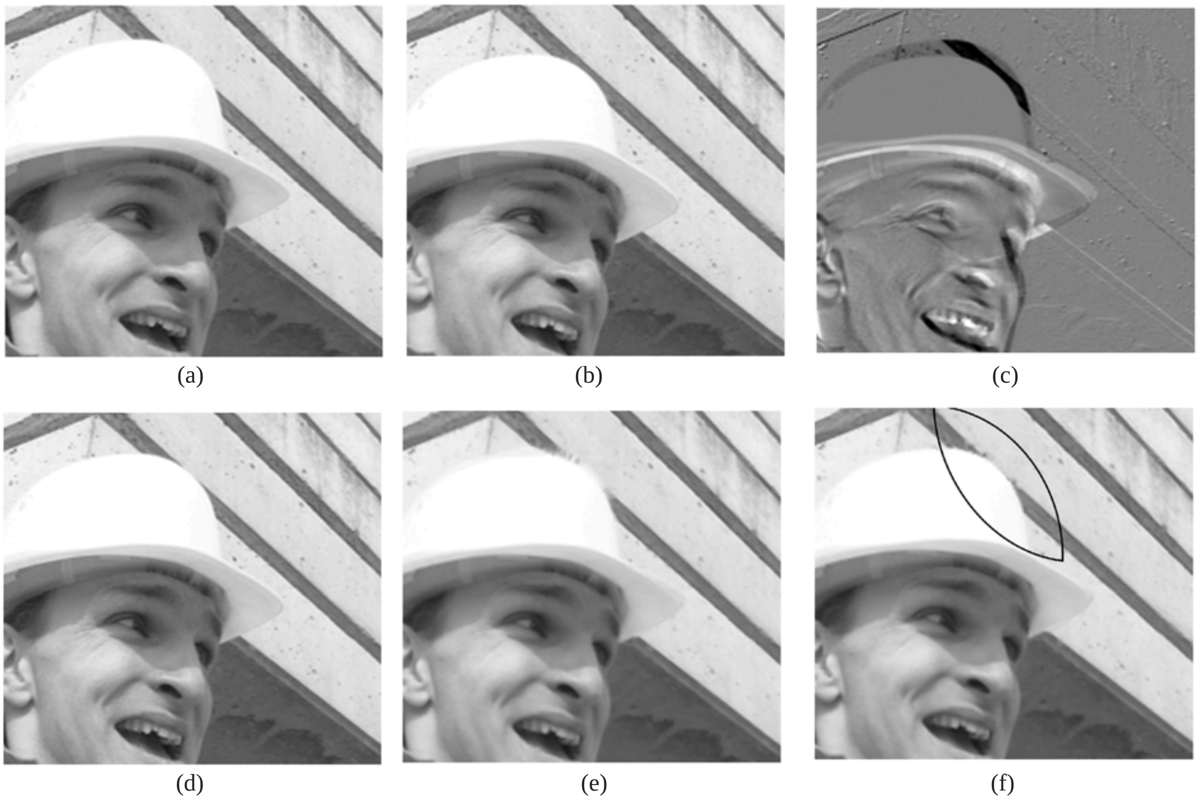


Fig.8 Interpolated frames of the Foreman sequence. (a) The 133rd frame; (b) The 135th frame; (c) Difference frame of (a) and (b); (d) The original 134th frame; (e) Interpolated frame by CMI; (f) Interpolated frame by ORCMI

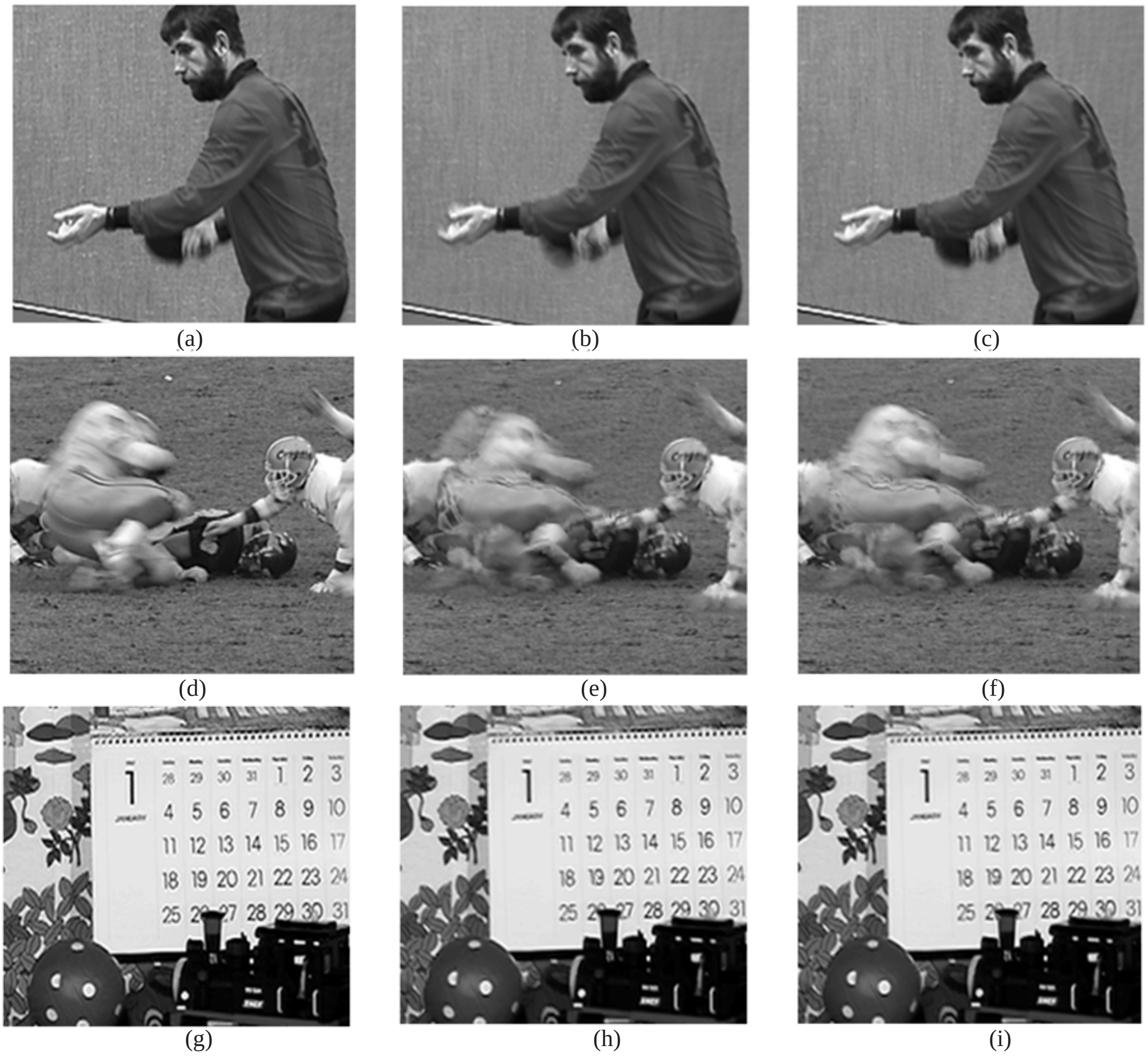


Fig.9 Interpolated frames of Table Tennis, Football and Mobile. (a), (d), (g): the original frames; (b), (e), (h): the interpolated frames by CMI; (c), (f), (i): the interpolated frames by ORCMI

frames for PSNR calculation. Fig.10 shows the PSNR of the first 50 interpolated frames of the four sequences. Table 1 summaries the average PSNR of each test sequence by the three FRUC algorithms. Compared to AOBMC, CMI achieves significant gains in PSNR. And the gains of Mobile and Football sequences are larger than those of the other sequences, because these two sequences have more complex motions. This demonstrates that the CPF matching algorithm can model deformable motions more accurately than BMA. Moreover, for Foreman and Football sequences, ORCMI provides greater PSNR gains, because they have more occlusion areas. It can also be seen that the performance improvement of some frames in Table Tennis sequence is not very obvious. The reason is that the displacement of the rolling ball between some frames of Table Tennis is so large that the elastic model in the CPF matching cannot work well. When compared to the AOBMC algorithm, ORCMI provides 1~2 dB PSNR improvement for the test sequences.

Table 1 Average *PSNR* comparison

Sequence	Average *PSNR* (dB)			Gain (dB)
	AOBMC	CMI	ORCMI	
Foreman	34.15	34.99	35.43	0.44
Mobile	24.18	26.41	26.57	0.16
Table Tennis	29.29	29.98	30.35	0.37
Football	22.84	24.15	24.90	0.75
Carphone	–	32.95	33.21	0.26
Coastguard	–	31.04	31.33	0.29

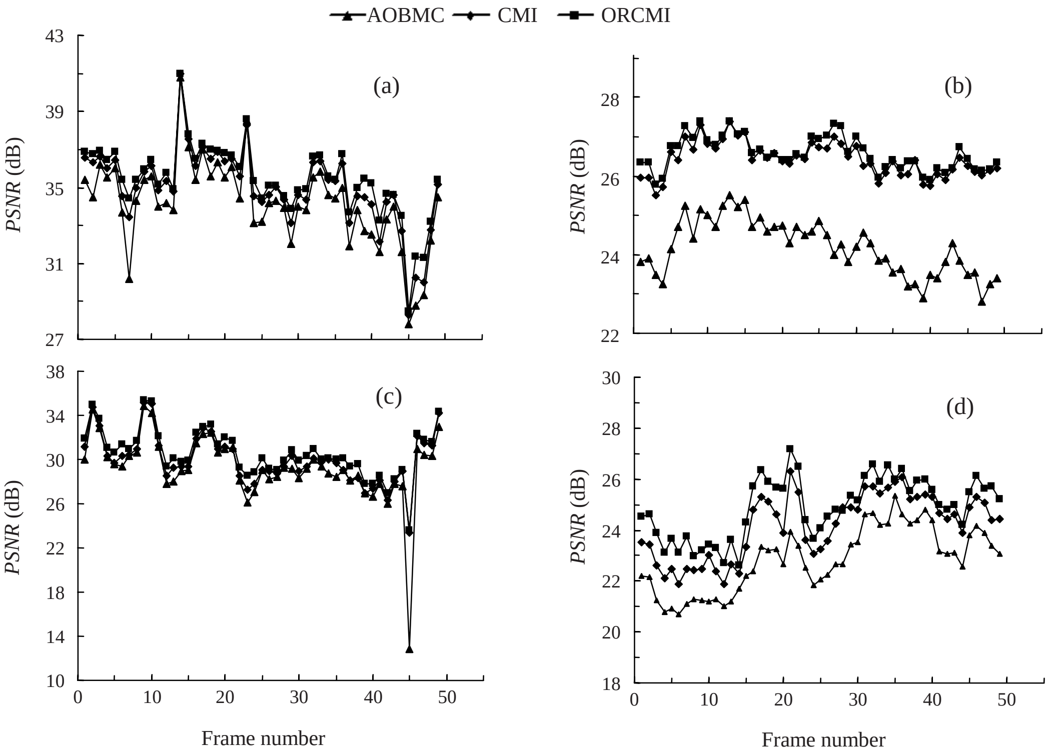


Fig.10 Comparison of the PSNR performance on different sequences
(a) Foreman; (b) Mobile; (c) Table Tennis; (d) Football

Novel evaluation method for FRUC algorithm

In this experiment, we developed a novel evaluation method for FRUC, as shown in Fig.11. First, we used the original sequence to interpolate a medium sequence. Then, the medium sequence was used to interpolate a virtual sequence again. In this way, the frame positions of the virtual sequence keep consistent with that of the original sequence. Finally, PSNR was calculated between the original sequence and the virtual sequence. This evaluation method has two advantages. One is that the input frame rate does not need to be reduced. The other is that the performance difference between different FRUC algorithms will be emphasized through two interpolation procedures. Hence, it is much easier to compare the performances of different FRUC algorithms. Table 2 shows the average PSNR performances of the first 100 frames interpolated by CMI and ORCMI algorithms with the proposed evaluation method. It is noted that the PSNR gain is slightly smaller than the results from the traditional evaluation method, which can be interpreted that the occlusion areas become smaller when the input frame rate is not reduced.

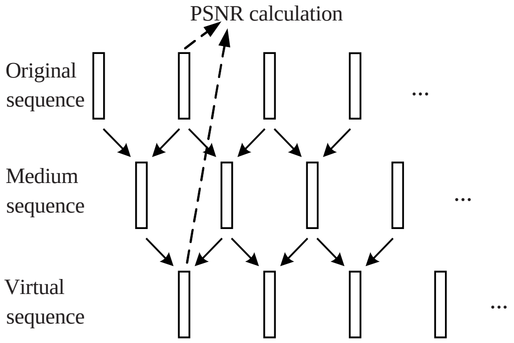


Fig.11 New FRUC evaluation method

Table 2 Average *PSNR* comparison using the proposed evaluation method

Sequence	Average *PSNR* (dB)		Gain (dB)
	CMI	ORCMI	
Foreman	35.62	35.97	0.35
Mobile	25.63	25.74	0.11
Table Tennis	30.57	30.85	0.28
Football	28.37	28.95	0.58
Carphone	35.26	35.48	0.22

We also note that, the proposed evaluation method provides higher PSNR values than the traditional evaluation method for all the test sequences except for Mobile. In natural scenes, motions, especially non-translational motions, are nonlinear. As a result, linear interpolation will lead to position deviation between the frames of the virtual sequence and the frames of the original sequence. Mobile sequence contains more non-translational motions, thus its position deviation must be larger than that of the other sequences.

CONCLUSION

In this paper, we introduce CPF method to FRUC. An occlusion refinement model is proposed to tackle the occlusion problem in CPF matching interpolation (CMI). Then, we develop a new FRUC scheme using occlusion refined CPF matching interpolation (ORCMI). We compare the performance of the ORCMI algorithm with the AOBMC method. Experimental results demonstrate that the occlusion refined technique improves the visual quality of the reconstructed frames, especially at the boundaries of moving objects where occlusion happens. Compared to AOBMC, ORCMI achieves 1~2 dB gains in PSNR. Moreover, to truly reflect the performances of FRUC algorithms, we propose a novel evaluation method, which can be easily applied to compare the performances of various FRUC algorithms. Although the experiments were mainly for doubling the input frame rate, the proposed ORCMI algorithm can be easily extended to various applications, such as film-to-video conversion and reconstruction of skipped frames in video communication.

References

Altnubasak, Y., Tekalp, A.M., 1997. Occlusion-adaptive, content-based mesh design and forward tracking. *IEEE Trans. on Image Processing*, **6**(9):1270-1280. [doi:10.1109/83.623190]

Beller, E.B., van Gurp, J.W., Janssen, J.G.W.M., Braspenning, R., Wittebrood, R., 2007. Solving Occlusion in Frame-rate Up-conversion. IEEE Int. Conf. Consumer Electronics, p.1-2. [doi:10.1109/ICCE.2007.341494]

Castagno, R., Haavisto, P., Ramponi, G., 1996. A method for motion adaptive frame rate up-conversion. *IEEE Trans. on Circuits Syst. Video Technol.*, **6**(5):436-446. [doi:10.1109/76.538926]

Chambers, B., Durand, J., Gans, N., Hutchinson, S., 2003. Dynamic Feature Point Detection for Visual Servoing Using Multiresolution Critical-point Filters. Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems, p.504-509. [doi:10.1109/IROS.2003.1250679]

Choi, B.D., Han, J.W., Kim, C.S., Ko, S.J., 2006. Frame rate up-conversion using perspective transform. *IEEE Trans. on Consum. Electron.*, **52**(3):975-982. [doi:10.1109/TCE.2006.1706496]

Choi, B.D., Han, J.W., Kim, C.S., Ko, S.J., 2007. Motion-compensated frame interpolation using bilateral motion estimation and adaptive overlapped block motion compensation. *IEEE Trans. on Circuits Syst. Video Technol.*, **17**(4):407-416. [doi:10.1109/TCSVT.2007.893835]

Durand, J., Hutchinson, S., 2003. Real-time Object Tracking Using Multiresolution Critical Point Filters. Proc. IEEE Int. Conf. Robots and Automation, **2**:1682-1687.

Ha, T., Lee, S., Kim, J., 2004. Motion compensated frame interpolation by new block-based motion estimation algorithm. *IEEE Trans. on Consum. Electron.*, **50**(2):752-759. [doi:10.1109/TCE.2004.1309458]

Haavisto, P., Juhola, J., Neuvo, Y., 1989. Fractional frame rate up-conversion using weighted median filters. *IEEE Trans. on Consum. Electron.*, **35**(3):272-278. [doi:10.1109/30.44281]

Habuka, K., Shinagawa, Y., 2004. Image interpolation using enhanced multiresolution critical-point filters. *Int. J. Comput. Vision*, **58**(1):19-35. [doi:10.1023/B:VISI.0000016145.44583.5f]

Lee, S.H., Shin, Y.C., Yang, S.J., Moon, H.H., Park, R.H., 2002. Adaptive MC interpolation for frame rate up-conversion. *IEEE Trans. on Consum. Electron.*, **48**(3):444-450. [doi:10.1109/TCE.2002.1037026]

Sanchez, O., 2003. Tracking of an Occluded Object in a Video Sequence. 7th Int. Symp. Signal Processing and Its Application, **2**:231-234. [doi:10.1109/ISSPA.2003.1224856]

Shinagawa, Y., Kunii, K.L., 1998. Unconstrained automatic image matching using multiresolutional critical-point filters. *IEEE Trans. on Pattern Anal. Machine Intell.*, **20**(9):994-1010. [doi:10.1109/34.713364]

Wittebrood, R.B., de Haan, G., Lodder, R., 2003. Tackling Occlusion in Scan Rate Conversion Systems. IEEE Int. Conf. Consumer Electronics, p.344-345. [doi:10.1109/ICCE.2003.1218963]

Yang, Y.T., Tung, Y.S., Wu, J.L., 2007. Quality enhancement of frame rate up-converted video by adaptive frame skip and reliable motion extraction. *IEEE Trans. on Circuits Syst. Video Technol.*, **17**(12):1700-1713. [doi:10.1109/TCSVT.2007.903806]

Zhai, J., Yu, K., Li, J., Li, S., 2005. A Low Complexity Motion Compensated Frame Interpolation Method. Proc. IEEE Int. Symp. on Circuits and Systems, **5**:4927-4930. [doi:10.1109/ISCAS.2005.1465738]