Computer Vision and Image Understanding 168 (2018) 64–78

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu

Modified non-local means for super-resolution of hybrid videos

Yawei Li*[a,b], Xiaofeng Li[a], Zhizhong Fu[a]

[a] University of Electronic Science and Technology of China, Chengdu, China
[b] ETH Zurich, Zurich, Switzerland

ARTICLE INFO

Keywords:
Adaptive parameters
Hybrid video
Non-local means
Non-local self similarity
Super-resolution

ABSTRACT

Hybrid videos that contain periodic low-resolution (LR) frames and high-resolution (HR) guide frames are largely used for the consideration of bandwidth efficiency and the tradeoff between spatial and temporal resolution. Super-resolution (SR) algorithms are necessary to refine the LR frames, in which non-local means (NLM) is a promising algorithm. NLM replaces every pixel with a weighted average of its neighbors based on non-local self-similarity between pixels. However, the fixed decaying factor of NLM cannot satisfy regions of distinct characteristics in LR frames. The fixed neighborhood or the so-called searching window fails to balance the requirements of low computation and advanced video quality. In this paper, we propose novel criteria to choose the parameters adaptively. The decaying factor is defined by patch difference of a pixel and guarantees NLM to find relevant pixels. Two methods, namely a predefined method inspired by motion estimation and an exhaustive method by searching progressively enlarged neighborhood are proposed to determine the neighborhood size. Bilateral adjacent HR guide frames are used to handle the occlusion problem. We also analyze the defined patch difference on pixel-, frame-, and sequence-level and reveal its influence on the algorithm. The experimental results verify the validity of the proposed method.

1. Introduction

Recently, hybrid video or the so-called reversed-complexity video coding (Brandi et al., 2008) and inconsistent scalable video streaming (Mahfoodh et al., 2015), which is a low-resolution (LR) video with periodic high-resolution (HR) frames, has been studied out of different considerations, applied under distinct scenarios, and enhanced by several super-resolution (SR) algorithms. On the one hand, from the perspective of video compression, the use of hybrid video (see Fig. 1) can not only reduce video's data size and improve the efficiency of bandwidth usage but also reduce encoding complexity. Multi-view mixed-resolution video originates from the same purpose (Garcia et al., 2012; Jin et al., 2015; Li et al., 2016a; 2016c; Richter et al., 2015). Mukherjee et al. adopted this concept and proposed a resolution reduction based coding mode in existing codecs where computational complexity was transferred from encoder side to the decoder side (Mukherjee et al., 2007). Brandi et al. directly proposed the use of hybrid video for the purpose of data-size reduction (Brandi et al., 2008).

On the other hand, the tradeoff between spatial and temporal resolution also leads to the incorporation of hybrid video (Ben-Ezra and Nayar, 2003; Tai et al., 2010). In order for a pixel to be detected, an image sensor needs a minimum exposure time to accumulate sufficient irradiance. Providing that the sensor size is constant, the footprint of every pixel on the sensor will be reduced with the increase of image spatial resolution, which means exposure time should be prolonged to accumulate the same amount of irradiance on a reduced pixel footprint. Hybrid cameras can simultaneously capture periodic HR snapshots with low rate and LR frames with high frame rate. Commercial cameras such as Canon EOS 500D and Sony HDR-SR11 support this application (Ancuti et al., 2010; Basavaraja et al., 2010).

To enhance the LR frames of a hybrid video, Brandi et al. proposed a motion estimation based SR method where HF components in the HR frames were used to recover those in the LR frames (Brandi et al., 2008). Song et al. further used hierarchical motion estimation to obtain as accurate motion vectors as possible and employed example-based SR when motion estimation failed (Song et al., 2011). Mahfoodh et al. utilized quad tree structure based motion estimation and incorporated their algorithm in VP9 spatial SVC (Mahfoodh et al., 2015; Mukherjee et al., 2013). All of the aforementioned algorithms recovered an LR patch using only one HR patch derived from motion estimation, which limited the algorithms' performance. Thus, Hung et al. developed example-based SR, by searching and combining multiple HR patches in codebooks derived from key frames, to super-resolve an LR patch (Hung et al., 2012). Bevilacqua et al. recovered an LR patch by taking sparse combinations of patches found in the adjacent HR frames (Bevilacqua et al., 2013).

* Corresponding author.
E-mail addresses: Li_Yawei@outlook.com, lywuestc@163.com (Y. Li), xfli@uestc.edu.cn (X. Li), fuzz@uestc.edu.cn (Z. Fu).

https://doi.org/10.1016/j.cviu.2017.11.010
Received 14 February 2017; Received in revised form 21 October 2017; Accepted 25 November 2017
Available online 06 December 2017

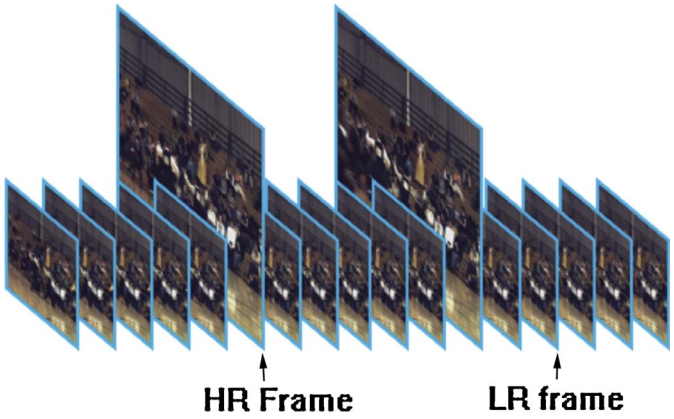


Fig. 1. Illustration of hybrid video.

1.1. Related work

The existing SR algorithms can be classified into three categories, namely interpolation-based, reconstruction-based, and example-based.

Basic interpolation-based methods such as bilinear or bicubic interpolation are based on the smoothness assumption of natural images. But they tend to blur the derived images especially at edges. Advanced approaches belong to a class of visually oriented interpolation techniques, including edge directed, content adaptive, and wavelet-based methods (Allebach and Wong, 1996; Li and Orchard, 2001; Wang and Ward, 2007; Zhang and Wu, 2006). However, the video sequences refined by interpolation suffer from perceived loss of detail in texture regions because they are unable to estimate HF information.

Reconstruction-based methods use subpixel shifts among several LR images of the same scene taken from multiple viewpoints. By estimating these shifts, pixels are rearranged into an HR grid and combined to complete an HR image (Bose and Ahuja, 2006; Farsiu et al., 2004; Takeda et al., 2007). Iterative back projection (IBP) recovers a final HR image by projecting the reconstruction error between the LR and intermediate HR images back to the HR image iteratively (Gan et al., 2013; Zhang et al., 2016). Maximum a posteriori probability (MAP) methods utilize Bayesian statistical properties of images and adopt prior information such as total variation, Tikhonov regularization (Fu et al., 2016a), and non-local prior (Zhang et al., 2012) to stabilize the solution. However, due to the limited information available, reconstruction-based methods hit a bottleneck in improving the recovered image quality.

Example-based algorithms use known HR images to build a database which consists of pairs of LF information and HF information in a training phase (Freeman et al., 2001; Timofte et al., 2016; Wang et al., 2016; Yin et al., 2015). Then the established database guides the learning phase to search a matching HR block for every block in the LR image. Under the paradigm of learning, some other algorithms adopt classical image processing techniques such as convolutional neural networks (CNN) (Kim et al., 2016), sparse representation (Polatkan et al., 2015), and neighborhood embedding (Chang et al., 2004) to address SR problems. These algorithms achieve state-of-the-art performances. For video SR, motion estimation plays a key role in making up for the motion between consecutive video frames (Brandi et al., 2008; Hung et al., 2012; Song et al., 2011). Thus, Liu and Sun proposed a Bayesian framework for video SR simultaneously estimating motion, blur kernel and noise level (Liu and Sun, 2014). However, the iterative procedure caused heavy computational burden. Therefore, Liao et al. solved the problem by employing a non-iterative method based on deep draft-ensemble learning (Liao et al., 2015). Kappeler et al. used explicit adaptive motion compensation as a preprocessing for video frames before they were fed into CNN framework (Kappeler et al., 2016). Other exampled-based video SR can be found in Dai et al. (2017); Huang et al. (2017); Shi et al. (2016).

Apart from the concept of learning, non-local self-similarity is another important concept for addressing image processing problems. The basic idea originates from the observation that similar image patches usually reproduce within the range of a natural image (Buades et al., 2005), its derivatives (Gilboa and Osher, 2008; Zhang et al., 2010), or even its sparse coding coefficients (Dong et al., 2013). Buades et al. first proposed a non-local means (NLM) filter used as an image denoising filter (Buades et al., 2005). Inspired by this idea, Kostadin et al. assigned image patches into three-dimensional (3D) groups according to the non-local self-similarity between them and devised a block-matching and 3D (BM3D) filter (Dabov et al., 2007). Some researchers also incorporated non-locality into a variational framework and proposed non-local total variation which was widely applied in image inpainting (Gilboa and Osher, 2008), motion estimation (Werlberger et al., 2010), and image SR (Dong et al., 2013; Ren et al., 2017).

Protter et al. first generalized NLM to SR from the viewpoint of error energy minimization (Protter et al., 2009). Basavaraja et al. combined the work in Brandi et al. (2008) and Protter et al. (2009) to compute the HF part of a pixel using NLM (Basavaraja et al., 2010). Lengyel et al. incorporated illuminance and gradient information into the similarity comparison and reduced the averaging pixels by thresholding (Lengyel et al., 2014).

1.2. Motivation and contribution

The classical NLM algorithm has two major steps. First, it compares the similarity between a pixel and its neighbors and assigns weights to these neighbors. The weight is some inverse function of the Euclidean distance between the patches surrounding two pixels. In the second step, NLM replaces every pixel with a weighted average of its neighbors. To adapt to SR tasks, the algorithm is altered in the second step where it only computes the HF part of a pixel by a weighted average of its neighbors' HF parts. Then the derived HF part is added to the interpolated LR frames to complete the SR processing.

In this paper, we focus on two major parameters of NLM for video SR including the decaying factor used to compute the weights and the size of the neighborhood (searching window) within which NLM searches to find similar pixels to the target pixel. However, the fixed decaying factor of NLM cannot satisfy regions of distinct characteristics in an LR frame. And the fixed searching window fails to balance the requirements of low computational complexity and high quality of super-resolved images. Thus, we propose a novel criterion to select the decaying factor adaptively. We also propose two methods to adaptively determine the size of searching window, namely the predefined searching window (Li et al., 2016b) and the exhaustive searching window (Li et al., 2016d). The predefined method is a preprocessing implemented before NLM which is inspired by motion estimation but more efficient to carry out than motion estimation. The exhaustive method is incorporated during the process of NLM and determines the window size by searching progressively enlarged window iteratively until local difference drops below a termination standard.

The following of the paper is organized as follows. Section 2 discusses hybrid videos and basic NLM algorithm. We also define patch difference, local difference, and global difference in this section. Section 3 explains the proposed algorithm. Section 4 analyzes the defined path difference on three levels, namely pixel-, frame-, and sequence-level and reveals its influence on NLM. Section 5 shows the experimental results. Section 6 concludes the paper.

2. Preliminary discussion and basic model

2.1. Hybrid video

In the image and video acquisition process, an image is usually

degraded by several processes such as blurring, decimation, and noise corruption, i.e.,

$$Y = DBX + n \tag{1}$$

where X is the ground-truth image of the actual scene, Y is the degraded image, B stands for blurring, D stands for decimation, and n is usually independent Gaussian noise.

For the hybrid video shown in Fig. 1, the basic LR frames and periodic HR frames follow distinct degradation models. The periodic HR frames are the ground-truth images of the actual scene. Thus, no degradation process is included. From the perspective of bandwidth usage, the main consideration is to reduce the data size of an HR video without introducing much degradation. Thus, it's usually assumed that LR frames only undergo blurring, and decimation but no noise corruption, leaving the LF components available and reliable. In this scenario, the main SR task is to recover the missing HF components of the LR frames.

2.2. Separation of HF and LF components

Before applying SR algorithms, the first step is to separate the HF and LF components of a hybrid video. Since the interpolated images of LR frames can act as reliable LF components, one only needs to recover the missing HF components, which also avoids the flickering or jerkiness effects in some applications where LF components are not available such as virtual view synthesis (Fu et al., 2016b; Zhao et al., 2011) and frame rate up-conversion (Choi et al., 2000; Wang et al., 2010). Inspired by example-based SR method (Freeman et al., 2001), Brandi et al. first proposed the method of separating HF and LF components (Brandi et al., 2008).

In this paper, the periodic HR frames and LR frames of a hybrid video are denoted by

$$\{F_k | k = Tz,\ z \in \mathbb{N}\} \tag{2a}$$

$$\{f_n | n = Tz + r,\ z \in \mathbb{N},\ r = 1, 2, ..., T-1\} \tag{2b}$$

where F_k is an HR frame, f_n an LR frame, and T the period of HR frames. In the sequel discussion, we will continue to use upper case F to denote all the intermediate results of HR guide frames and lower case f to denote those of LR frames. The superscript of F and f discriminates HF and LF components. The HR frames are blurred, decimated, and interpolated to generate their LF parts, namely,

$$F_k^L = UDBF_k \tag{3}$$

where B, D, and U are the blurring, decimation, and interpolation operation, F_k^L is the LF part of F_k. Then F_k^L is subtracted from F_k, resulting in the HF detail of F_k, namely,

$$F_k^H = F_k - F_k^L. \tag{4}$$

The LR frames are also scaled to the same resolution as the HR frames, namely,

$$\widetilde{f}_n^L = Uf_n. \tag{5}$$

F_k^L and $\widetilde{f}_n^L$ contain the basic structure information and are used to compute the weights in NLM algorithm. F_k^H is used to recover the missing details of $\widetilde{f}_n^L$.

2.3. Framework of NLM for SR

In the image restoration tasks, in order to recover a pixel, additional information from relevant pixels is necessary. Thus, SR algorithms need a relevant pixel or patch selection process which is based on patch similarity. Since there is motion between consecutive frames of a video, motion estimation is the first technique that comes into mind to play a role in similarity comparison. However, motion estimation and compensation recovers an image patch by patch and only selects one target patch with the minimum Euclidean distance to the anchor patch. Thus, the performance of the algorithm (Brandi et al., 2008) is limited.

As a result, NLM arises as a new similarity comparer. Contrary to motion estimation and compensation, NLM relies on pixel-wise computation and recovers a pixel by a weighted average of its similar neighbors. In fact, NLM can also be regarded as an implicit motion estimator (Protter et al., 2009).

NLM exploits the redundancy of a video to recover a pixel. The HF part of an LR pixel is calculated by the weighted average of its HR neighbors' HF parts, namely,

$$\widetilde{f}_n^H(x, y) = \frac{\sum_{(i,j)\in\Omega_{xy}} \omega_{x,y}(i, j) \cdot F_k^H(i, j)}{\sum_{(i,j)\in\Omega_{xy}} \omega_{x,y}(i, j)} \tag{6}$$

where Ω_{xy} is the neighborhood (searching window of size $\mathcal{N} \times \mathcal{N}$) of the LR pixel (x, y) in the HR frame, (i, j) is a pixel in Ω_{xy}. The weight assigned to pixel (i, j) reflects the similarity between (i, j) and (x, y) and is computed as

$$\omega_{x,y}(i, j) = \exp\left(\frac{-\|(R_{x,y}^S \widetilde{f}_n^L - R_{i,j}^S F_k^L) G_{\sigma_s}\|_2^2}{2\sigma^2}\right) \tag{7}$$

where σ is a fixed decaying factor, $R_{i,j}^S$ is an operator that extracts a patch of size $S \times S$ centered at (i, j), and G_{σ_s} is a two-dimensional Gaussian kernel with 0 mean, variance σ_s^2, and size $S \times S$. The difference between the two extracted patches is multiplied pixel-wise by the Gaussian kernel. The kernel penalizes pixels off the center based on pixel coordinates, i.e.,

$$G_{\sigma_s}(x, y, i, j) = \frac{1}{2\pi\sigma_s^2} \exp\left(-\frac{(x-i)^2 + (y-j)^2}{2\sigma_s^2}\right). \tag{8}$$

At last, the HF detail $\widetilde{f}_n^H(x, y)$ and LF structure $\widetilde{f}_n^L(x, y)$ are added to form the recovered pixel value $\widetilde{f}_n(x, y)$, namely,

$$\widetilde{f}_n(x, y) = \widetilde{f}_n^H(x, y) + \widetilde{f}_n^L(x, y). \tag{9}$$

The framework of NLM for SR is shown in Fig. 2.

Here we make several definitions for the convenience of the sequel discussion. The patch difference between pixel (x, y) and (i, j) is defined by the Euclidean distance between the patches surrounding them

$$E_{x,y}(i, j) = \|(R_{x,y}^S \widetilde{f}_n^L - R_{i,j}^S F_k^L) G_{\sigma_s}\|_2^2. \tag{10}$$

The local difference of the LR pixel (x, y) is defined as the minimum patch difference within its neighborhood, i.e.,

$$E(x, y) = \min_{(i,j)\in\Omega_{xy}} \{E_{x,y}(i, j)\}. \tag{11}$$

The global difference of an LR frame is defined as the maximum local difference across the whole image, i.e.,

$$E_{global} = \max_{(x,y)} \{E(x, y)\}. \tag{12}$$

The defined patch difference, local difference, and global difference measure the features of an image at different levels. To super-resolve an LR pixel, the focus is on its most similar neighbors with small patch differences because NLM needs to assign larger weights to those pixels. The Euclidean distance between an LR pixel and its most similar neighbors becomes vital because it can indicate the quality of a recovered pixel. Thus, local difference arise as a descriptor to measure how similar an LR pixel is to its neighbors. Although the local differences in an image vary within a large range, we observe that the maximum local difference is related to the motion behavior of a video. The larger the maximum local difference, the larger the motion of a video. Thus, global difference is defined as the maximum local difference to measure this behavior.

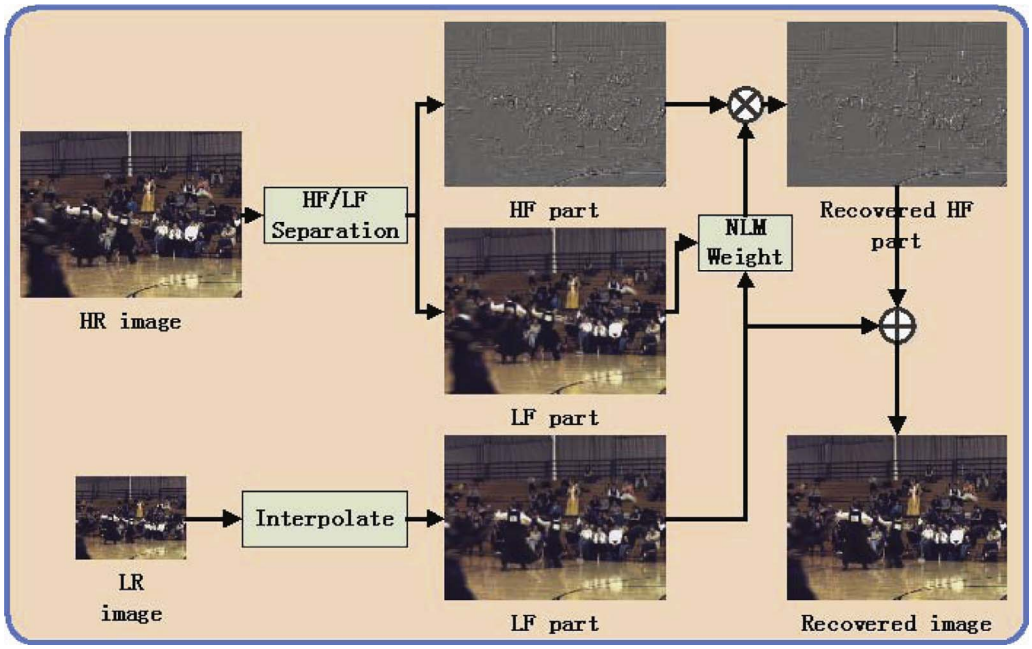


Fig. 2. Framework of NLM for SR.

3. The proposed algorithm

3.1. Bilateral video super-resolution

NLM searches in the neighborhood of a center pixel to find similar pixels it. The method can be considered as a coarse and implicit motion estimator.

Generally, a single frame in a video sequence can be divided into background and foreground objects. The background is usually stable or moves slowly and it is enough to super-resolve the background pixels in an LR frame using a forward or backward HR frame. However, the foreground object may move fast so that an object around the boundary moves in or out of the current scene. Thus, if only one HR frame is used to refine an LR frame, mismatches between pixels will occur and the effectiveness of NLM will be adversely affected. Fig. 3 clearly shows this problem. The pixels of the moving dancers in Frame 9 have no correspondences in Frame 6. Thus, it is impossible to refine these pixels using only Frame 6. This problem can be solved by including both Frame 6 and Frame 12 into the SR algorithm.

Therefore, we use bilateral adjacent frames to ease the motion problem. For every LR frame, its bilateral adjacent HR frames are

$$\{F_b | b \in \Phi\} \tag{13}$$

where $\Phi = \{\lfloor n/T \rfloor \times T, \lceil n/T \rceil \times T\}$, $\lfloor \cdot \rfloor$ is the round-down operator, and $\lceil \cdot \rceil$ the round-up operator.

The above discussion leads to the conclusion that the bilateral SR algorithm using both forward and backward frames outperforms the normal method using only one frame. Of course, using more spatial and temporal adjacent frames will improve the SR results further. However, the computation cost should be considered in that case.

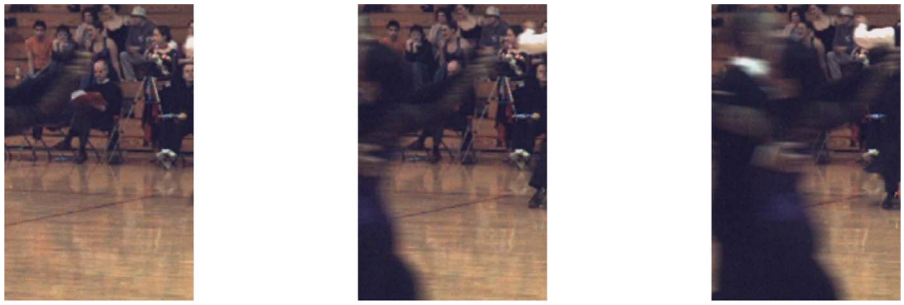
Bilateral SR is somewhat similar to motion-compensated frame rate up-conversion (MC-FRC) (Choi et al., 2000; Wang et al., 2010) in that both techniques use information in the adjacent frames to recover a middle frame. However, they differ mainly in two aspects. Firstly, bilateral SR refines an LR frame using HF information while MC-FRC aims at generating a non-existent frame. Secondly, compared with MC-FRC, bilateral SR doesn't rely on explicit motion estimation.

3.2. Adaptive decaying factor

The decaying factor has a profound influence on the performance of NLM. However, it is not very easy to select a uniform decaying factor for all of the pixels in an image. Here we detail the dilemma about the choosing of decaying factor by comparing the weights of two neighbors of an LR pixel. One of them is with minimum patch difference, namely local difference $E(x, y)$ while the other is an arbitrary pixel with patch difference $E_{x,y}(p, q)$. We compare the two neighbors by the quotient of their weights, i.e.,

$$\eta = \frac{\exp[-E_{x,y}(p, q)/2\sigma^2]}{\exp[-E(x, y)/2\sigma^2]} = \exp(-A/2\sigma^2) \tag{14}$$

where $A = E_{x,y}(p, q) - E(x, y)$ and A is larger than 0. Thus, keeping A



(a) (b) (c)

Fig. 3. Parts of (a) Frame 6, (b) Frame 9, and (c) Frame 12 of *Ballroom*. Frame 6 and Frame 12 are HR guide frames and Frame 9 is the frame to be super-resolved. The aim is to recover Frame 9 using the HF information from Frame 6 and Frame 12.

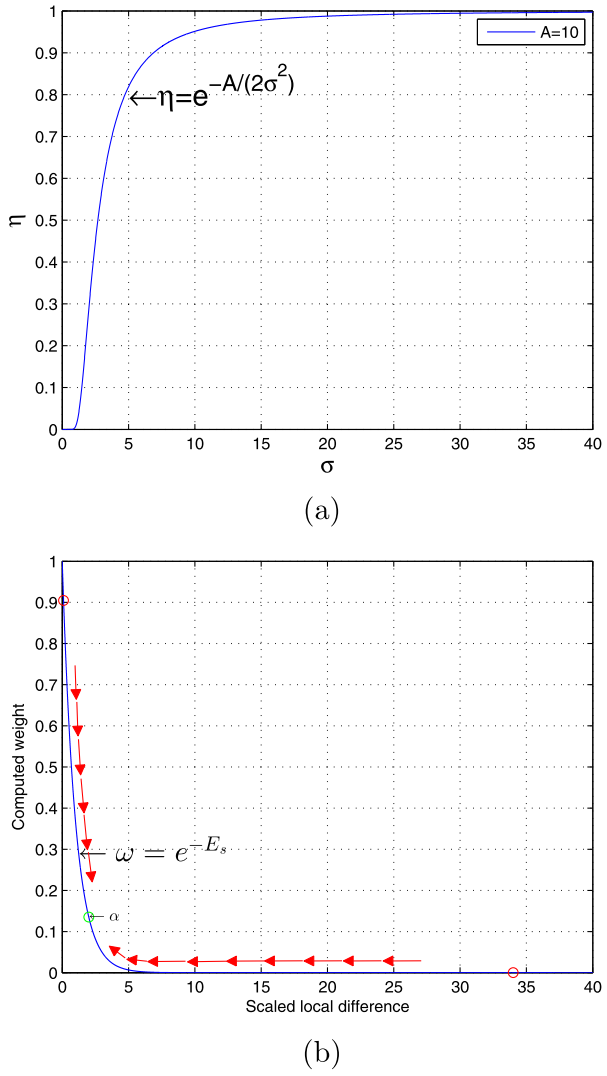


Fig. 4. (a) The relationship between η and σ (b) choosing of adaptive decaying factor.

as a constant, the quotient η is a monotonically increasing function of the decaying factor σ. Fig. 4(a) shows the $\sigma - \eta$ curve. If σ is large, η approaches unit and all the weights tend to be the same, which is an undesirable behavior because NLM acts as a mean filter in this case. One the other hand, a very small σ leads to a small quotient. As a result, only the pixel with local difference has a principle weight close to 1 after normalization. This is also undesirable because in this case NLM becomes so selective that it tends to use only one pixel to recover an LR pixel. In the meanwhile, if the decaying factor is small, the weights will be close to 0. In the worst case, all the weights decays to 0.

Due to the different (smooth and fast) motion characteristics of image regions, local difference changes within a large range, say from 0.1 for background to 50 for moving foreground object. That is, for slow-moving background between consecutive video frames, small local difference features the smooth motion. The same is true for foreground objects. In the following, we discuss two extreme situations with varying local difference which correspond to the case where the decaying factor σ in (14) is fixed but A changes. Assume the local differences scaled by $2\sigma^2$ ($E/2\sigma^2$) for a background pixel and a moving object pixel are $E_S^{B1} = 0.1$ and $E_S^{F1} = 34$. Two other neighbors of the two pixels have the scaled patch differences $E_S^{B2} = 0.2$ and $E_S^{F2} = 36$. If a fixed decaying factor is used, the computed weights of the background pixel and its neighbor are almost equal (0.90 and 0.82 resp.) although the scaled patch difference of the latter is twice of the former ($E_S^{B2}/E_S^{B1} = 2$). This multiple relationship between patch differences discriminates inaccurate and accurate matches for background pixels and thus should be kept in the computed weights. On the other hand, for the moving object pixel, the found neighbor with scaled patch difference $E_S^{F2} = 36$ should be classified as an accurate match since E_S^{F2} is comparable with the scaled local difference ($E_S^{F2}/E_S^{F1} = 1.06$). However, the computed weights discriminates the two pixels.

In the above discussion, when judging whether a neighbor is accurate, the value of scaled local difference is used as a baseline for comparison. If the comparison baseline in the above two cases are slid respectively from 0.1 and 34 to a moderate value say 2 by choosing adaptive decaying factors (see Fig. 4(b)), the desired features of patch difference in the two extreme cases are kept in the computed weight. For example, the computed weights corresponding to E_S^{B1} and E_S^{B2} are 0.368 and 0.135, which distinguish the two neighbors. Thus, the selection criterion for adaptive decaying factor here is to force the comparison baseline, i.e., local difference $E(x, y)$, after smoothed by $2\sigma^2$, to decay to a predefined value α, namely,

$$E(x, y)/2\sigma_A^2 = \alpha. \tag{15}$$

Solving the above equation results in

$$\sigma_A = \sqrt{E(x, y)/2\alpha}. \tag{16}$$

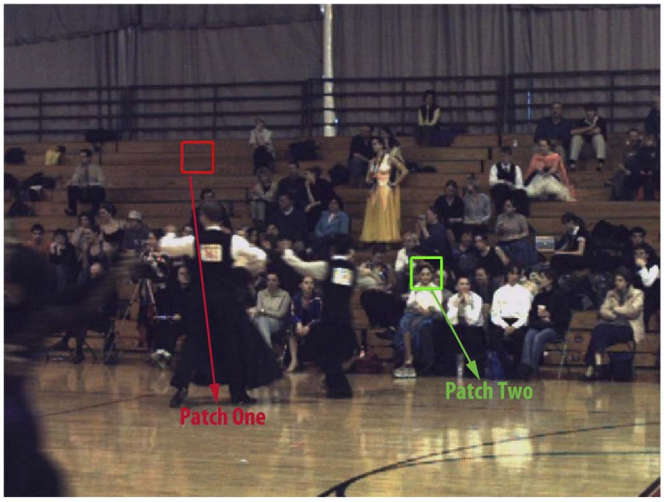
This adaptive decaying factor incorporates local difference as expected. The advantage of this trick is that it meets the requirement of distinguishing between accurate and inaccurate matches according to the value of local difference. The added computation is almost insignificant compared with that of the whole algorithm. What's more, the estimation of α is more robust than that of σ because one needs to use different σ for different videos but a fixed α can already achieve satisfying performance. In this paper, we tested different values of α and found that values around 2 work out best. Thus, we use $\alpha = 2$ in all the experiments. Note that the local difference may be zero. This means that an exactly identical pixel is found in the neighborhood of a pixel. In this case, one can directly set the weight of the identical pixel to 1 and all the other weights to 0.

As is said, the fixed decaying factor is not suitable for regions with different characteristics. This phenomenon becomes especially obvious when Gaussian kernel is not used in the computing process of the weights. Thus, Fig. 5 gives an example under this circumstance. For this explanation, the patch size is 5×5 and the size of the searching area in Fig. 5(a) and (b) is 31×31. In Fig. 5(a), $\sigma = 50$ may be a proper setup for Patch Two. However, the same parameter setup can cause obvious over-smoothing for Patch One. Fig. 5(b) and (c) show the weights computed for Patch One and Patch Two, respectively. For a better illustration, the weights are normalized by their maxima. Each small square stands for the a neighbor's weight. The brighter the square, the larger the weight and vice versa. It is obvious that for Patch Two, the weights decay rapidly with only a few weights being non-zero values. However, for Patch One, most of the weights are significant, making the NLM act like a mean filter. Thus, an efficient and adaptive decaying factor is necessary.

3.3. Predefined searching window

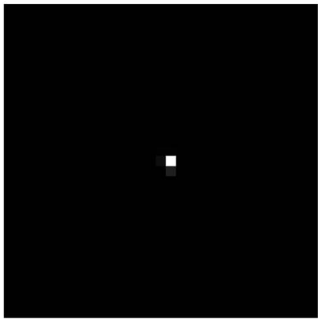
The size of the searching window $\mathcal{N}$ of NLM is a fixed parameter which is not suitable for the background and foreground objects in a frame undergoing different motions. The background is usually stable or moves smoothly. So a small searching window is enough for NLM to find similar pixels to a center pixel. However, the foreground objects may move fast and out of the range of the searching window. In this case, one has to enlarge the size of the searching window in order to recover the details of moving pixels. This treatment certainly increases the computational complexity. Thus, a fixed window size cannot meet both of the demands.

We propose two different methods to determine the size of the adaptive searching window, i.e., the predefined searching window and



(a)

(b)



(c)

Fig. 5. (a) Frame 9 of *Ballroom* sequence with two patches marked by squares and NLM weight distributions at (b) Patch One and (c) Patch Two.

Table 1
Determination of the exhaustive searching window.

Objective: Determine the window size for pixel (x, y).
Initialization: $E(x, y) = \xi + 1$, $\mathscr{P} = \mathscr{P}_{min}$
Pseudocode:
While $E(x, y) > \xi$ and $\mathscr{P} \leq \mathscr{P}_{max}$
Compute patch difference for newly added pixels.
Compute new local difference $E(x, y)$.
$\mathscr{P} = \mathscr{P} + \delta$.
end
Final window size: $\mathscr{P} = \mathscr{P} - \delta$.

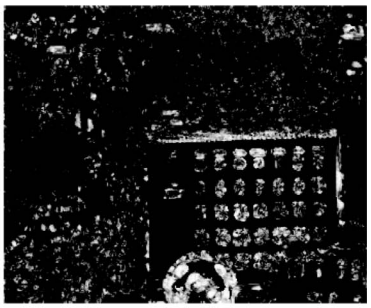
the exhaustive searching window. The predefined searching window is inspired by motion estimation. However, motion estimation involves complex computation. Thus, we propose to use varying searching window whose computation is relative simple although at the cost of accuracy. First of all, the absolute difference between two LF images F_b^L and $\widetilde{f}_n^L$ is calculated, i.e.,

$$\Delta_b = |F_b^L - \widetilde{f}_n^L|. \tag{17}$$

Then a map is established by comparing Δ_b with its mean

$$\mathscr{M}_b(x, y) = \begin{cases} 1, \Delta_b(x, y) > m_b \\ 0, otherwise \end{cases} \tag{18a}$$

(a)



(b)

Fig. 6. (a) Interpolation of the LR Frame 9 of *Mobile* sequence (b) reduction rate of local difference in Frame 9.

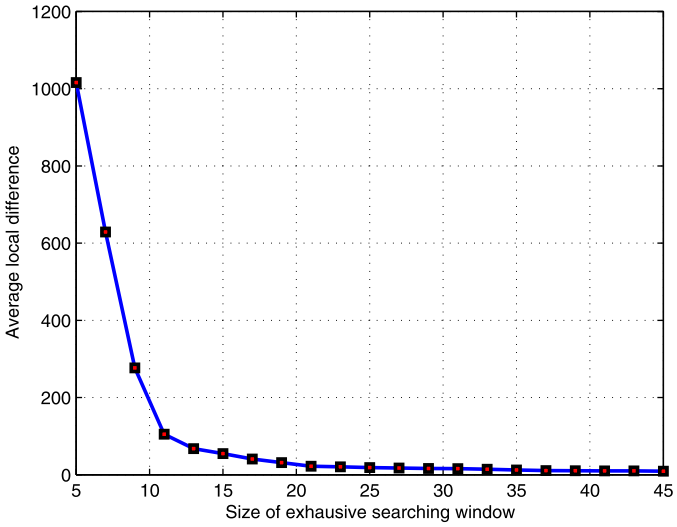


Fig. 7. Change of the average local difference with the increase of exhaustive searching window for selected pixels.

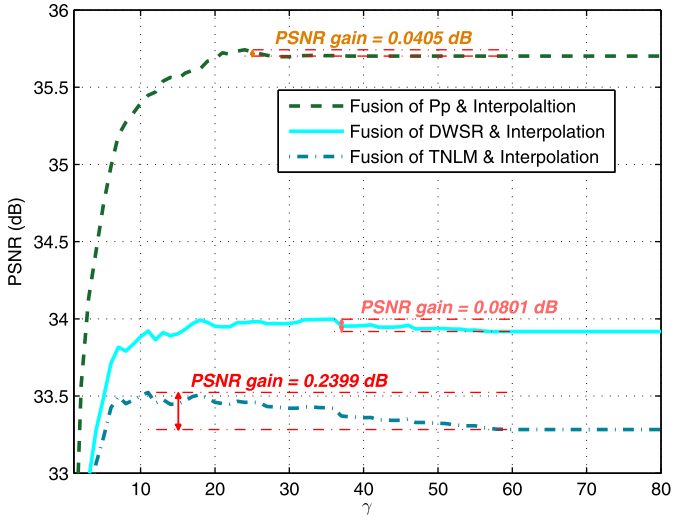


Fig. 8. Influence of the allowance for local difference.

Table 2
PSNR (dB) comparison between fused image and completely super-resolved image for Pp.

Period T	2	6	10	14	18	22	30
Pp	39.4612	38.1692	37.25	36.7717	36.3042	36.148	35.6285
Fusion of Pp & Inter-p.	39.4612	38.1706	37.25	36.7933	36.4187	36.3442	35.8514
PSNR gain	0	0.0014	0	0.0216	0.1144	0.1962	0.223

$$m_b = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\Delta_b(p, q) \tag{18b}$$

where M and N are the height and width of the image. One can already distinguish the background and the foreground using (17) and (18). In order to discriminate pixels with different motion, an indicator is calculated for every pixel, namely,

$$\mathscr{I}_b(x, y) = \sum_{(p,q)\in O_{xy}} \mathscr{M}_b(p, q) \tag{19}$$

where O_{xy} is a neighborhood of (x, y) of size $K \times K$, (p, q) is a pixel in O_{xy}. Then, a window size is assigned to every pixel with respect to the value of the indicator, namely,

$$\mathscr{S}_b(x, y) = s_l, \quad n_l \le \mathscr{I}_b(x, y) < n_{l+1} \tag{20}$$

where s_l's are the adaptive window sizes and n_l's are the thresholds between two consecutive levels with $n_0 = 0$ and $n_L = \max\{\mathscr{I}(x, y)\}$. The thresholds n_l's are uniformly distributed between $n_0 = 0$ and n_L. That is,

$$n_l = n_0 + l\cdot\Delta n \tag{21}$$

where $\Delta n = (n_L - n_0)/L$. In this paper, we use $L = 10$ levels of thresholds. Finally, the maximum window size among bilateral HR guide frames is calculated for every pixel

$$\mathscr{S}(x, y) = \max_{b\in\Phi}\{\mathscr{S}_b(x, y)\}. \tag{22}$$

The purpose of the above computation is the same as that of motion estimation, i.e., determining the size of the searching window for every pixel adaptively. But there are differences between them. The above computation avoids expensive motion estimation while its accuracy is worse than motion estimation. Thus, it is a tradeoff between computational complexity and accuracy.

3.4. Exhaustive searching window

The second method for determining the adaptive searching window is an exhaustive searching method with iterations. To determine the

Table 3
Global difference of 13 sequences using predefined window.

Frame	8	9	10	11	12
Ballroom	146	477	2928	2855	98
Foreman	27	41	38	52	34
Mobile	226	241	263	260	188
News	56	55	70	42	35
Hall	18	136	21	27	43
Flower	77	70	79	86	91
Container	45	79	124	70	40
Waterfall	27	39	29	36	24
Coastguard	69	35	65	94	45
Mother-Daughter	22	8	21	10	20
Crowd	123	290	894	1400	522
Exit	94	71	69	73	201
Vassar	40	39	140	60	45

Table 4
Global difference of 13 sequences using exhaustive window.

Frame	8	9	10	11	12
Ballroom	133	477	2928	2855	98
Foreman	27	41	38	52	34
Mobile	165	223	263	260	115
News	56	55	70	42	35
Hall	18	136	21	27	43
Flower	77	67	56	59	75
Container	45	79	124	70	40
Waterfall	27	39	29	36	24
Coastguard	69	35	65	82	45
Mother-Daughter	22	8	21	10	20
Crowd	122	290	894	875	363
Exit	94	71	69	73	201
Vassar	40	39	140	60	45

Table 5
Parameter setup.

Parameter	$\mathscr{N}$	S	K	L	α	σ	σ_s	$\mathscr{P}_{\min}$	$\mathscr{P}_{\max}$	ξ	δ
Value	9	5	10	10	2	0.2	1	5	45	1	4

Table 6
PSNR (dB) results of 13 test sequences.

Sequence	BI	TNLM	DWSR	FWp	FWe	FHL	WHLp	WHLe	Pp	Pe
Ballroom	31.09	35.03	36.5	35.21	35.19	37.23	37.54	37.48	**38.7**	38.63
Foreman	28.66	33.28	33.92	35.37	35.31	35	34.79	34.79	**35.7**	35.69
Mobile	20.09	22.26	22.45	24.33	24.51	24.12	22.75	22.96	24.49	**24.69**
News	25.57	37.24	38.07	39.1	39.08	39.49	38.21	38.24	**39.83**	39.81
Hall	25.42	36.72	38.4	37.54	37.08	39.37	38.16	37.8	**38.97**	38.59
Flower	21.11	23.17	23.45	26.87	26.77	25.04	25.65	25.59	**27.04**	26.94
Container	24.42	32.18	32.34	33.36	33.38	33.53	32.24	32.28	33.48	**33.51**
Waterfall	28.58	31.57	31.91	32.89	32.84	33.14	31.94	31.94	**33.16**	33.13
Coastguard	26.24	28.25	28.51	29.92	29.51	29.74	28.89	28.57	**30.08**	29.68
Mother-Daughter	32.41	42.12	43.52	42.34	42.28	**43.93**	43.24	43.39	43.42	43.71
Crowd	26.99	31.55	32.69	34.02	34.06	33.84	33.83	33.86	35.33	**35.35**
Exit	33.57	37.89	39.21	39.91	39.46	39.62	40.56	40.18	**41.08**	40.83
Vassar	32.73	36.37	37.48	37.56	37.35	38.33	37.72	37.55	**38.63**	38.5

Table 7
SSIM results of 13 test sequences.

Sequence	BI	TNLM	DWSR	FWp	FWe	FHL	WHLp	WHLe	Pp	Pe
Ballroom	0.95	0.971	0.984	0.976	0.975	0.986	0.987	0.986	**0.989**	0.988
Foreman	0.89	0.932	0.943	0.949	0.947	0.949	0.951	0.949	**0.955**	0.954
Mobile	0.654	0.84	0.846	0.877	0.879	0.876	0.853	0.857	0.882	**0.884**
News	0.87	0.978	0.982	0.986	0.986	0.988	0.982	0.982	**0.988**	0.988
Hall	0.857	0.943	0.957	0.952	0.946	0.963	0.959	0.953	**0.963**	0.96
Flower	0.732	0.886	0.893	0.936	0.933	0.911	0.927	0.925	**0.938**	0.936
Container	0.803	0.948	0.952	0.95	0.952	0.955	0.949	0.952	0.953	**0.955**
Waterfall	0.75	0.879	0.887	0.897	0.897	0.905	0.887	0.887	**0.903**	0.903
Coastguard	0.672	0.786	0.797	0.823	0.8	0.826	0.804	0.78	**0.83**	0.81
Mother-Daughter	0.911	0.976	0.981	0.976	0.976	**0.981**	0.98	0.981	0.98	0.981
Crowd	0.955	0.982	0.99	0.989	0.989	0.991	0.993	0.993	**0.994**	0.994
Exit	0.963	0.974	0.984	0.98	0.975	0.985	0.987	0.985	**0.988**	0.986
Vassar	0.946	0.968	0.979	0.972	0.97	0.982	0.981	0.979	**0.983**	0.982

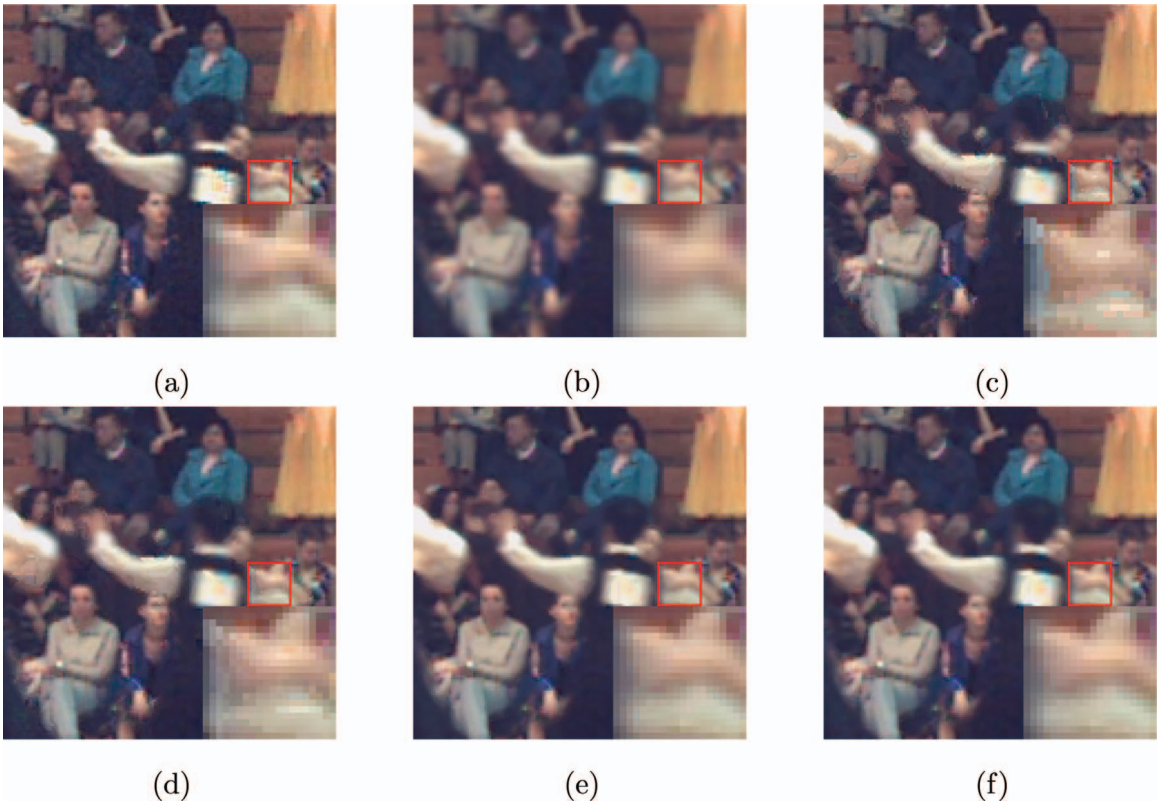


Fig. 9. Comparison of *Ballroom* (a) ground truth, (b) BI, (c) TNLM, (d) DWSR, (e) FWp, and (f) Pp.

Fig. 10. Comparison of *Foreman* (a) ground truth, (b) BI, (c) TNLM, (d) DWSR, (e) FWp, and (f) Pp.

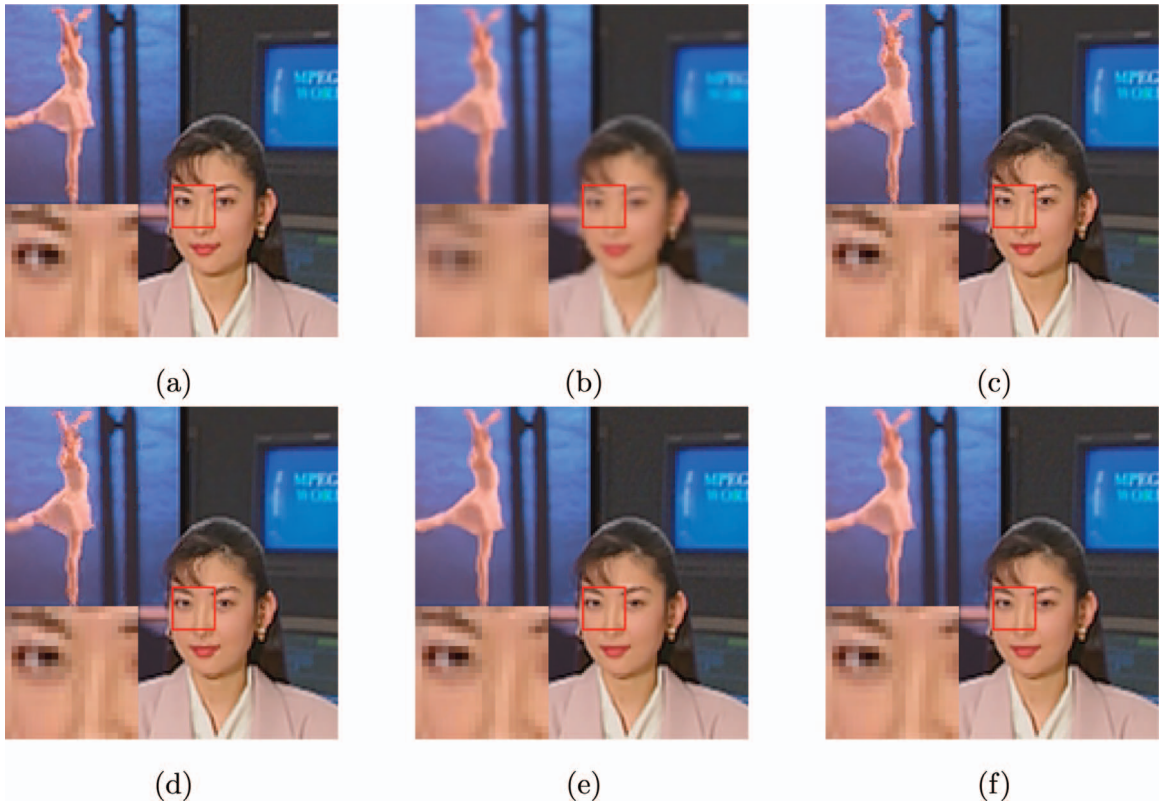


Fig. 11. Comparison of *News* (a) ground truth, (b) BI, (c) TNLM, (d) DWSR, (e) FWp, and (f) Pp.

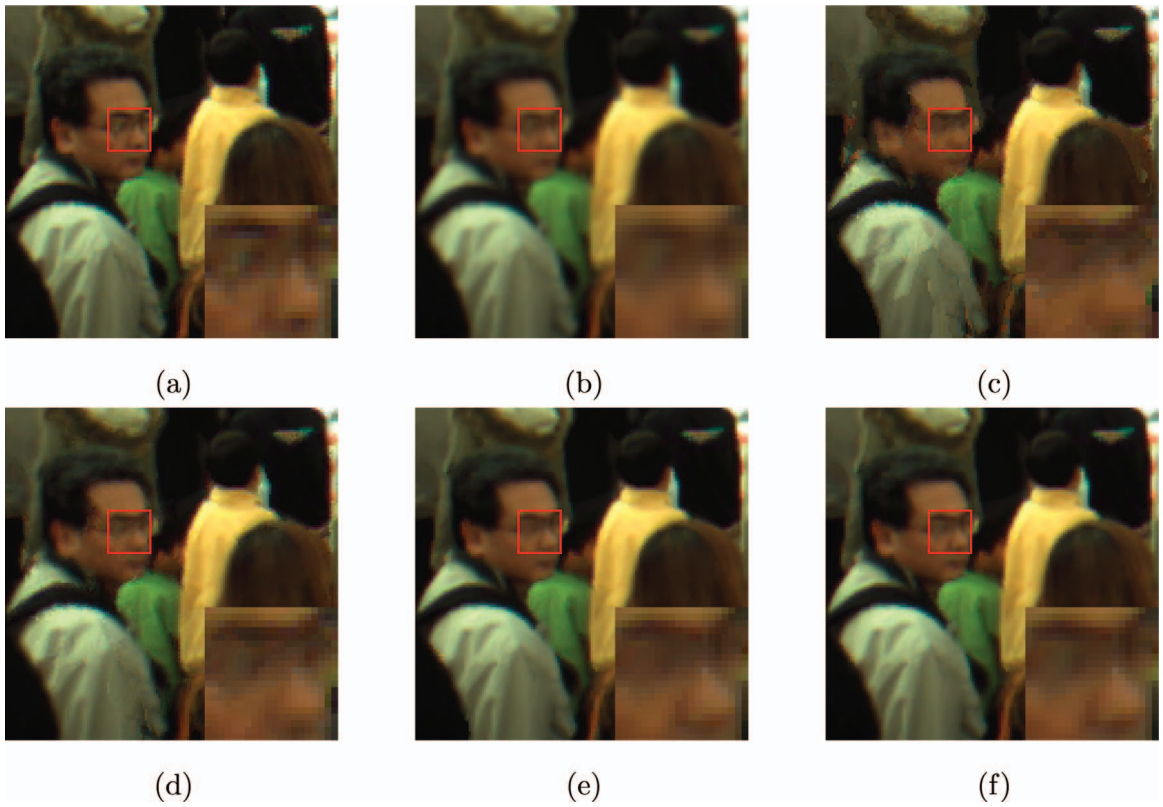


Fig. 12. Comparison of *Crowd* (a) ground truth, (b) BI, (c) TNLM, (d) DWSR, (e) FWp, and (f) Pp.

Table 8
PSNR(dB)/SSIM results of *Ballroom*.

	Frame 8	Frame 9	Frame 10	Frame 11	Frame 12
BI	31.06 /0.9484	31.08 /0.9491	31.09 /0.9498	31.23 /0.9506	31.29 /0.9516
TNLM	36.26 /0.9752	35.63 /0.9732	35.03 /0.9707	35.05 /0.9691	35.82 /0.9705
DWSR	37.81 /0.9862	37.10 /0.9854	36.50 /0.9841	36.63 /0.9834	37.60 /0.9844
Pp	**39.58 /0.9895**	**39.31 /0.9891**	**38.70 /0.9886**	**38.96 /0.9884**	**39.56 /0.9894**
Pe	39.50 /0.9891	39.24 /0.9887	38.63 /0.9882	38.87 /0.9880	39.47 /0.9888

Table 9
PSNR(dB)/SSIM results of *Waterfall*.

	Frame 8	Frame 9	Frame 10	Frame 11	Frame 12
BI	28.51 /0.7483	28.54 /0.7487	28.58 /0.7501	28.56 /0.7489	28.64 /0.7518
TNLM	34.05 /0.9346	32.79 /0.9105	31.57 /0.8793	31.75 /0.8831	33.33 /0.9207
DWSR	34.72 /0.9424	33.29 /0.9187	31.91 /0.8870	32.08 /0.8895	33.93 /0.9284
Pp	**35.70** /0.9493	**34.33** /0.9290	**33.16 /0.9035**	**33.19 /0.9040**	**34.87** /0.9370
Pe	35.67 **/0.9496**	34.30 **/0.9291**	33.13 /0.9030	33.16 /0.9037	34.85 **/0.9373**

size of a pixel's searching window, the algorithm begins with the default size, namely the minimum window size $\mathscr{P}_{\min}$. By comparing the local difference with the termination standard ξ, the algorithm judges whether to enlarge the window size and continue the iteration. If the local difference is larger than ξ, then enlarge the window size $\mathscr{P}$ by a step δ and repeat the above process. The iteration stops if the local difference is no larger than ξ or $\mathscr{P}$ exceeds a maximum window size $\mathscr{P}_{\max}$. At last, the window size in the last iteration is chosen as the desired one. Note that during the iteration, one need only to compute the patch difference of the new added pixels, which can save lots of computation. The iteration process is summarized in Table 1.

3.5. Allowance of local difference

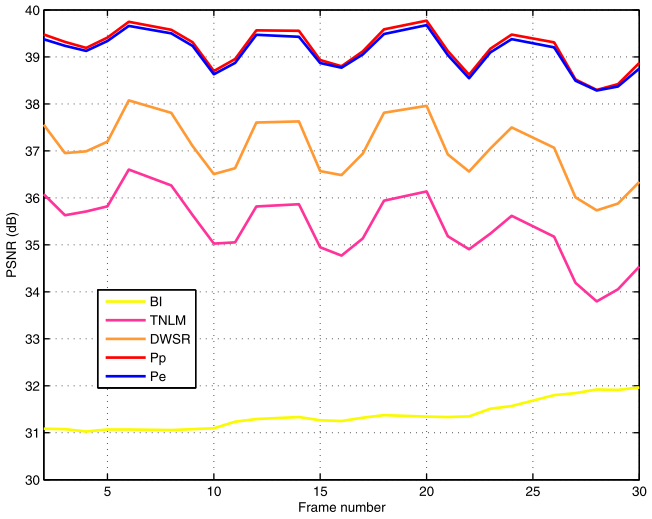
In the proposed NLM algorithm, local difference describes how similar a pixel is to its neighbors. A smaller local difference indicates that appropriate matches are found for a pixel within its neighborhood. This means that it is proper to recover the pixel using SR methods. However, the local differences of some peculiar pixel such as those undergoing fast motion remain large even if the adaptive searching window is employed. In this case, it might be improper to super-resolve these pixels since they are dissimilar to their neighbors. Instead, interpolation may be a more accurate solution. Thus, we set an allowance γ for the local difference according to which the SR and interpolated versions of an LR image are fused. The fusion operation is defined as

$$\widetilde{f}_n^F(x,y) = \begin{cases} \widetilde{f}_n(x,y), & E(x,y) \le \gamma \\ \widetilde{f}_n^L(x,y), & E(x,y) > \gamma \end{cases} \tag{23}$$

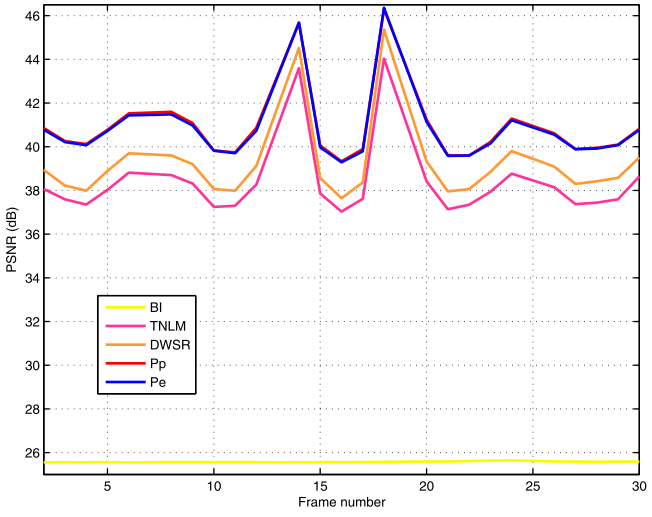
where $\widetilde{f}_n$, $\widetilde{f}_n^L$, and $\widetilde{f}_n^F$ are the interpolated, super-resolved, and finally fused versions of an LR image f_n. According to the equation, if a pixel's local difference is smaller than γ, it is recovered by SR method. Otherwise, it is replaced by an interpolated value. By altering the allowance γ, the SR image and interpolated image are fused to different degrees.

4. Analysis of patch difference

The patch difference has a huge influence on the performance of



(a)



(b)

Fig. 13. (a) PSNR (dB) results for *Ballroom* sequence (b) and *News* sequence.

NLM algorithm. Analysis of the patch difference and the concept defined by patch difference can lead to insightful understanding of the mechanics of the algorithm. Thus, in this section, we analyze patch difference in three levels, namely pixel-level, frame-level, and sequence-level. In this section, unless otherwise stated, the period T of the HR frames of a hybrid video is 6. That is, every seventh frame from the first one is HR frame. The magnification factor is 2. In the following of the analysis, Frame 7 and Frame 13 act as the HR guide frames while the aim is to super-resolve Frame 8 to Frame 12.

4.1. Pixel-level analysis

Pixel-level analysis tries to disclose how local difference changes with the increase of neighborhood size in the exhaustive searching window method. By enlarging the size of the searching window, the local difference will definitely reduce for most of the pixels in an image. The reduction rate of local difference for a pixel (x, y) is defined as

$$\rho(x, y) = \frac{E(x, y) - E_{exh}(x, y)}{E(x, y)} \tag{24}$$

where $E_{exh}(x, y)$ is the local difference of (x, y) after using the exhaustive searching window, $E(x, y)$ is the formerly defined local difference of (x, y) without using adaptive searching window.

Fig. 6 shows the reduction rate of local difference for Frame 9 of *Mobile* sequence. For the pixels where there is large motion or at the object edges, the reduction rate is close to 1. This observation demonstrates the effectiveness of the exhaustive searching window.

Fig. 7 shows the relationship between the average local difference and the size of the exhaustive searching window for selected pixels with the reduction rate $\rho > 0.99$ and the final window size $\mathscr{P} = \mathscr{P}_{\max} = 45$. It is obvious that the average local difference diminishes quickly in the initial iterations and tends to reduce to a fixed value at the final stage. In this case, it is no longer helpful to enlarge the searching window further.

4.2. Frame-level analysis

Frame-level analysis illustrates how the allowance γ of local difference described in Section 3.5 affects the objective performance of NLM, i.e., the peak signal-to-noise rate (PSNR). The analyzed methods include traditional NLM without adaptive parameters (TNLM) (Buades et al., 2005), detail warping based SR (DWSR) (Basavaraja et al., 2010), the proposed algorithm with adaptive decaying factor and predefined window (Pp). We set different γ ranging from 1 to the maximum possible local difference in an image, i.e., the global difference. Note that an allowance γ below 1 is nonsense because in this case most of the pixels are calculated by interpolation instead of SR method, which deteriorates the performance. The super-resolved image can be calculated in advance. After that, we fuse the super-resolved image and interpolated image according to (23). Thus, the analysis is entirely a post-processing and can be implemented efficiently. Fig. 8 shows the PSNR results of images fused by their interpolated and super-resolved versions for Frame 10 of *Foreman* sequence. When γ increases from 1, the PSNR curve of the fused image increases gradually, reaches its maxima at $\gamma = 11, 24, 36$ for TNLM, Pp, DWSR resp., and decays to a value where fusion is not used, i.e., the whole image recovered by SR. The optimal PSNR gain of fused images over completely super-resolved images are 0.2399 dB, 0.0801 dB, and 0.0405 dB for TNLM, DWST, and Pp respectively (Fig. 8). The PSNR gain of fused image is especially small for Pp, making us doubt the actual effect of the fusion operation. Thus, another example is given in Table 2 for Pp where Frame 30 of *Ballroom* sequence is recovered using different periods T of HR frames. Frame 30 is in the center of two HR

Table 10
PSNR (dB) and SSIM of *Foreman* for different periods T.

Period T		2	6	10	14	18	22	30
TNLM	PSNR	39.0717	34.8038	31.7836	29.3905	27.9885	28.0915	28.6042
	SSIM	0.9650	0.9409	0.9093	0.8639	0.8322	0.8305	0.8399
DWSR	PSNR	39.7055	35.4031	32.5575	30.1974	28.7774	28.9742	29.5381
	SSIM	0.9687	0.9489	0.9259	0.8917	0.8679	0.8684	0.8756
Pp	PSNR	39.9956	37.1453	35.0733	34.293	33.6475	33.5894	33.4994
	SSIM	0.9707	0.9589	0.9473	0.9357	0.931	0.9276	0.9284
Pe	PSNR	40.0139	37.0155	34.8804	34.1517	33.7646	33.633	33.36
	SSIM	0.9711	0.9571	0.9437	0.9303	0.9272	0.9246	0.9244

Table 11
PSNR (dB) and SSIM of *News* for different periods *T*.

Period *T*		2	6	10	14	18	22	30
TNLM	PSNR	39.0975	37.2185	35.8448	34.7526	34.7526	34.1917	34.4666
	SSIM	0.9824	0.9783	0.974	0.9697	0.9696	0.9667	0.9665
DWSR	PSNR	39.7928	38.0858	36.6127	35.5383	35.3617	34.8917	35.1739
	SSIM	0.9852	0.9821	0.9785	0.975	0.9745	0.9724	0.9719
Pp	PSNR	41.1628	39.7089	38.8282	37.9538	37.9198	37.4634	37.3906
	SSIM	0.9905	0.9881	0.9863	0.9843	0.9842	0.9833	0.9825
Pe	PSNR	41.1774	39.6843	38.7906	37.918	37.8954	37.4019	37.3096
	SSIM	0.9906	0.9881	0.9862	0.9842	0.9842	0.9831	0.9823

Table 12
PSNR (dB) and SSIM of *Flower* for magnification factors.

Magnification Factor		2	4	8	16	32
TNLM	PSNR	37.2438	31.6217	25.519	23.7388	21.8957
	SSIM	0.9782	0.9418	0.8752	0.8176	0.7504
DWSR	PSNR	38.0658	31.5028	25.4682	23.7309	21.8866
	SSIM	0.9818	0.9407	0.8744	0.8173	0.75
Pp	PSNR	39.829	33.1009	26.9612	23.8568	21.0897
	SSIM	0.988	0.9567	0.8963	0.8413	0.7583
Pe	PSNR	39.8145	33.1084	27.1596	24.5027	22.4404
	SSIM	0.988	0.9569	0.9011	0.8614	0.8041

Table 13
PSNR (dB) and SSIM of *Container* for magnification factors.

Magnification Factor		2	4	8	16	32
TNLM	PSNR	23.1746	18.3968	15.7006	15.5546	15.5616
	SSIM	0.8862	0.6831	0.4975	0.4885	0.4883
DWSR	PSNR	23.4484	18.3915	15.6566	15.4949	15.504
	SSIM	0.8931	0.6827	0.4958	0.4876	0.4873
Pp	PSNR	27.0359	20.3046	17.2625	16.64	16.262
	SSIM	0.9384	0.7157	0.5234	0.496	0.4909
Pe	PSNR	26.9398	20.2553	17.2408	16.6063	16.2245
	SSIM	0.9362	0.7106	0.5203	0.4972	0.4891

guide frames. In Table 2, the PSNR gain increases and the advantage of fusion operation stands out with the increase of *T*. This is because inaccurate matches are more likely to appear for a larger *T*. As said, the aim of the fusion operation is using interpolated pixels to alleviate the drawback of SR methods where it is inaccurate to refine foreground dancers. Since *Ballroom* sequence has a stable background that accounts for most of the regions of the image, the 0.233 dB PSNR gain for period *T* is quite reasonable. Thus, we conclude that the fusion operation has a profound impact only in the case where SR condition deteriorates (e.g. a large *T*).

4.3. Sequence-level analysis

The sequence-level analysis focuses on the global difference defined by (12) which reveals some important features (e.g. motion feature) of a video. Analyzing the global difference is helpful for the implementing of the algorithm. Table 3 shows the global difference of five consecutive frames for 13 different video sequences using the predefined searching window. Table 4 shows the corresponding data for exhaustive searching window. There are three noticeable features of the global difference. First of all, the frames in the middle (Frame 9, Frame 10, Frame 11) usually have the largest global difference. The pattern is in consistence with the results in Table 8 for *Ballroom* sequence where Frame 10 has the worst results. This is because the interval between the HR guide frames and the middle LR frames are larger compared with other frames. Secondly, the global difference diverges for different sequences. For example, the global differences of *Mother-Daughter* sequence are very small for all of the five consecutive LR frames, which indicates that the sequence undergoes very slow motion. In this scenario, the allowance of local difference is no longer necessary since the global difference is so small. Thirdly, the global difference using predefined searching window is not smaller than that using exhaustive searching window.

5. Experimental results

In this section, we show the experimental results of the proposed and compared methods. All of these methods have been tested on 13 video sequences with different characteristics including *Ballroom, Foreman, Mobile, News, Hall, Flower, Container, Waterfall, Coastguard, Mother-Daughter, Crowd, Exit*, and *Vassar*. The compared methods include bilinear interpolation (BI), TNLM (Buades et al., 2005), DWSR (Basavaraja et al., 2010), DWSR with adaptive decaying factor (FHL), DWSR with predefined searching window (WHLp), DWSR with exhaustive searching window (WHLe), TNLM with adaptive decaying factor and predefined searching window (FWp), TNLM with adaptive decaying factor and exhaustive searching window (FWe), the proposed method with adaptive decaying factor and predefined searching window (Pp), the proposed method with adaptive decaying factor and exhaustive searching window (Pe). Bilinear interpolation acts as the baseline for comparison. The parameter setup is listed in Table 5. Using (21) and the parameters *K, L*, the thresholds n_l's in (20) are 0, 10, 20, ...,100, respectively. The corresponding window sizes s_l's of each level are 5, 9, 13, ...,45 and they are chosen empirically. Usually, the window size should not be smaller than the patch size *S* (5 in this paper). Thus, the minimum possible window size is set to 5. On the other hand, as in Fig. 7, the marginal benefit of increasing window size reduces. The average local difference tends to approach a constant when the window size is around 45. Thus, we empirically set the maximum possible window size to 45. Since 10 levels are used in (20), the step of window size is 4.

Since there was no available hybrid videos, we simulated hybrid videos from the original full-resolution video sequence as follows. The LR frames were acquired from the original full resolution frames after blurring and down-sampling. Lanczos filter acted as a point spread function (PSF) to simulate the blurring operation *B* during image acquisition process (Duchon, 1979). The down-sampling/magnification factor was 2. The full resolution frames corresponding to the HR frames remained unchanged. PSNR and Structural SIMilarity (SSIM) (Wang et al., 2004) were used to compare and evaluate different methods.

We implemented different experiments to test the proposed algorithm. In the first experiment, the period *T* of the HR guide frames was 6. That is, Frame 1, 7, 13, ··· were HR frames while the others were LR frames. The PSNR and SSIM results of Frame 10 for all of the 13 sequences are shown in Tables 6 and 7, respectively. By comparing the results of FHL, WHLp, WHLe with those of DWSR, one can find that each of the proposed adaptive parameters can improve the performance of DWSR. The separation of HF and LF components is essential for improving the performance of TNLM. But by only combining the

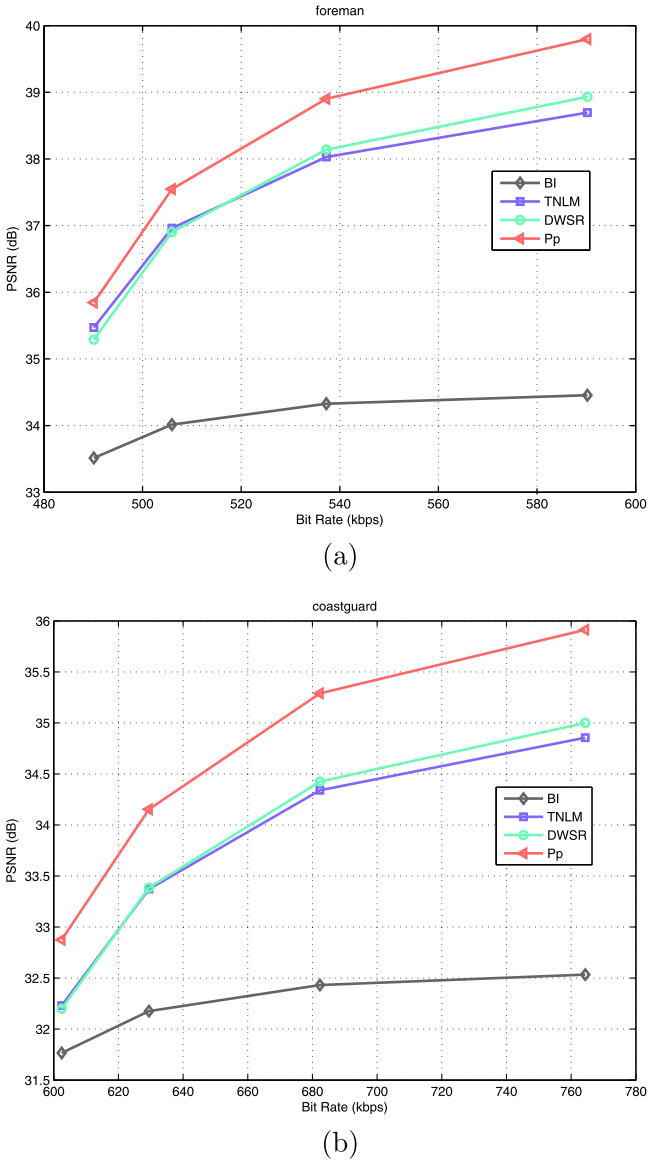


Fig. 14. (a) Rate-distortion comparison between different method for *Foreman* sequence (b) and *Coastguard* sequence.

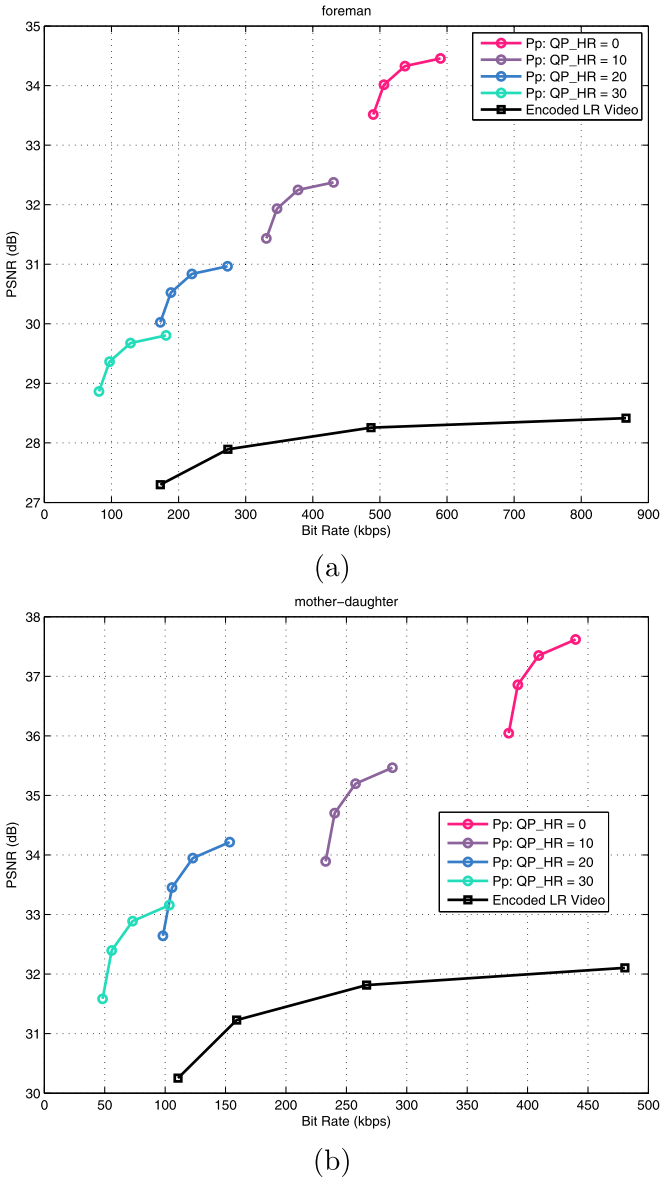


Fig. 15. (a) Rate-distortion comparison between hybrid video coding scheme and LR video coding scheme for *Foreman* sequence (b) and *Mother-Daughter* sequence.

adaptive decaying factor and searching window with TNLM, the algorithms FWp and FWe outperforms DWSR for *Foreman, Mobile, News, Flower, Container, Waterfall, Coastguard, Crowd*, and *Exit* even without the separation of HF and LF components. The proposed algorithms Pp and Pe achieves the best results among all of the method. It is interesting that the algorithm with predefined searching window (FWp v.s. FWe, WHLp, v.s. WHLe, Pp v.s. Pe) works out better than that with exhaustive searching window for most of the sequences while exhaustive searching window stands out only for *Mobile, Container*, and *Crowd*. The reason is that, in order to reach the termination standard Φ, the exhaustive searching continues unless the maximal window size gets hit. Thus, compared with predefined searching window, exhaustive searching window tends to be larger, causing the involvement of more pixels in the averaging process of NLM. These excessive pixels of exhaustive searching are unnecessary and sometimes undesired because the over-averaging or over-smoothing may cause the loss of HF details to a certain degree. Note that for *Mother-Daughter*, the algorithm without adaptive searching window, namely FHL outperforms all of the other algorithms. Because, from the sequence-level analysis, *Mother-Daughter* has the smallest global difference and the motion of the sequence is slow. The refined images by different methods are shown in Figs. 9–12.

In the second experiment, we tested the algorithm for the whole sequence. Tables 8 and 9 show the results for 5 consecutive frames of *Ballroom* and *Waterfall*. For both of the sequences, Frame 10 is with the lowest objective quality. The PSNR results of *Ballroom* and *Waterfall* for a video segment containing 30 frames is shown in Fig. 13. The PSNR curve shows periodicity for the video segments as expected.

In the third and fourth experiments, we changed the period T of HR guide frames and the magnification factor. In the third experiment, the SR algorithms recover Frame 30 which is an LR frame centered between two HR guide frames, with the aid of the two HR frames. The results of *Foreman* and *News* sequences for different periods T are shown Tables 10 and 11. With the decrease of T, the results of all algorithms improve steadily. The data size of the hybrid video before encoding with period T only accounts for $(T+3)/4T$ of that of the original videos. Tables 12 and 13 show the SR results of *Flower* and *Container* sequences in the fourth experiment where LR Frame 10 is super-resolved with different magnification factors using HF information from Frame 7 and Frame 13. As expected, the PSNR and SSIM values of SR algorithms drop with the enlargement of magnification factor but the

proposed algorithm still achieves the best performance.

In the final experiment, we analyzed the rate-distortion (RD) performance of the proposed algorithm. Since there were two different resolutions of hybrid video frames, we encoded HR frames and LR frames separately using H.264 inter coding scheme. There were two parameters that could be tuned in the encoding process, i.e., the quantization parameter (QP) of HR frames and the QP of LR frames. The set of QPs of LR frames was {22, 27, 32, 37}. We first compared the RD curves of different SR methods. The QP of HR frame was 0. Fig. 14 shows the comparison results for *Foreman* and *Coastguard* sequences. Compared with bilinear interpolation, all of the three SR methods achieves larger PSNR values for various bit rates, with TNLM and DWSR comparable with each other and Pp surpassing their performances. Then the coding schemes of hybrid video and totally LR video were compared. All of the LR frame of the two coding scheme were still encoded using H.264 coding with QP set {22, 27, 32, 37}. The QP set of HR frames of hybrid video is {0, 10, 20, 30}. Fig. 15 shows the results of *Foreman* and *Mother-Daughter* sequences. Since the LR video is low-pass filtered and compressed, the PSNR result is upper bounded. By configuring the QPs of HR and LR frames of hybrid video, the SR algorithm trades off between required bit rates and achieved PSNR.

6. Conclusion

The NLM algorithm has a very promising application in SR tasks. However, the traditional NLM algorithms suffer from two main drawbacks, i.e., the fixed decaying factor and searching window. The fixed decaying parameter is unfit for regions with different characteristics. It tends to blur the relatively flat regions in the image, resulting in perceived loss of detail. On the other hand, the fixed searching window leads to mismatches between pixels, causing unbearable degradation of the video. In this paper, we proposed to use adaptive decaying factor, predefined searching window and exhaustive searching window to improve the performance of NLM. For most of the tested sequences, predefined searching window achieved better results. We defined local difference and global difference based on patch difference and analyzed their influences on NLM. The influence of allowance γ of local difference was analyzed for different algorithms. The experimental results validated the effectiveness and robustness of the proposed algorithm.

Acknowledgment

This work was supported by the Natural Science Foundation of China (61671126).

References

Allebach, J., Wong, P.W., 1996. Edge-directed interpolation. Proc. IEEE International Conference on Image Processing. 3. pp. 707–710.
Ancuti, C., Ancuti, C.O., Bekaert, P., 2010. Video super-resolution using high quality photographs. Proc. IEEE International Conference on Acoustic, Speech, and Signal Processing. pp. 862–865.
Basavaraja, S.V., Bopardikar, A.S., Velusamy, S., 2010. Detail warping based video super-resolution using image guides. Proc. IEEE International Conference on Image Processing. pp. 2009–2012.
Ben-Ezra, M., Nayar, S.K., 2003. Motion deblurring using hybrid imaging. Proc. IEEE Conference on Computer Vision and Pattern Recognition. pp. 657–664.
Bevilacqua, M., Roumy, A., Guillemot, C., Morel, M.-L.A., 2013. Video super-resolution via sparse combinations of key-frame patches in a compression context. Proc. Picture Coding Symposium. pp. 337–340.
Bose, N.K., Ahuja, N.A., 2006. Superresolution and noise filtering using moving least squares. IEEE Trans. Image Process. 15 (8), 2239–2248.
Brandi, F., de Queiroz, R., Mukherjee, D., 2008. Super-resolution of video using key frames and motion estimation. Proc. IEEE International Conference on Image Processing. pp. 321–324.
Buades, A., Coll, B., Morel, J.M., 2005. A non-local algorithm for image denoising. Proc. IEEE Conference on Computer Vision and Pattern Recognition. 2. pp. 60–65.
Chang, H., Yeung, D.-Y., Xiong, Y., 2004. Super-resolution through neighbor embedding. Proc. IEEE Conference on Computer Vision and Pattern Recognition. pp. 275–282.
Choi, B.-T., Lee, S.-H., Ko, S.-J., 2000. New frame rate up-conversion using bi-directional motion estimation. IEEE Trans. Consum. Electron. 46 (3), 603–609.
Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K., 2007. Image denoising by sparse 3-D transform-domain collaborative filtering. IEEE Trans. on Image Process. 16 (8), 2080–2095.
Dai, Q., Yoo, S., Kappeler, A., Katsaggelos, A.K., 2017. Sparse representation-based multiple frame video super-resolution. IEEE Trans. Image Process. 26 (2), 765–781.
Dong, W., Zhang, L., Shi, G., Li, X., 2013. Nonlocally centralized sparse representation for image restoration. IEEE Trans. Image Process. 22 (4), 1620–1630.
Duchon, C.E., 1979. Lanczos filtering in one and two dimensions. J. Appl. Meteorol. 18, 1016–1022.
Farsiu, S., Robinson, M.D., Elad, M., Milanfar, P., 2004. Fast and robust multiframe super resolution. IEEE Trans. Image Process. 13 (10), 1327–1344.
Freeman, W.T., Jones, T.R., Pasztor, E.C., 2001. Example-based super-resolution. IEEE Comput. Graph. Appl. 22 (2), 56–65.
Fu, Z., Li, Y., Li, Y., Ding, L., Long, K., 2016. Frequency domain based super-resolution method for mixed-resolution multiview images. J. Syst. Eng. Electron. 27 (6), 1303–1314.
Fu, Z., Wang, X., Li, Y., Yang, X., Xu, J., 2016. Adaptive luminance adjustment and neighborhood spreading strength information based view synthesis. J. Syst. Eng. Electron. 27 (3), 721–729.
Gan, Z., Cui, Z., Chen, C., Zhu, X., 2013. Adaptive joint nonlocal means denoising back projection for image super resolution. Proc. IEEE International Conference on Image Processing. pp. 630–634.
Garcia, D.C., Dorea, C., de Queiroz, R.L., 2012. Super resolution for multiview images using depth information. IEEE Trans. Circuits Syst. Video Technol. 22 (9), 1249–1256.
Gilboa, G., Osher, S., 2008. Nonlocal operators with applications to image processing. Multiscale Model. Simul. 7 (3), 1005–1028.
Huang, Y., Wang, W., Wang, L., 2017. Bidirectional recurrent convolutional networks for multi-frame super-resolution. IEEE Trans. Pattern Anal. Mach. Intell. PP (99). 1–1
Hung, E.M., de Queiroz, R.L., Brandi, F., de Oliveira, K.F., Mukherjee, D., 2012. Video super-resolution using codebooks derived from key-frames. IEEE Trans. Circuits Syst. Video Technol. 22 (9), 1321–1331.
Jin, Z., Tillo, T., Yao, C., Xiao, J., 2015. Virtual view assisted video super-resolution and enhancement. IEEE Trans. Circuits Syst. Video Technol. 26 (3), 467–478.
Kappeler, A., Yoo, S., Dai, Q., Katsaggelos, A.K., 2016. Super-resolution of compressed videos using convolutional neural networks. Proc. IEEE International Conference on Image Processing. pp. 1150–1154.
Kim, J., Kwon Lee, J., Mu Lee, K., 2016. Accurate image super-resolution using very deep convolutional networks. Proc. IEEE Conference on Computer Vision and Pattern Recognition. pp. 1646–1654.
Lengyel, R., Soroushmehr, S.R., Shirani, S., 2014. Multi-view video super-resolution for hybrid cameras using modified NLM and adaptive thresholding. Proc. IEEE International Conference on Image Processing. pp. 5437–5441.
Li, X., Orchard, M.T., 2001. New edge-directed interpolation. IEEE Trans. Image Process. 10 (10), 1521–1527.
Li, Y., Li, X., Fu, Z., Niu, T., Long, K., 2016. Spatiotemporal super-resolution for multiview video in transform domain. Proc. IEEE International Conference on Visual Communications and Image Processing. pp. 1–4.
Li, Y., Li, X., Fu, Z., Yin, X., Zhao, Y., 2016. Bilateral video super-resolution using non-local means with adaptive parameters. Proc. IEEE International Conference on Image Processing. pp. 1155–1159.
Li, Y., Li, X., Fu, Z., Zhong, W., 2016. Multiview video super-resolution via information extraction and merging. Proc. ACM International Conference on Multimedia. pp. 446–450.
Li, Y., Li, X., Yao, C., Fu, Z., Yin, X., 2016. Video super-resolution using non-local means with adaptive decaying factor and searching window. Proc. Asian Conference on Computer Vision Workshops.
Liao, R., Tao, X., Li, R., Ma, Z., Jia, J., 2015. Video super-resolution via deep draft-ensemble learning. pp. 531–539.
Liu, C., Sun, D., 2014. On Bayesian adaptive video super resolution. IEEE Trans. Pattern Anal. Mach. Intell. 36 (2), 346–360.
Mahfoodh, A.-T., Mukherjee, D., Radha, H., 2015. Super-resolution for inconsistent scalable video streaming. Proc. IEEE International Conference on Image Processing. pp. 3019–3023.
Mukherjee, D., Bankoski, J., Grange, A., Han, J., Koleszar, J., Wilkins, P., Xu, Y., Bultje, R., 2013. The latest open-source video codec VP9-an overview and preliminary results. Proc. Picture Coding Symposium. pp. 390–393.
Mukherjee, D., Macchiavello, B., de Queiroz, R.L., 2007. A simple reversed-complexity Wyner-ziv video coding mode based on a spatial reduction framework. SPIE Visual Communications and Image Processing. pp. 1–12.
Polatkan, G., Zhou, M., Carin, L., Blei, D., Daubechies, I., 2015. A Bayesian nonparametric approach to image super-resolution. IEEE Trans. Pattern Anal. Mach. Intell. 37 (2), 346–358.
Protter, M., Elad, M., Takeda, T., Milanfar, P., 2009. Generalizing the nonlocal-means to super-resolution reconstruction. IEEE Trans. Image Process. 18 (1), 36–51.
Ren, C., He, X., Nguyen, T.Q., 2017. Single image super-resolution via adaptive high-dimensional non-local total variation and adaptive geometric feature. IEEE Trans. on Image Process. 26 (1), 90–106.
Richter, T., Seiler, J., Schnurrer, W., Kaup, A., 2015. Robust super-resolution for mixed-resolution multiview image plus depth data. IEEE Trans. Circuits Syst. Video Technol. 26 (5), 814–828.
Shi, W., Caballero, J., Huszár, F., Totz, J., Aitken, A.P., Bishop, R., Rueckert, D., Wang, Z., 2016. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. Proc. IEEE Conference on Computer Vision and Pattern Recognition. pp. 1874–1883.
Song, B.C., Jeong, S.-C., Choi, Y., 2011. Video super-resolution algorithm using bi-

directional overlapped block motion compensation and on-the-fly dictionary training. IEEE Trans. Circuits Syst. Video Technol. 21 (3), 274–285.
Tai, Y.-W., Du, H., Brown, M.S., Lin, S., 2010. Correction of spatially varying image and video motion blur using a hybrid camera. IEEE Trans. Pattern Anal. Mach. Intell. 32 (6), 1012–1028.
Takeda, H., Farsiu, S., Milanfar, P., 2007. Kernel regression for image processing and reconstruction. IEEE Trans. Image Process. 16 (2), 349–366.
Timofte, R., Rothe, R., Van Gool, L., 2016. Seven ways to improve example-based single image super resolution. Proc. IEEE Conference on Computer Vision and Pattern Recognition. pp. 1865–1873.
Wang, D., Zhang, L., Vincent, A., 2010. Motion-compensated frame rate up-conversion – part i: fast multi-frame motion estimation. IEEE Trans. Broadcast. 56 (2), 133–141.
Wang, H., Gao, X., Zhang, K., Li, J., 2016. Single-image super-resolution using active-sampling gaussian process regression. IEEE Trans. Image Process. 25 (2), 935–948.
Wang, Q., Ward, R.K., 2007. A new orientation-adaptive interpolation method. IEEE Trans. Image Process. 16 (4), 889–900.
Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P., 2004. Image quality assessment: from error visibility to structural similarity. IEEE Trans. Image Process. 13 (4), 600–612.
Werlberger, M., Pock, T., Bischof, H., 2010. Motion estimation with non-local total variation regularization. Proc. IEEE Conference on Computer Vision and Pattern Recognition. pp. 2464–2471.
Yin, M., Gao, J., Cai, S., 2015. Image super-resolution via 2D tensor regression learning. Comput. Vis. Image Underst. 132, 12–23.
Zhang, K., Gao, X., Tao, D., Li, X., 2012. Single image super-resolution with non-local means and steering kernel regression. IEEE Trans. Image Process. 21 (11), 4544–4556.
Zhang, L., Wu, X., 2006. An edge-guided image interpolation algorithm via directional filtering and data fusion. IEEE Trans. Image Process. 15 (8), 2226–2238.
Zhang, X., Burger, M., Bresson, X., Osher, S., 2010. Bregmanized nonlocal regularization for deconvolution and sparse reconstruction. SIAM J. Imaging Sci. 3 (3), 253–276.
Zhang, X., Liu, Q., Li, X., Zhou, Y., Zhang, C., 2016. Non-local feature back-projection for image super-resolution. IET Image Process. 10 (5), 398–408.
Zhao, Y., Zhu, C., Chen, Z., Tian, D., Yu, L., 2011. Boundary artifact reduction in view synthesis of 3d video: from perspective of texture-depth alignment. IEEE Trans. Broadcast. 57 (2), 510–522.