

# Enhanced Multi-Frame Based Super-Resolution Algorithm by Normalizing the Information of Registration

Soon-Chan Kwon\* and Jisang Yoo<sup>†</sup>

**Abstract** – In this paper, a new super-resolution algorithm is proposed by using successive frames for generating high-resolution frames with better quality than those generated by other conventional interpolation methods. Generally, each frame used for super-resolution must only have global translation and motions of sub-pixel unit to generate good result. However, the newly proposed MSR algorithm in this paper is exempt from such constraints. The proposed algorithm consists of three main processes; motion estimation for image registration, normalization of motion vectors, and pattern analysis of edges. The experimental results show that the proposed algorithm has better performance than other conventional algorithms.

**Keywords:** Super-resolution, Interpolation, Motion estimation, 6-Tap filter, Image restoration

## 1. Introduction

Recently, there has been a growing need for interpolation of video content because of the numerous digital imaging devices that still produce low-resolution images. Image interpolation involves inserting appropriate value between the original pixels to enlarge the size of an image. Already existing image interpolation methods such as bi-linear and bi-cubic methods are based on the weighted-sum of the pixels of a low-resolution image [1]. These methods act similar to low-pass filters, thus inevitable artifact such as blurring of the edges of some objects may occur. It is therefore an ill-posed problem because we have to create a completely new value with only insufficient information.

To solve this problem, various super-resolution algorithms have been proposed. Super-resolution algorithms can be categorized as single-frame based super-resolution (SSR) and multi-frame based super-resolution (MSR) [2].

There are two general SSR algorithms. One involves the use of sub-bands that result from applying discrete wavelet transform (DWT) to the input image [3]. It skips the down-sampling process in the general DWT to the input image. By using this property, LL-LH-HL-HH sub-bands have the same size as the input image. Another SSR algorithm works by generating training set of numbers of high-resolution patches and then reconstructing the high-resolution image by using the training set [4]. This process is called the exemplar-based super-resolution algorithm.

However, there are some problems with the existing SSR algorithms. The first algorithm based on DWT usually performs better than other interpolation methods but is still

bound by the lack of information available. In the second SSR algorithm mentioned above, we need many resources of memory for producing the training set and making of the training set is time-consuming. Also, in the case that the target resolution of the input image exceeds the resolution of the high-resolution patch in training set, the resolution that can be produced is limited. To solve these problems, various multi-frame based super-resolution (MSR) algorithms have been proposed. Most of them assume that successive frames in a scene are generated from one desired high-resolution image with some degrading processes. Finding these processes reversely is defined as a MSR algorithm [2].

Fig. 1 shows the steps of generating the low-resolution degraded image from the original image in general. A low-resolution image is usually degraded in the analog to digital converting (ADC) process where down-sampling and image sensing errors exist.

Most of the previously proposed MSR algorithms assumed that low-resolution images should have only global translation and small sub-pixel unit motion for good results [5]. However, these constraints are not satisfied with most of video contents. To contend with this, an algorithm using a generalized non-local mean filter has been proposed [6], but its heavy computational power resulted in a blurred artifact nevertheless. Thus, an algorithm using a key-frame containing high-resolution information has been proposed, but this algorithm cannot be applied when there is no key-frame available in a given video content [7]. Also, an MSR algorithm using 6-tap FIR

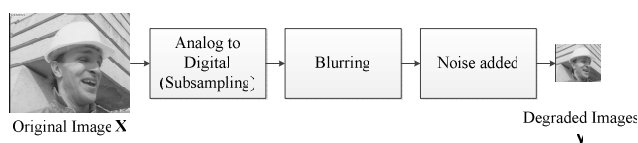


Fig. 1. Generating the low-resolution image

<sup>†</sup> Corresponding Author: Dept. of Electronic Engineering, Kwang-woon University, Korea. (jsyoo@kw.ac.kr)

\* Dept. of Electronic Engineering, Kwang-woon University, Korea. (hoenst\_soenc@kw.ac.kr)

Received: June 2, 2013; Accepted: October 7, 2013

(finite impulse response) filter and motion estimation has been proposed, but the algorithm does not consider boundary information of the frames while interpolating so that some of the results come out distorted [8].

In this paper, we proposed a new multi-frame based super-resolution algorithm to solve the problems mentioned above. In the proposed algorithm, we can get more usable information for interpolation by normalizing the motion vector obtained in the matching process. We can also make the algorithm more robust by analyzing 2\*2 block edge patterns for those matching points.

This paper is organized as follows. In section 2, we give a detailed description of the conventional MSR algorithms and the newly proposed algorithm. In section 3, we compare the performance of the proposed algorithm with that of the conventional ones through experiments and discuss the results. We give concluding remarks in section 4.

## 2. Proposed Algorithm

In this section, we will discuss the existing algorithms in detail. First, in 2.1, we will discuss how the existing algorithms work along with their limits. Then from 2.2 to 2.5, we will talk about the specific algorithms and explain how they work. Especially, the newly proposed algorithm in this paper is described in 2.3 and 2.4.

### 2.1 Conventional multi-frame based super-resolution algorithm

Fig. 2 shows a block diagram of a general multi-frame based super-resolution (MSR) algorithm. Several frames including a target frame are used to create an image or video for super-resolution. In this figure, input frames  $y_1 \sim y_t$  are successive, and there is no scene change. That is, the successive frames are correlated with each other. Most of MSR algorithms use the fact that in a video content, there usually exists a correlation between the successive frames because 20~30 images are usually acquired in one second. High-resolution images can be generated by blending the information from the successive frames [2].

As shown in Fig. 2, MSR algorithms are usually composed of three steps. In the first step, we register the input low-resolution images to target image. As mentioned in section I, we assume that the successive frames are generated from one desired high-resolution image. Thus, we can acquire the relationship of the pixel-position between each frame in the registration process. The registration process is very important because it controls



Fig. 2. Block diagram of multi-frame based super-resolution

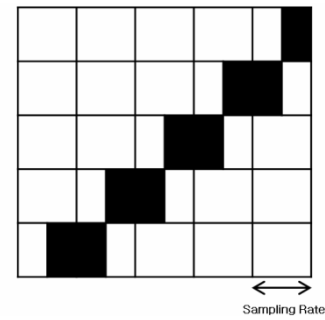


Fig. 3. Aliasing effect in down-sampling process

the performance of entire algorithm.

The importance and the principle of registration are like this. As shown in Fig. 3, the stair-shaped black squares represent the boundary of an object (edge) in the image. Originally, the high-frequency component of an analog image signal gets damaged from going through the sampling process of the digital device because of the sampling interval [11]. When increasing the resolution of an image through this process, the necessary information can be received. However, there is one more required condition in order to receive the information. The left column in Fig. 4 represents the image from  $f(t-1)$ , the right column shows the image shown from  $f(t)$ . In a video that normally shows 20~30 images per second, the images from adjacent time intervals when  $f(t-1)$  is included show very little difference. There are two ways that the edges in the images from adjacent time intervals can move. One way is as shown in Fig. 4(a) where the edge moves along the sampling interval (integer-pixel).

In this case, the pixel information from the edges in the images from the adjacent time intervals to  $f(t)$  are redundant so the super-resolution algorithm cannot be applied. The other way that the edges move is by moving along the other sampling interval (sub-pixel). In this case,

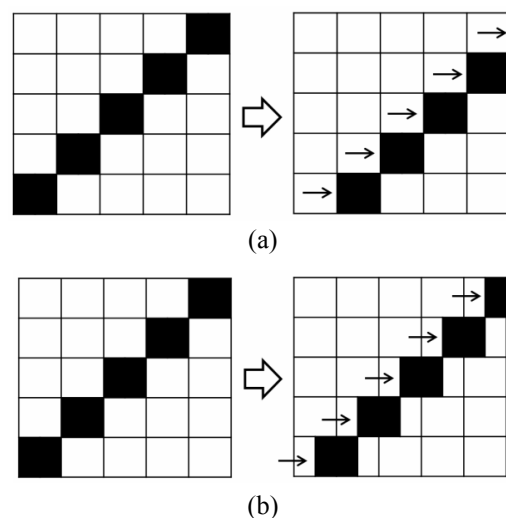
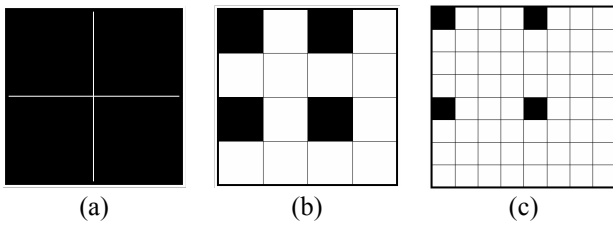


Fig. 4. Motion with (a) integer unit; (b) sub-pixel unit,  $f(t-1)$ : left column and  $f(t)$ : right column



**Fig. 5.** Example of a sub-pixel: (a) Integer-pixel; (b) 1/2 pixel; (c) 1/4 pixel

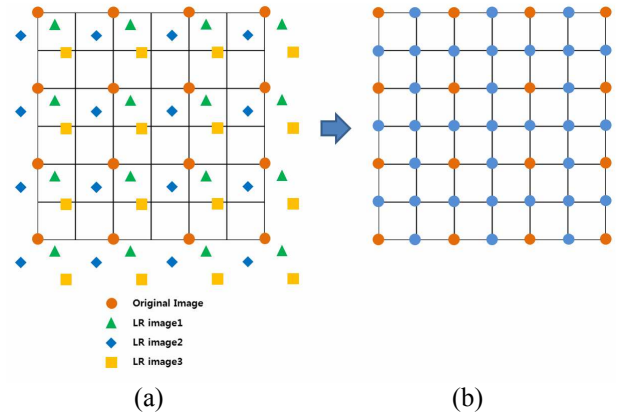
the two time intervals possess different pixel information on the same edges so the super-resolution algorithm can be applied [2]. Therefore, the registration of images is used to distinguish the movement of integer pixel and sub pixel intervals in MSR.

Image registration is classified by processing the domain. One of methods is using discrete Fourier transform (DFT) to input images including target frame, and registration is done by using the normalized cross power spectrum at the frequency domain [9]. Registration can also be done in the spatial domain with a motion vector of each block by using motion estimation or optical-flow methods.

If the registration is done in the frequency domain, we do not need specific sub-pixel interpolation and can acquire information for registration even up to the point unit but need an assumption of global motion with translation [9]. If we have an image containing a natural view or objects with negligible motion, the difference of each frame coming from the movement of the camera usually gives the global motion with translation. But, most of the video contents do not meet this condition; that is, most of the images contain the local motion so that the registration process done in the frequency domain cannot be used widely.

However, the registration process done in spatial domain divides the input image into specific sizes of patches: even in the target frame, and then acquires information for registration from each patch. This approach does not need the global translation assumption and is more effective to apply to images. We define a block as each patch. But, dividing an input image into patches is generally a time consuming job and if we have inaccurate motion vectors, it gives inaccurate result [6]. Also, for the estimation of the motion vectors in the sub-pixel unit, we need the step of sub-pixel interpolation first. But because the registration process done in spatial domain is free from the global translation constraint, majority of the previously proposed MSR algorithms use a registration process done in the spatial domain.

Fig. 5(a) shows a 2\*2 pixel block. (b) and (c) show 1/2 pixel and 1/4 pixel blocks respectively. To improve the accuracy of the sub-pixel, sub-pixel interpolation can be repeated. Sub-pixel interpolation is also a way of increasing the resolution and bi-linear and bi-cubic interpolation are some of the common methods. For



**Fig. 6.** (a) Registration points; (b) High-resolution grid

example, interpolation of the 1/2 pixel unit of sub-pixel done through common interpolation methods to increase the resolution is part of the MSR and is not part of the results we envisioned.

After the interpolation of the sub-pixel, the approximation of movement within the grid or optical flow method is used to register the pixel units.

In the second step of the MSR algorithm, the values of pixels of high-resolution grid are calculated. Fig. 6 shows this step schematically. Performance of MSR algorithm is also affected by the specific interpolation method. The pixel value corresponding to each of the high-resolution grid in Fig. 6(b) is determined by weighted-sum of each pixel value of registration points in Fig. 6(a) and each weight is decided inversely proportionally to the distance between the target grid and each of the registration points.

After reconstructing a high-resolution image through these two steps, we still need a post-process for de-blurring or de-blocking to get the final result. Post processing step is out of scope of this paper. The details are available in [10].

## 2.2. Sub-pixel motion estimation with 6-tap FIR filter

The rest of this section including this part will explain in detail the steps of the proposed Multi-frame based super resolution algorithm. Especially in part 2.3 and 2.4, we will talk about the new algorithm. As mentioned in section 1, in order to differentiate the movement of sub-pixels or integer pixels between the target image and the input image, we need to interpolate the sub-pixels first.

This process is also a variation of increasing the resolution and can be done through many methods including bi-linear, bi-cubic and Lanczos interpolation. However, in this paper, we will interpolate sub-pixels using the H.264/AVC 6-tap FIR filter that has been experimentally proven to be effective [8]. There are 7 continuous images that we want to use the super-resolution algorithms on including the target image. We will use the 6-tap filter and bi-linear interpolation to find up to 1/4 pixels (quarter-pixel) for

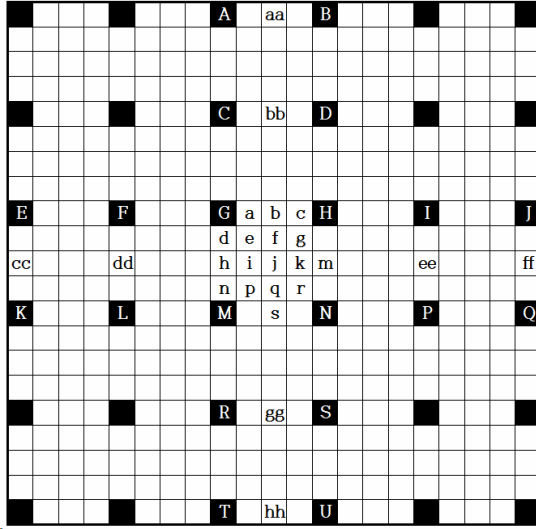


Fig. 7. Pixel chart to describe 6-tap FIR filter

each image. First, we introduce the 1/2 pixel (half-pixel) interpolation in Eq. (1) as follows [12] :

$$\begin{aligned}
 b_1 &= (E - 5F + 20G + 20H - 5I + J) \\
 h_1 &= (A - 5C + 20G + 20M - 5R + T) \\
 b &= (b_1 + 16) \gg 5 \\
 h &= (h_1 + 16) \gg 5 \\
 j_1 &= cc - 5dd + 20h_1 + 20m_1 - 5ee + ff \\
 j &= (j_1 + 512) \gg 10
 \end{aligned} \quad (1)$$

where  $b$  and  $h$  are pixel values of the half-pixel unit. Also, the operator ‘ $\gg$ ’ is a simple shift operator. For example, when  $x \gg y$ , it means that  $x$  is divided by 2 raised to the power of  $y$ .  $A, C, G, M, R, T, E, F, H, I$  and  $J$  are integer unit pixel values of the original image as shown in Fig. 7. In Fig. 7, the capital letters in the black squares correspond to the original integer pixels.

Now, quarter-pixels can be calculated by using the bi-linear interpolation method as shown in Eq. (2) [12] :

$$a = (G + b + I) \gg 1 \quad (2)$$

If the 1/2 and 1/4 sub pixels are found using Eq. (1) and (2), total of 7 images are multiplied by 4 times horizontally and vertically to become 16 times the size of the original image. The resolution does increase but this is not the final pixel value we want: this temporary increase is merely used to register the sub-pixel unit. However, based on the registration results, the interpolated sub-pixels have a direct effect on the resulting quality because they are used for the kernel based pixel value estimation mentioned in section 2.5.

### 2.3. Normalization of motion vector

After the interpolation of sub-pixels, the estimation of

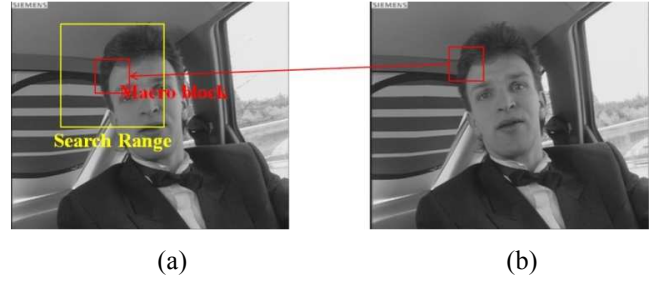


Fig. 8. Sub-pixel motion estimation between: (a)  $(t-1)^{th}$  frame and (b)  $t^{th}$  frame

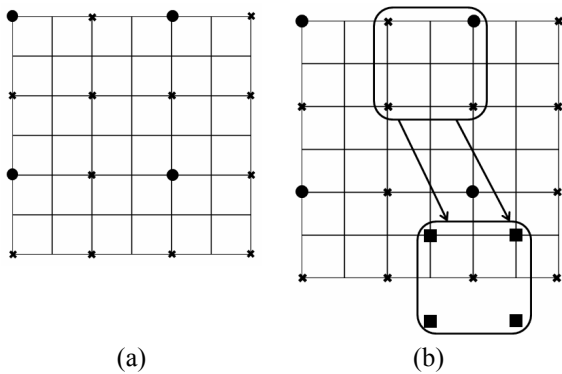
the movement based on the normalization of the motion vector is used to register the input image. The estimation of movement is based on the compression principle of removing time-based redundancies as shown in Fig. 8. If we say that (a) is the image from  $f(t-1)$ , and (b) is the image from  $f(t)$ , the two images only have a very slight difference as mentioned in 2.1. At this time, we divide each image into same-sized blocks then find the most similar block from  $f(t-1)$  image compared to a specific block from  $f(t)$ . After finding the corresponding block, figure out how far the block is from the  $f(t)$  block horizontally and vertically. This information is called a motion vector. This process is repeated for all the blocks in the  $f(t)$  image and when coding the image, we can create good compression effect through using the  $f(t-1)$  location and the vector difference found above. In this paper, compression is not the reason for using the estimation of movement. We use the estimation of movement to figure out the location relation of the images using blocks.

In this paper, because we use a  $2 \times 2$  size of block for motion estimation, image registration even with complicated or local motion is possible. We use SAD (sum of absolute difference) as the criterion for motion estimation as in Eq. (3) [13] :

$$SAD(i, j) = \sum_{y=0}^1 \sum_{x=0}^1 |B_t(x, y) - B_p(i+x, j+y)| \quad (3)$$

where  $SAD(i, j)$  means the value of SAD at  $(i, j)$ , and  $x$  and  $y$  have values of 0 or 1 and represent coordinates in a macro block.  $B_t(x, y)$  is the  $(x, y)$  location's pixel value within the target image's unit block.  $B_p(i+x, j+y)$  represents the  $(i+x, j+y)$  location's pixel value within the registered area of the other input images. The position with the lowest SAD value within the search range is considered the most similar block to the one in  $f(t)$ .

Since motion estimation process is a time-consuming task, many fast algorithms exist [14, 15]. But there is a trade-off between time efficiency and estimation accuracy. To get a better quality of the result of the super-resolution algorithm, we try to find a more accurate motion vector. Consequently we use a traditional full search method for the motion estimation in this paper.



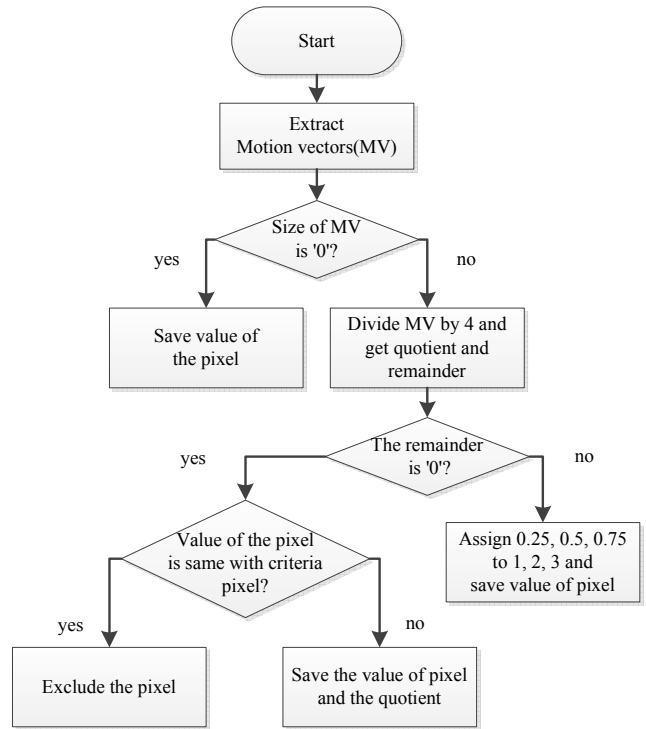
**Fig. 9.** Pixel chart for depicting registration: (a) before registration; (b) after registration

Next, we normalize each block location found using the motion estimation. In the existing multi-frame based super-resolution algorithms, it has been assumed that the sub-pixel movement in the input image happened only in the integer pixel [5, 9]. This constraint makes it harder to apply super-resolution algorithm to video sequences. However, if the normalization of the movement vector process as mentioned in this paper is used, the super-resolution algorithm can be applied even if the integer pixel unit or the movement within the image is large.

After estimating the movement of the 2\*2 block, the relationship between images can be achieved as shown in Fig. 9(b). Fig. 9(a) represents the pixel chart before interpolation. The original pixel from the target image is represented as a circle, and three pixels appear between each circle after 1/4 unit sub-pixel interpolation. The goal of this paper is to increase the resolution by 2 times both vertically and horizontally compared to the original image using the super resolution algorithm. The final value estimation of the pixel location is a cross-shaped figure. We call this the target pixel. Although the sub-pixel value can be found using sub-pixel interpolation, the resulting value is not the target value: it is only used for motion estimation based on Eq.(3) and used as the base information for finding the target pixel value. After the motion estimation based on Eq.(3), results similar to those in Fig. 9(b) can be seen and horizontal and vertical motion vectors can be gained. Fig. 10 shows the block diagram of the motion vector normalization.

At first, we apply 2\*2 block matching motion estimation to the 6 input images. As a result, we obtain 6\*2\*2 motion vectors. If there is no motion (motion vector is 0), the distance between the target pixel and the registered pixel is defined as 0. If the motion vector has a value other than 0, we divide it by 4 because we extend the input image as much as 4 times in both the vertical and horizontal directions for the quarter pixel accuracy motion estimation. We assign the value of 0.25, 0.5 and 0.75 to the remainders 1, 2 and 3, respectively and save them with quotient.

In other words, in the case that movement that is equal



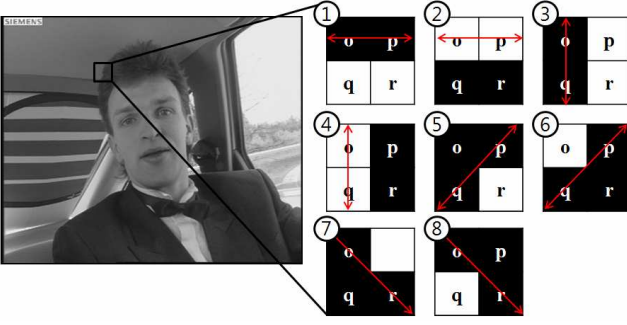
**Fig. 10.** Normalization of motion vector

to or larger than a pixel exists, it can still be used for the target pixel estimation by using weighted values. The pixel value found using the normalized motion vector and the corresponding vector is entered into the kernel that will be described in section 2.5. By normalizing the motion vector and the corresponding vector, the super-resolution algorithm can be used even if there is movement bigger or equal to a pixel unit during the registration stage.

### 2.4 Analysis of edge patterns

Existing multi-frame based super-resolution (MSR) algorithms have used registration points during the interpolation process without considering the accuracy of the registration or patterns of the edges. In this case, the interpolated pixel value may not be correct around the edge area and this causes a downgrade of the quality of the algorithm. To avoid this effect, we have to consider the edge patterns in the interpolation process.

Because we used the 2\*2 block in motion estimation, we used the 2\*2 block to categorize edge patterns as Fig. 11. Eq. (4) shows parameters for defining each edge pattern are shown through the 8 blocks. If  $\alpha_1$  and  $\alpha_2$  in Eq. (4) are very small and  $\alpha_3$  and  $\alpha_4$  are relatively large, then we define the edge pattern as ① in Fig. 11. If the current block has edge pattern of ① in Fig. 11, we use the pixels at the position of o and p in each matched block of seven input frames as the registered pixel to interpolate the target pixel a. Also, by adjusting the threshold value, we can control the sensitivity of the pattern-analyzing algorithm.



**Fig. 11.** Edge patterns of 2\*2 block for pattern analysis

$$\begin{aligned} |o - p| &= \alpha_1, & |q - r| &= \alpha_2, & |o - q| &= \alpha_3, \\ |p - r| &= \alpha_4, & |o - r| &= \alpha_5, & |p - q| &= \alpha_6 \end{aligned} \quad (4)$$

The sensitivity of the outline pattern found by differentiating each  $\alpha$ -value can be adjusted using the experimental threshold value. The algorithm is applied to a high-frequency area which is a small part of the image and the PSNR will increase slightly if the algorithm is not used taking into account the threshold value. In section 3, the optimal threshold value found through experimentation with 10 different sequences is used.

However, there still may be a big difference between the values of target pixel and register points even after the considering the edge patterns. Thus, it is possible that we may not get the satisfied interpolation result. Therefore, we tried to get rid of some registered points which have much different value with the target pixel by using Eq. (5).

$$|V(P_t) - V(P_r)| < T \quad (5)$$

where  $V(P_t)$  represents the value of the target pixel and  $V(P_r)$  is the value of registered points. In the same manner,  $T$  is the optimal threshold value based on experimentation.

### 2.5 Kernel regression based interpolation

After selecting the registered points by using algorithms in 2.3 through 2.4, we needed to find out the target pixel value by using these registered points. In the proposed algorithm, we used the kernel regression based on the pixel value estimation algorithm. This algorithm calculates the value of the target pixel by non-parametrically using the registered pixel values. Unlike the parametric method, it estimates the target pixel value by a variety of kernel estimators with the registered pixel values. Eq. (6) is the most widely used Nadaraya-Watson kernel estimator based on a Gaussian model [16, 17]

$$\hat{f}_{NW}(x) = \frac{\sum_{i=1}^N Y_i K\left(\frac{x_i - x}{h}\right)}{\sum_{i=1}^N K\left(\frac{x_i - x}{h}\right)} \quad (6)$$

where the  $x_i - x$  represents the value resulting from  $i^{\text{th}}$  registered point of motion vector normalization processes and the  $Y_i$  is its pixel value.  $h$  indicates the bandwidth of the kernel estimator.  $K(u)$  of Eq.(7) is the Gaussian based kernel.

$$K(u) = 0.6171 \exp(-u^2/2) I(-1/2 \leq u \leq 1/2) \quad (7)$$

$$I(z) = \begin{cases} 1 & \text{a is true} \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

$I(z)$  of the Eq.(8) is an indicator and has the value of 1 if  $z$  is ‘‘true’’ and 0 otherwise, and  $u$  which is the input of the kernel shall not be less than 0. If  $h$  gets larger, more registered pixels can be used for the interpolation but it results in blurring effect. In the other case, some noises could occur when less registered pixels used.

### 3. Experimental Results

In this section, we show the performance of the proposed algorithm with experimental results. First, we subsampled 30 successive frames of the original video sequences with the size of 352\*288 (CIF), 832\*480 and a factor of 4. There are five 352\*288 sized images: ‘Mother and daughter’, ‘News’, ‘Carphone’, ‘Coastguard’ and ‘Tempete’. There are four 832\*480 images: ‘BasketBall’, ‘BQTmall’, ‘PartyScene’ and ‘Racehorses’. We compared the performance of the proposed algorithm with bi-linear interpolation, single-frame based super-resolution (SSR) that is based on discrete wavelet transform (DWT) [3] and an existing multi-frame based super-resolution [8]. We used PSNR in Eq. (9) for objective comparison [18].

$$\begin{aligned} \text{PSNR}(f, g) &= 10 \log_{10}(255^2 / \text{MSE}(f, g)) \\ \text{where, } \text{MSE}(f, g) &= \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N (f_{ij} - g_{ij})^2 \end{aligned} \quad (9)$$

where  $f$  and  $g$  indicate the original and the result images and  $M$  and  $N$  are the width and height of an image, respectively.

The first experimental procedure is as follows. Total of four algorithms including the one proposed in this paper is applied to the Luminance component of the 5 sub-sampled images obtained from the 352\*288 sized images to restore the image to its original size. Then the PSNR is compared. Table 1 shows the measured PSNR values for each test

**Table 1.** PSNR of each test sequence (352\*288) (dB)

	Bi-linear	SR[3]	SR[8]	Proposed
Mother and daughter	31.32	34.83	34.32	35.04
News	24.92	27.93	27.95	28.35
Carphone	27.15	28.88	28.82	29.19
Coastguard	24.78	26.74	26.92	27.28
Tempete	22.99	24.28	24.30	24.57

sequence. The results are average values because we used 30 frames of each sequences. As shown in Table 1, the performance of the proposed algorithm is better than that of the other algorithms.

Next, for the 832\*480 sized images, the RGB components

**Table 2.** PSNR of each test sequence (832\*480/R) (dB)

	Bi-linear	SR[3]	SR[8]	Proposed
BasketBall	32.80	38.33	38.92	39.46
BQTMall	30.58	35.04	36.15	36.27
PartyScene	28.63	32.31	32.99	34.17
Racehorses	31.23	37.30	38.00	39.73

**Table 3.** PSNR of each test sequence(832\*480/G) (dB)

	Bi-linear	SR[3]	SR[8]	Proposed
BasketBall	34.00	38.72	39.23	40.21
BQTMall	30.58	35.56	36.27	36.45
PartyScene	29.03	32.31	33.00	34.22
Racehorses	31.14	37.41	37.55	39.78

**Table 4.** PSNR of each test sequence(832\*480/B) (dB)

	Bi-linear	SR[3]	SR[8]	Proposed
BasketBall	32.93	37.42	39.51	39.72
BQTMall	30.91	35.29	36.02	36.58
PartyScene	29.28	32.37	33.52	34.17
Racehorses	31.57	37.48	37.69	38.82

are individually sub-sampled then for each component the four algorithms are applied to restore the image. Also, the PSNR is compared.

Tables 2 to 4 show the restoration results for each RGB components of all the images.

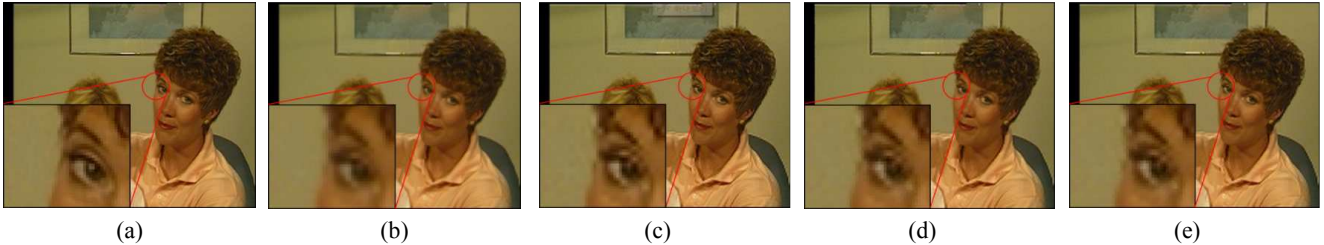
The results from the tables above show that the proposed algorithm works well for both of the components. Especially, for the experiment involving the larger images, the PSNR value difference is noticeable. Through looking at Figs. 12 to Fig. 16, the resolution difference from the first experiment can be seen.

We can see a big difference between the bi-linear algorithm and the other algorithms just as in Table 1. In the enlarged part of each figure, we could find the performance differences among the results. Especially, the proposed algorithm has a clearer boundary around face area.

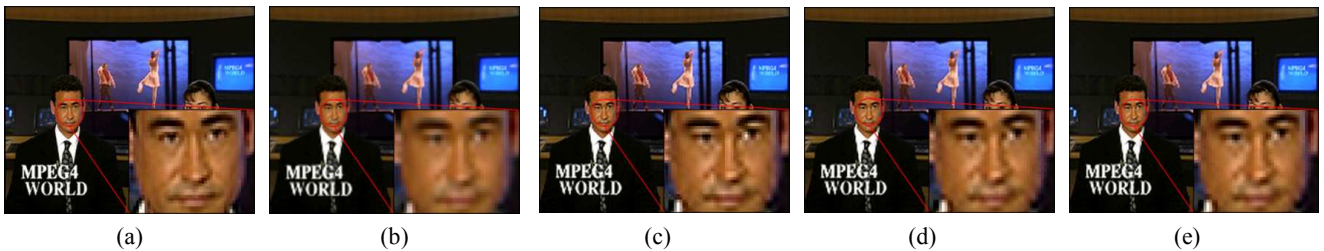
From the enlarged image in Fig. 15(d), noise can be seen from the registration error within the vicinity of the boundary between the water and the boat. Likewise, identical noise can be seen on the flower petal in Fig. 16(d).

Corresponding impulse noise cannot be observed in each and every image but can sometimes result in the decrease of the image resolution. In Figs. 15 and Fig. 16, it can be seen that these types of noise is removed from the resulting images.

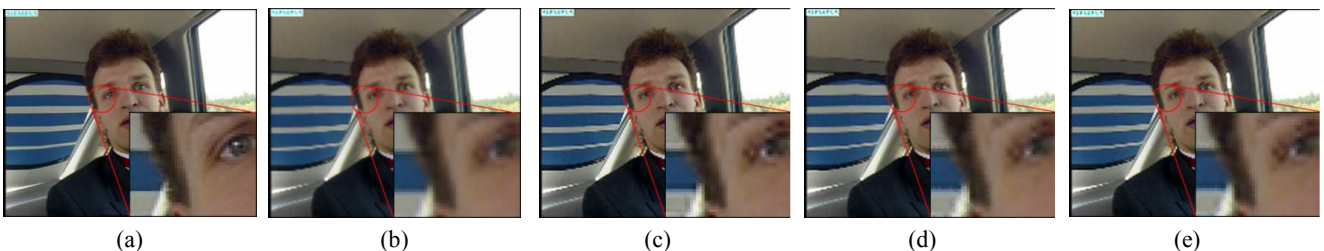
The PSNR difference is minor between the proposed



**Fig. 12.** Mother and daughter (6<sup>th</sup> frame) (a) ground-truth; (b) bi-linear; (c) SR[3]; (d) SR[8] and (e) the proposed



**Fig. 13.** News (8<sup>th</sup> frame) (a) ground-truth; (b) bi-linear; (c) SR[3]; (d) SR[8] and (e) the proposed



**Fig. 14.** Carphone (19<sup>th</sup> frame) (a) ground-truth; (b) bi-linear; (c) SR[3]; (d) SR[8] and (e) the proposed

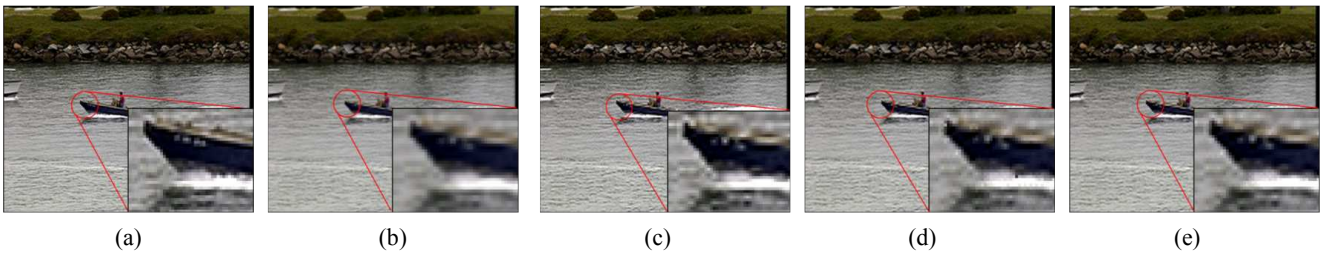


Fig. 15. Coastguard (4<sup>th</sup> frame): (a) ground-truth; (b) bi-linear; (c) SR[3]; (d) SR[8] and (e) the proposed



Fig. 16. Tempete (3<sup>th</sup> frame) (a) ground-truth; (b) bi-linear; (c) SR[3] ; (d) SR[8] and (e) the proposed

algorithm and the existing algorithm because the proposed algorithm applies to the high frequency component of the image. The area of the image that the high frequency component takes up is small. However, in all the experimental images, the proposed algorithm gave better performance than other algorithms in all experimental results.

The resulting video from the second experiment can be seen at <ftp://128.134.65.1> from using a guest ID.

Through PSNR estimation and subjective comparison of the resolution, each image had slightly different results. However, the proposed algorithm showed better results than that of simple interpolation techniques and the existing SSR and MSR super resolution algorithms. The existing SSR algorithms provide little information or lack of training sets in general and that limits the increase in resolution. However, the existing SSR algorithms provide more efficient processing time than the MSR algorithms and can be used as alternatives for scenes with fade in or fade out effects where the MSR algorithm cannot be used.

Although most MSR algorithms take a long time to register the image, it is possible to obtain a large sum of information regarding the target image by controlling the number of images used for increasing the resolution. Therefore, above average level of resolution is guaranteed. Specifically in the case of registration, the recently developing research in the parallel computing field will be apt for the repeated processes that are applied to all areas of the image. Research is needed for enhancing the accuracy and the processing time for the MSR super resolution algorithms.

#### 4. Conclusion

In this paper, a new super-resolution algorithm was proposed by using successive frames for generating high-resolution frames with better quality than that of other conventional interpolation methods. The proposed super-resolution algorithm includes two new parts. One is the motion vector normalization and the other is analysis of the patterns of the edges. The experimental results showed that the proposed algorithm performs better than other conventional algorithms.

#### Acknowledgements

This work was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT and Future Planning (MSIP). (2011-0006791)

#### References

- [1] G. B. Kang, Y. S. Yang and J. H. Kim, "A study on interpolation for enlarged still image," *KIICE General Conference*, pp. 643-648, Bukyung Univ., Korea, May 2001.
- [2] S. C. Park, M. K. Park and M. G. Kang, "Super resolution image reconstruction: a technical overview," *IEEE Signal Processing Magazine*, vol. 20, no. 3, pp. 21-36, May, 2003.
- [3] J. M. Lim and J. Yoo, "Super-resolution algorithm using discrete wavelet transform for single-image," *KOSBE Journals*, vol. 17, no. 2, pp. 344-353, Mar.



- 2012.
- [4] H. Chang, D. Y. Yeung and Y. Xiong, "Super-resolution through neighbor embedding," Conference on Computer Vision and Pattern Recognition Proceedings of the 2004, *IEEE Computer Society*, vol. 1, pp. 275-282, Jun. 2004.
- [5] H. G. Ha, I. S. Jang, K. W. Ko and Y. H. Ha, "Sub-pixel shift estimation in noisy image using iterative phase correlation of a selected local region," *IEEK Journals*, vol. 47, no. 1, pp. 103-111, Jan. 2010.
- [6] S. Farsiu and P. Milanfar, "Kernel regression for image processing and reconstruction," *IEEE Trans. on Image Processing*, vol. 16, no. 2, pp. 349-366, Feb. 2007.
- [7] S. C. Jeong and Y. L. Choi, "Video super resolution algorithm using bi-directional overlapped block motion compensation and on-the-fly dictionary training," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 21, no. 3, pp. 274-285, Mar. 2011.
- [8] S. C. Kwon and J. Yoo, "Super resolution algorithm by motion estimation with sub-pixel accuracy using 6-tap FIR filter," *KICS Journals*, vol. 37, no. 6, pp. 464-472, June 2012.
- [9] Y. M. Seong and H. W. Park, "Super resolution image reconstruction using phase correlation based subpixel registration from a sequence of frames," *IEEK general conference*, Seoul Univ., Korea, pp. 481-484, Nov. 2005.
- [10] A. L. Peter, A. Joch, J. Lainema, G. Bjontegaard and M. Karczewicz, "Adaptive de-blocking filter," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 614-619, July 2003.
- [11] T. H. Kim, Y. S. Moon and C. S. Han, "Estimation of real boundary with subpixel accuracy in digital imagery," *KSPE Journals*, vol. 16, no. 8, pp. 16-22, Aug. 1999.
- [12] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE trans. on circuits and systems for video technology*, vol. 13, no. 7, pp. 560-576, July 2003.
- [13] H. M. Wong, O. C. Au, A. Chang, S. K. Yip and C. W. Ho, "Fast mode decision and motion estimation for H.264(FMDME)," *IEEE Int. Symposium on Circuits and Systems*, pp. 21-24, Greece, May 2006.
- [14] Y. Ismail, J. B. McNeely, M. Shaaban, H. Mahmoud and M. A. Bayoumi, "Fast motion estimation system using dynamic models for H.264/AVC video coding," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 22, no. 1, pp. 28-42, Jan. 2012.
- [15] N. Hirai, T. Kato, T. Song and T. Shimamoto, "An efficient architecture for spiral-type motion estimation for H.264/AVC," *IEEK General Conference in the Korea*, Je-ju, Jul. 2009.
- [16] S. Farsiu and P. Milanfar, "Kernel regression for image processing and reconstruction," *IEEE trans. on Image Processing*, vol. 16, no. 2, pp. 349-366, Feb. 2007.
- [17] V. K. Asari, M. N. Islam and M. A. Karim, "Super resolution enhancement technique for low resolution video," *IEEE trans. on Consumer Electronics*, vol. 56, no. 2, pp. 919-924, May 2010.
- [18] S. Winkler, "The evolution of video quality measurement: From PSNR to hybrid metrics," *IEEE trans. on Broadcasting*, vol. 54, no. 3, pp. 660-668, Sep. 2008.



**Soon-chan Kwon** He received B.S. degree in electronics engineering from Kwangwoon University in 2012. His research interests are super-resolution, non-linear filtering and image restoration.



**Jisang Yoo** He received the B.S. and M.S. degrees from Seoul national university, Seoul, Korea in 1985 and 1987, all in electronics engineering, and Ph.D. degree from Purdue University, West Lafayette, IN, USA, in electrical engineering in 1993, respectively.

From September 1993 to august 1994, he worked as a senior research engineer in industrial electronics R&D center at Hyundai Electronics Industries Co., Ltd, Incheon, Korea, in the area of image compression and HDTV. He is currently a professor with the department of electronics engineering, Kwangwoon University, Seoul, Korea. His research interests are in signal and image processing, nonlinear digital filtering, and computer vision. He is now leading 3DTV broadcast trial project in Korea.