IPSJ Transactions on Computer Vision and Applications Vol. 1 277–287 (Dec. 2009)

Research Paper

Construction Method of Efficient Database for Learning-Based Video Super-Resolution

Kiyotaka Watanabe,[†1] Yoshio Iwai,[†2]
Tetsuji Haga,[†1] Koichi Takeuchi[†1]
and Masahiko Yachida[†3]

There are two major problems with learning-based super-resolution algorithms. One is that they require a large amount of memory to store examples; while the other is the high computational cost of finding the nearest neighbors in the database. In order to alleviate these problems, it is helpful to reduce the dimensionality of examples and to store only a small number of examples that contribute to the synthesis of a high quality video. Based on these ideas, we have developed an efficient algorithm for learning-based video super-resolution. We introduce several strategies to construct an efficient database. Through the evaluation experiments we show the efficiency of our approach in improving super-resolution algorithms.

1. Introduction

As a result of progress in video technology, high definition (HD) consumer devices have recently become more widespread. For example, full HD video cameras with about two million pixels are currently available. As for digital still cameras, resolution of over ten million pixels has been achieved. It is, however, very difficult to achieve such a very high resolution (HR) video sequence because of the limited sweep time of the camera. Hence, HR is incompatible with a high frame rate. There are certain HR video cameras available for special use, such as a digital cinema, but these are very expensive and are thus unsuitable for general use.

To enhance the resolution of a HR image, simple interpolation methods such as the bilinear or bicubic spline are widely used, but these methods tend to generate blurred images. A more sophisticated technique to synthesize a HR image is super-resolution (SR). SR algorithms are roughly classified as either reconstruction-based SR or learning-based SR.

In reconstruction-based SR techniques, the basic assumption for increasing the spatial resolution is the availability of multiple low resolution (LR) images captured from the same scene. Each of these LR images must have a different subpixel shift. SR algorithms are then used to estimate relative motion information between these LR images (or video sequences) and increase the spatial resolution by fusing them into a single frame. Conventional techniques for obtaining a super-resolved image from still images have been summarized in the literature[1)]. By extending the idea of SR to the temporal domain, Shechtman, et al.[2)] proposed space-time SR, which enhances both spatial and temporal resolution simultaneously.

Learning-based SR algorithms extract a relationship between the HR images and their corresponding LR ones, which are used as training data. The algorithms construct a large database of examples from the training data set. The resolution of the input LR images can then be enhanced by adding high frequency detail via a nearest neighbor search in the database. Baker and Kanade[3)] provided fundamental limits on reconstruction-based SR, and demonstrated that facial images and text images can be super-resolved by using a large number of examples. Freeman, et al.[4)] showed that the idea of learning-based SR can be applied to general images. Indeed, several learning-based SR algorithms have been proposed for both still images[5)–7)] and video sequences[8),9)]. Learning-based SR algorithms are characterized by the ability to reconstruct a HR image from a single image using a training database.

There are, however, two major problems with learning-based SR algorithms. One is that they require a large amount of memory to store examples, while the other is the high computational cost of finding nearest neighbors in the database. Therefore, it is less practical to conduct learning-based SR for video sequences since a large amount of data has to be processed. In this paper, we propose an efficient algorithm for learning-based video SR. To improve the efficiency, we introduce a compact feature vector using discrete cosine transform (DCT)

[†1] Advanced Technology R&D Center, Mitsubishi Electric Corporation
[†2] Graduate School of Engineering Science, Osaka University
[†3] Faculty of Information Science and Technology, Osaka Institute of Technology

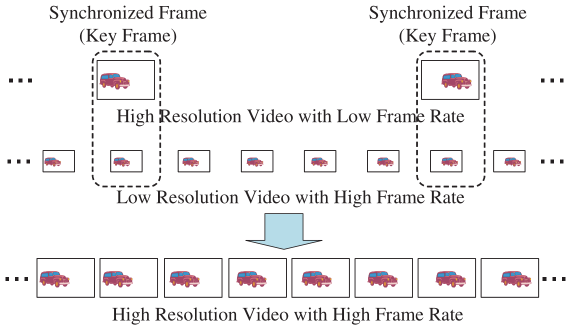

Synchronized Frame
(Key Frame)
Synchronized Frame
(Key Frame)
High Resolution Video with Low Frame Rate
Low Resolution Video with High Frame Rate
High Resolution Video with High Frame Rate

Fig. 1 Problem formulation.

coefficients, and design procedures to determine which examples need to be stored in the database. Through the evaluation experiments we show the efficiency of our approach in improving SR algorithms.

The algorithm proposed in this paper is based on the problem formulation shown in **Fig. 1**. Our algorithm synthesizes HR video with a high frame rate from two video sequences, that is, a HR sequence with low frame rate and a LR sequence with high frame rate. We assume that there are synchronized frames in the HR and LR sequences, which are referred to as "key frames" in this paper. The dual sensor camera proposed by Nagahara, et al.[10] is able to capture two such video sequences simultaneously. The concept of a dual sensor camera is shown in **Fig. 2**. The two video sequences captured by the camera have the same field of view. Key frames can be obtained by feeding a pulse signal into the camera. Matsunobu, et al.[11] and Watanabe, et al.[12],[13] proposed algorithms to synthesize a HR video with high frame rate using the dual sensor camera. In this paper we propose a novel algorithm that constructs an example database using key frames as training data, and conducts SR for all LR frames except the key frames.

The rest of the paper is organized as follows. We explain the strategies used to improve the efficiency of SR algorithms in the next section. Section 3 introduces the proposed algorithm for video SR. In Section 4 we show the effectiveness of our approach in improving SR algorithms through experimental results using moving picture experts group (MPEG) test sequences. We also compare the quality of synthesized video using the proposed method and conventional ones. Section 5 concludes this paper.

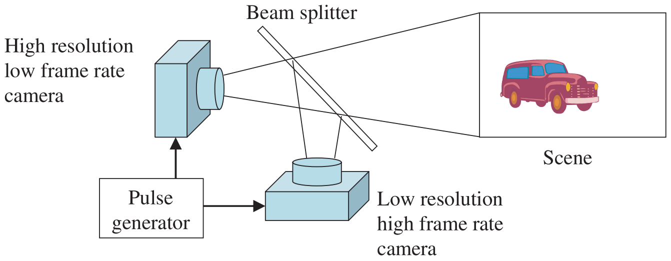


Fig. 2 Concept of dual sensor camera.

2. Strategy to Improve the Efficiency of SR

As mentioned in the previous section, learning-based SR requires a large amount of memory and high computational cost. To alleviate these problems, it is helpful to adopt the following strategies.

(1) Apply a fast searching algorithm.
(2) Reduce the dimensionality of examples stored in the database.
(3) Reduce the number of examples stored in the database.

With respect to (1), Freeman, et al.[4] used a fast searching algorithm[14] to find multi-dimensional nearest neighbors. We use the approximate nearest neighbor (ANN) search algorithm[15] to find the nearest neighbors quickly in exchange for allowing a small error.

Strategies (2) and (3) are effective because both the time and space complexity of learning-based SR algorithms are dependent on the dimensionality and the number of examples stored in the database. As for (2), Bishop, et al.[8] used principal component analysis (PCA) to reduce the dimensions of the feature vector from 147 to 20. However, PCA is itself computationally expensive and is thus unsuitable for video SR, which must process a large amount of data. In the proposed method, DCT is applied to each block and then low frequency components of the DCT coefficients are extracted to compose a feature vector.

Strategy (3) has not been considered in previous studies. It is efficient to store only a small number of examples that can contribute to the synthesis of high quality video. Under the condition of limited memory space, we introduce several ideas to help decide which examples are to be replaced (or stored) in the database. First, we assume that although the size of the database is limited, it is large enough to store all the examples extracted from one key frame. Through the experiments in Section 4.1, we compare the following procedures for the replacement of examples.

Fix Construct the database using the first key frame, and then do not update it.

Clear For every key frame, delete all examples in the database and then reconstruct it, using the newly input key frame as training data.

Random If the database is not full, add the new example to it. Otherwise, replace a randomly selected example in the database with the new one.

FC (Frequency Count) If the database is not full, add the new example to it. Otherwise, replace the example that is referred to least frequently in the database with the new one.

LRU (Least Recently Used) If the database is not full, add the new example to it. Otherwise, replace the example least recently referred to in the database with the new one.

Next, we consider the more difficult condition in which all the examples extracted from one key frame cannot be stored in the database. We present an algorithm that decides sequentially which examples are to be stored in the database in Section 3.1.

3. Proposed Algorithm

We extend learning-based SR methods for still images[5),6)] to video sequences, and adopt DCT coefficients as feature vector components. In the proposed method, an SR procedure is conducted for the luminance component since human vision is insensitive to spatial variations in color. The chrominance components of the synthesized frames are interpolated using bicubic spline interpolation from the LR frames to reduce the computational cost.

3.1 Construction of Database Using Key Frames

We consider two *sub*-databases $\mathcal{D}_1$ and $\mathcal{D}_2$. The proposed method first constructs $\mathcal{D}_1$, which is composed of examples randomly selected in the key frame. Then, it constructs $\mathcal{D}_2$, which is composed of examples with large distances to nearest neighbors in $\mathcal{D}_1$. Finally, by integrating $\mathcal{D}_1$ and $\mathcal{D}_2$, example database $\mathcal{D} = \mathcal{D}_1 \cup \mathcal{D}_2$ is constructed. Let n be the upper limit of the number of examples that can be stored in $\mathcal{D}$, and $|\mathcal{D}|$ denotes the number of examples stored in $\mathcal{D}$.

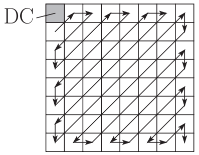


Fig. 3 Zigzag scan for AC coefficients of DCT.

Given the pair consisting of a HR image I_H and LR image I_L (key frame), example database $\mathcal{D}$ is constructed according to the following procedure.

(1) Enlarge I_L using bicubic spline interpolation to generate an image I_H^ℓ with the same size as I_H.

(2) Set $\mathcal{D}_1 = \mathcal{D}_2 = \emptyset$, and fix n_1 and n_2, the upper limits of the number of examples that can be stored in $\mathcal{D}_1$ and $\mathcal{D}_2$, respectively, where $n = n_1 + n_2$. Arrange a priority queue $\mathcal{Q}$ with length n_2. Initially, set $\mathcal{Q} = \emptyset$.

(3) (Random sampling of examples) Repeat the following steps until $|\mathcal{D}_1| = n_1$.

  (a) Extract a block B_r^ℓ of size $K \times K$ from I_H^ℓ randomly and without duplication. The extracted block B_r^ℓ is added to the set $\mathcal{B}$.

  (b) Calculate the contrast $c^\ell = E[|x^\ell - \mu^\ell|]$, where $\mu^\ell = E[x^\ell]$ is the average number of pixels $x^\ell \in B_r^\ell$.

  (c) If c^ℓ is smaller than a threshold θ_c, the remaining steps are not executed for this block. Otherwise, proceed to the next step.

  (d) Conduct DCT for B_r^ℓ to obtain DCT coefficients $\mathcal{C}[B_r^\ell]$, and extract d AC coefficients of $\mathcal{C}[B_r^\ell]$ in a zigzag scan order (as shown in **Fig. 3**, where $K = 8$) to compose a d-dimensional vector $\boldsymbol{v}_r^\ell$.

  (e) Extract a block B_r^h of size $K \times K$ from I_H to obtain DCT coefficients $\mathcal{C}[B_r^h]$.

  (f) Add the example $(\boldsymbol{v}_r^\ell, \mathcal{C}[B_r^h])$ to $\mathcal{D}_1$.

(4) (Example selection using nearest neighbor search) Conduct the following steps in raster-scan order.

  (a) Extract a block $B_e^\ell \notin \mathcal{B}$ of size $K \times K$ from I_H^ℓ, i.e., B_e^ℓ is a block that has not been extracted from I_H^ℓ in Step (3a).

(b) If the contrast of block B_e^ℓ is smaller than a threshold θ_c, the following process is not executed for this block. Otherwise, execute the process given in Steps (3d) and (3e) for B_e^ℓ to obtain the example $(\boldsymbol{v}_e^\ell, \mathcal{C}[B_e^h])$.

(c) Find the nearest neighbor of $\boldsymbol{v}_e^\ell$ in $\mathcal{D}_1$. We denote the found item as $\boldsymbol{v}_{\mathrm{NN}}^\ell$. Measure the Euclidean distance $\mathrm{dist}(\boldsymbol{v}_e^\ell, \boldsymbol{v}_{\mathrm{NN}}^\ell)$.

(d) Add $(\boldsymbol{v}_e^\ell, \mathcal{C}[B_e^h])$ to $\mathcal{Q}$, giving higher priority to an example with greater distance, which must remain in $\mathcal{Q}$.

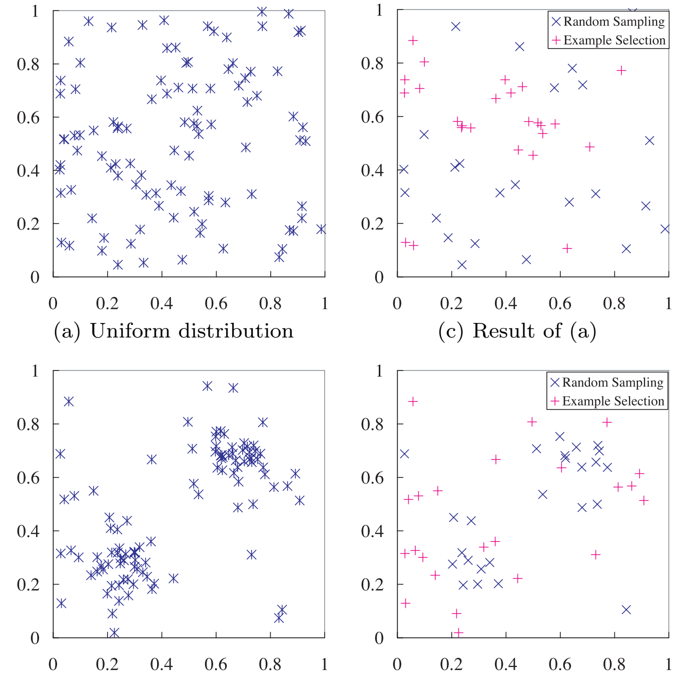
(5) Dequeue n_2 examples from $\mathcal{Q}$, and store them in $\mathcal{D}_2$.

(6) Construct $\mathcal{D} = \mathcal{D}_1 \cup \mathcal{D}_2$ by integrating examples in $\mathcal{D}_1$ and $\mathcal{D}_2$.

$\mathcal{D}_1$, which is constructed by random sampling, is a data set that approximates the distribution of examples in the feature space. Since we assume that key frames (used as training data) and super-resolved frames are temporally-closed, they are strongly correlated. Therefore, in the synthesis step of HR frames, nearest neighbors can be found within a small distance in $\mathcal{D}_1$. However, nearest neighbors may not be found within a small distance due to the motion in the scene. We introduce $\mathcal{D}_2$ to construct a data set whose elements are spread throughout the feature space. Generalization of the resulting database can eventually be enhanced.

Figure 4 illustrates an execution sample of data selection applied to 2-dimensional data. Figure 4 (a) shows 100 examples sampled from a uniform distribution, whereas (b) shows 100 examples with a biased distribution (20 examples sampled from a uniform distribution and 80 examples sampled from two normal distributions). Figure 4 (c) and (d) give the results of the selection of 50 examples from (a) and (b), respectively, using the proposed selection algorithm, where $n_1 = n_2 = 25$. Many examples are selected from densely distributed areas in the random sampling step. The subsequent step (example selection by nearest neighbor search) preferentially selects the examples from sparsely distributed areas.

Various algorithms exist for choosing representative examples from a large number of data, such as the data condensation algorithm [16], clustering and so on. However, these are all decremental methods, which first need to expand all the examples in memory before selecting the representative ones. As a result, the methods require both a large amount of memory for the data selection, and high computational cost to calculate the distances between examples. The proposed algorithm sequentially selects examples to reduce the memory requirement and computational cost. Comparative results for the proposed method and other procedures for data selection are presented in Section 4.3.



(a) Uniform distribution

(c) Result of (a)

(b) Uniform distribution + mixtured Gaussian

(d) Result of (b)

Fig. 4 Execution sample of data selection algorithm.

3.2 Synthesis of HR Frames

A LR frame I_L', which is not a key frame, is super-resolved according to the following procedure.

(1) Enlarge $I_L{}'$ using bicubic spline interpolation to generate an image $I_H^\ell{}'$ with the same size as the target HR image.

(2) Extract a block B_q^ℓ of size $K \times K$ from $I_H^\ell{}'$ and execute the following steps

for each block. Each extracted block overlaps its neighboring blocks with an overlap of width w.

(a) Calculate the contrast c_q^ℓ of the block B_q^ℓ. If $c_q^\ell < \theta_c$, SR is not carried out on this block and the values of $I_H^{\ell'}$ are used for the target HR frame. Otherwise, proceed to the next step.

(b) Conduct DCT for B_q^ℓ to obtain DCT coefficients $\mathcal{C}[B_q^\ell]$. Extract d AC coefficients of $\mathcal{C}[B_q^\ell]$ in a zigzag scan order to compose a d-dimensional vector $\boldsymbol{v}_q^\ell$.

(c) (k-nearest neighbor search) Find the k nearest neighbors $\boldsymbol{v}_1^\ell, \boldsymbol{v}_2^\ell, \cdots, \boldsymbol{v}_k^\ell$ of $\boldsymbol{v}_q^\ell$ in $\mathcal{D}$.

(d) (Local neighbor embedding) Define the Gram matrix G_q as follows:

$$G_q = (\boldsymbol{v}_q^\ell \mathbf{1}^T - L)^T (\boldsymbol{v}_q^\ell \mathbf{1}^T - L) \tag{1}$$

where $\mathbf{1}$ is a vector of ones and L is a $d \times k$ matrix with colums $\boldsymbol{v}_1^\ell, \boldsymbol{v}_2^\ell, \cdots, \boldsymbol{v}_k^\ell$.

Solve the linear system of equations $G_q \boldsymbol{w}_q = \mathbf{1}$ for $\boldsymbol{w}_q$ and then normalize the weights so that $\sum_{i=1}^k w_{qi} = 1$.

(e) Let $\mathcal{C}[B_i^h]$ $(i = 1, 2, \cdots, k)$ be the DCT coefficients of the HR blocks corresponding to $\boldsymbol{v}_i^\ell$. All DC coefficients of the k DCT blocks $\mathcal{C}[B_i^h]$ are replaced with the DC coefficient of $\mathcal{C}[B_q^\ell]$ to correct the illumination change. We denote the resulting DCT blocks as $\mathcal{C}[B_i^{h'}]$.

(f) Calculate the linear combination of $\mathcal{C}[B_i^{h'}]$ by applying $\boldsymbol{w}_q$:

$$\mathcal{C}[B_t^h] = \sum_{i=1}^k w_{qi} \mathcal{C}[B_i^{h'}]. \tag{2}$$

(g) Conduct inverse DCT for $\mathcal{C}[B_t^h]$ to transform the block to the image space.

(3) Construct the target HR frame by arranging the blocks obtained in the previous step with an overlap of width w. The pixel values in the overlapping regions are averaged to obtain a smooth image.

4. Experiments

We conducted evaluation experiments using MPEG test sequences to verify the efficiency of the proposed method. To evaluate the quality of the synthesized video sequence, we used the peak signal to noise ratio (PSNR) values between the synthesized frames and original frames. The proposed method was implemented in Visual C++ 2005 and run on a Windows XP PC (CPU: Intel Pentium4 3.0 [GHz], RAM: 512 [MB]). We used the ANN Library [17] for nearest neighbor searching. The parameters of the proposed algorithm were set as follows: $\theta_c = 8.0$, $w = 4$, $k = 2$, and $K = 8$.

Table 1 shows the MPEG test sequences used in the experiments. The two video sequences used as input were created from the original sequence as described below. A LR video sequence ($M/4 \times N/4$ [pixels], 30 [fps]) was obtained by a 25% scaling down of the original sequence ($M \times N$ [pixels], 30 [fps]). A HR video with low frame rate ($M \times N$ [pixels], 30/7 [fps]) was obtained by selecting every seventh frame from the original sequence. The proposed method synthesizes an HR video with high frame rate ($M \times N$ [pixels], 30 [fps]) from these two video sequences.

4.1 Comparison of Various Data Replacement Procedures

We compared the various data replacement procedures mentioned in Section 2. We set n = 100,000, which is large enough to store all the examples extracted from one key frame[⋆1]. To evaluate the effectiveness of the data replacement procedures, we also set $n = n_1$ and $n_2 = 0$, i.e., data selection was not carried out.

Table 1 Description of test sequences.

Sequence Name	Spatial Resolution	Frame No.
Coast guard	352 × 288	0 - 270
Football	352 × 240	1 - 121
Foreman	352 × 288	0 - 270
Hall monitor	352 × 288	0 - 270

[⋆1] The number of examples extracted from one key frame depends on the value of the threshold θ_c, and varies with the contents of each frame. In this experiment, the number of examples varies approximately between 10,000 and 40,000.

Table 2 Comparison of various data replacement procedures.

Sequence Name	Fix	Clear	Random	FC	LRU
Coast guard	22.68	24.72	24.55	22.82	23.77
Football	21.39	22.23	22.14	21.47	22.04
Foreman	28.48	29.93	29.73	28.19	29.10
Hall monitor	28.85	29.57	28.68	26.95	27.30

Table 3 Effects on dimension reduction.

	Coast guard		Football		Foreman		Hall monitor	
Dimension d	PSNR [dB]	Time [msec]	PSNR [dB]	Time [msec]	PSNR [dB]	Time [msec]	PSNR [dB]	Time [msec]
3	22.54	41	20.89	48	27.89	47	26.41	48
5	24.03	53	21.76	76	29.40	56	28.95	56
10	24.66	103	22.19	185	29.88	97	29.54	89
15	24.71	125	22.23	226	29.92	114	29.56	107
20	24.72	134	22.23	228	29.93	123	29.57	111
25	24.72	138	22.24	237	29.93	129	29.57	116

Table 4 Effects on database allocation. ($d = 20$, $n = 10{,}000$)

Size		PSNR [dB]			
n_1	n_2	Coast guard	Football	Foreman	Hall monitor
2,000	8,000	24.39	22.03	29.56	28.03
4,000	6,000	24.43	22.04	29.59	28.21
5,000	5,000	24.43	22.05	29.59	28.26
6,000	4,000	24.44	22.04	29.57	28.24
8,000	2,000	24.40	22.02	29.54	28.07
10,000	0	24.28	21.95	29.41	27.69

Table 2 gives the PSNR results for the various data replacement procedures. These results show that the data replacement procedure "Clear" gives the best results for all four test sequences. As time progresses, the scene depicted in the video sequence changes, and thus the examples stored in the database correlate less with the current frame. We should, therefore, store the examples extracted from the latest key frame, and replace all the examples in the database at every key frame. Neither the referral frequency nor the time of the most recent referral of the examples is as important in synthesizing a high quality video. Procedure "Clear" is, therefore, used in all subsequent experiments.

4.2 Effects on Dimension Reduction

Next, we show the effectiveness of dimension reduction. We measured the PSNR and average time of synthesizing one HR frame, varying d, to confirm the performance as d decreases. The input video sequences were created as mentioned in the previous section. The other experimental conditions are the same as those given in the previous section.

Table 3 gives the PSNR and time results for various dimension settings. When d is smaller than 15, the PSNR also decreases for both sequences. However, there is almost no variation in the PSNR when d is larger than 15. Therefore, approximately 15 AC coefficients contain most of the information in the extracted block. Even if we set $d \geq 20$, it takes a long time to synthesize a frame and thus we cannot expect a high PSNR.

4.3 Verification of Data Selection Procedure

4.3.1 Performance Evaluation of Database Allocation

We examined the effects on database allocation to confirm the effectiveness of the data selection algorithm. We set $d = 20$, $n = 10{,}000$ and measured PSNR, varying n_1 and n_2. **Table 4** gives the PSNR results for various database allocations. Compared with the simple strategy in which all examples stored in the database are randomly selected ($n_1 = 10{,}000$, $n_2 = 0$), the proposed method of data selection achieves better results. In particular, a higher PSNR can be obtained for most sequences when we set n_1 equal to n_2. In the subsequent experiments, we set the size of the example database to be $n_1 = n_2$.

4.3.2 Performance Evaluation of Dimension and Database Size

We evaluated the performance of generating the HR frame using the test sequence "Foreman", and varying d and/or n_1, n_2. We compared the PSNR, processing time of synthesizing one HR frame, and memory required for storing examples in the database. The processing time represents the average time of synthesizing one HR frame after the example database has been constructed. With respect to the memory requirement, we assume that one double precision real number is stored in 8 bytes. Since the pair in example $(\boldsymbol{v}^\ell, \mathcal{C}[B^h])$ is composed of $d + K^2$ real numbers, the total memory required to store n examples is $n(d + K^2) \times 8/1{,}024$ [kB]. **Table 5** gives the comparative results for different dimensions and database sizes. The row labeled "All" represents the case in which all examples are stored, and assuming $n = 100{,}000$. As d and/or $n_1 = n_2$

Table 5 Performance comparison with different vector dimensions and database sizes.

Size of	$d = 5$			$d = 10$			$d = 15$		
database $n_1 = n_2$	PSNR [dB]	Time [msec]	Memory [kB]	PSNR [dB]	Time [msec]	Memory [kB]	PSNR [dB]	Time [msec]	Memory [kB]
1,000	28.45	39	1,078	28.67	43	1,156	28.70	46	1,234
2,000	28.75	42	2,156	29.04	49	2,312	29.08	52	2,469
5,000	29.12	46	5,391	29.54	61	5,781	29.58	66	6,172
10,000	29.35	49	10,781	29.82	73	11,563	29.85	82	12,344
All	29.40	56	53,906	29.88	97	57,813	29.92	114	61,719

decrease, the PSNR also decreases. Nevertheless, a reduction in processing time and the required memory can be achieved. That is, we can control the quality of the synthesized images, computational cost and memory required by selecting appropriate values for d and n_1, n_2. **Figure 5** shows the images synthesized under different d and n conditions. Even when both dimension reduction and example selection are carried out, the synthesized images are not that bad in terms of subjective quality. Therefore, with limited memory space the proposed algorithm is able to synthesize a HR video quickly without deterioration in the quality of the synthesized images.

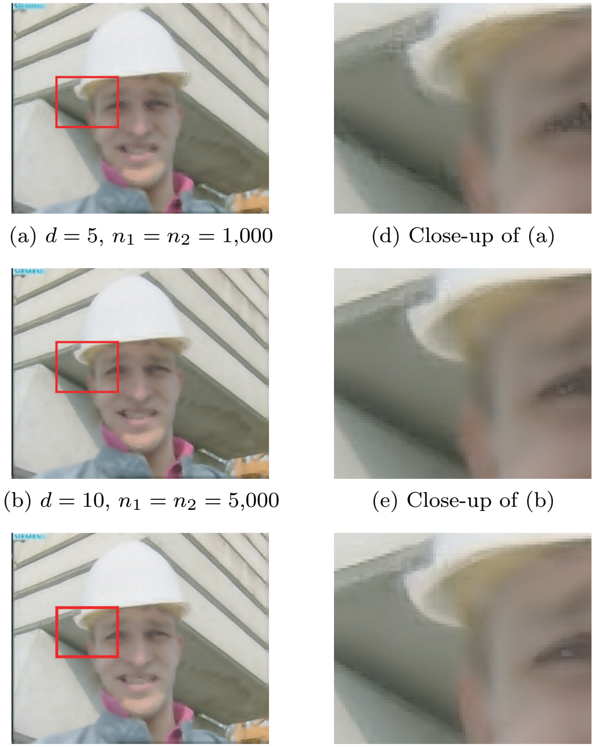
4.3.3 Comparison of Data Selection Procedure

We introduced the procedures for constructing an example database from key frames in Section 3.1. Here we show the evaluation results for several procedures for data selection (i.e., database construction). We set $d = 20$ and the size of the database $n = 10{,}000$ ($n_1 = n_2 = 5{,}000$) in this experiment. We compared the proposed algorithm with the following methods.

Random selection Select 10,000 examples randomly. This corresponds to the proposed method under the condition $n_1 = 10{,}000$ and $n_2 = 0$.

Vector quantization Initially store all the examples extracted from a key frame. Next, conduct k-means clustering, where the number of the clusters is 10,000, and then select the nearest examples for each centroid of the resulting 10,000 clusters. Selected examples are stored in the database.

HR videos with high frame rates were synthesized using these methods for data selection to construct the example database. We measured the PSNR of the synthesized videos, and the average time for constructing the example database from one key frame. The other experimental conditions remained the same as



(a) $d = 5$, $n_1 = n_2 = 1{,}000$ (d) Close-up of (a)

(b) $d = 10$, $n_1 = n_2 = 5{,}000$ (e) Close-up of (b)

(c) $d = 15$, $n_1 = 100{,}000$, $n_2 = 0$ (f) Close-up of (c)

Fig. 5 Synthesized images for various d and n values (Test sequence "Foreman" 45th frame).

Table 6 Comparison of data selection procedure.

Selection Procedure	Coast guard		Football		Foreman		Hall monitor	
	PSNR [dB]	Time [msec]	PSNR [dB]	Time [msec]	PSNR [dB]	Time [msec]	PSNR [dB]	Time [msec]
Random selection	24.30	157	21.95	157	29.41	163	27.69	166
Vector quantization	24.41	8,309	21.99	10,414	29.50	9,220	28.02	12,375
Proposed method	24.44	672	22.05	854	29.59	673	28.24	764

those in the previous section.

Table 6 gives the comparative results for the data selection procedures. The proposed method gives the highest PSNR compared with the other methods. Vector quantization takes the longest time to construct the example database. This method is computationally expensive because it requires a large number of calculations for the nearest neighbor search.

As for the memory requirement, the proposed algorithm is also superior to vector quantization. Since vector quantization is a decremental algorithm, it needs to expand all the examples in memory before selecting representative examples. In other words, it requires as much memory as shown in the bottom row of Table 5. The reduction in the memory requirement of the proposed algorithm depends on the values of d and n as mentioned in the previous section. With n = 10,000, the proposed algorithm achieves a 90% reduction in memory space compared with vector quantization. Hence, the proposed method is useful because it is able to synthesize video sequences with higher quality, and is less expensive than other methods in terms of both processing time and memory required.

4.4 Comparison with Conventional Methods

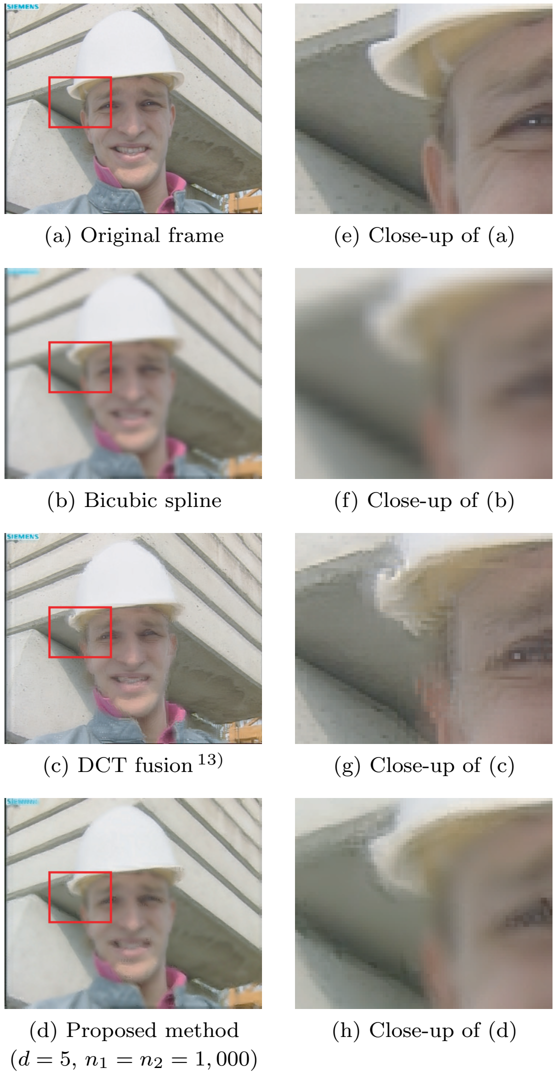
We compared the quality of the synthesized HR video using the proposed algorithm and conventional methods. We included interpolation methods (nearest neighbor and bicubic spline interpolation) and the DCT spectral fusion method [13] as conventional methods. HR videos with high frame rate were synthesized using these methods. Results of these comparative experiments are shown in terms of the quality of the synthesized videos.

Table 7 gives the PSNR results for the proposed method and the conventional ones. The conditions for the proposed algorithm are the same as those in the previous section, i.e., $d = 20$ and n = 10,000 ($n_1 = n_2$ = 5,000). PSNR values of the proposed method in Table 7 are the same as those in Table 6. The proposed method gives the highest PSNR results for the test sequences "Football" and "Foreman". These two sequences contain a large number of dynamic regions, whereas the other two sequences ("Coast guard" and "Hall monitor") are composed of simple motion. "Coast guard" contains pure translation, while "Hall monitor" is a sequence captured by a static camera. Since the DCT spectral fusion method synthesizes HR video using motion estimation, any error in the motion estimation could affect the quality of the synthesized videos. However, the proposed method synthesizes the HR video without using motion information, and thus gives the best results for the sequences with many dynamic regions. As mentioned in Section 4.3.2, the quality of the videos synthesized using the proposed method is dependent on the settings for d and n. However, the subjective quality of the synthesized video is satisfactory even when d and/or n are set relatively low.

Table 7 Comparison of synthesis algorithms.

Sequence Name	Nearest Neighbor	Bicubic Spline	Proposed Method	DCT Fusion [13]
Coast guard	22.01	22.05	24.44	24.59
Football	20.75	20.77	22.05	20.68
Foreman	26.13	26.26	29.59	27.41
Hall monitor	23.09	23.06	28.24	30.54

Figure 6 shows an original frame, (a) (e), the enlarged LR frame using bicubic spline interpolation (b) (f), and the synthesized frame using the DCT spectral fusion method [13] (c) (g). Figure 6 (d) (h) shows a synthesized frame using the proposed method, where the image has been synthesized under the conditions $d = 5$ and $n_1 = n_2$ = 1,000. Thus this image is the same as Fig. 5 (a) (d). We can see



(a) Original frame (e) Close-up of (a)

(b) Bicubic spline (f) Close-up of (b)

(c) DCT fusion [13] (g) Close-up of (c)

(d) Proposed method ($d = 5$, $n_1 = n_2 = 1,000$) (h) Close-up of (d)

Fig. 6 Comparative results (Test sequence "Foreman" 45th frame).

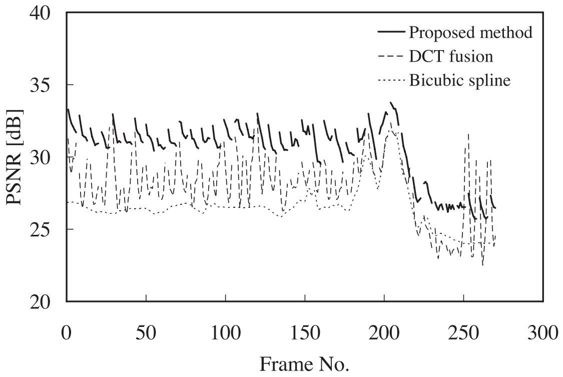


Fig. 7 PSNR variation for test sequence "Foreman".

in Fig. 6 (d) (h) that the diagonal edge in the background is sharply synthesized. However, in the central part in Fig. 6 (h) the edge is blurred, similar to the image synthesized using bicubic spline interpolation (Fig. 6 (b) (f)). The reason for this is that the central part does not have sufficient contrast to reconstruct texture (i.e., $c_q^\ell < \theta_c$), and thus SR is not carried out. In Fig. 6 (h) the texture of the wall behind the man is not synthesized for a similar reason.

Figure 7 shows the temporal variation in PSNR when comparing the proposed method and conventional methods for the test sequence "Foreman". Gaps in the line in Fig. 7 correspond to key frames. If the conventional method (DCT fusion) is used, the error in the motion estimation accumulates as the time gap between the synthesized and key frames increases. Therefore, the PSNR value varies greatly. As for the proposed method, the transition of PSNR is moderate because motion estimation is not carried out, i.e., any error in the motion estimation does not affect the results. The PSNR value for the proposed method remains high, irrespective of the time gap between the synthesized and key frames.

5. Conclusion

In this paper, we introduced several ideas to improve the efficiency of learning-based SR algorithms. To construct a compact database, we adopted a feature vector using DCT coefficients and data selection procedures. We showed through experimental results that both the memory requirement and computational cost can be reduced, while preserving the quality of the synthesized video. We also conducted comparative experiments to verify the superiority of the proposed method over conventional methods.

We assumed in this paper that the input data is obtained from a dual sensor camera. However, if we could obtain a pair of images with different resolutions by other means, our method would not necessarily require the dual sensor camera. Many of the latest digital cameras and cellular phones can capture both HR still images and LR video sequences by switching between shooting modes. For example, if the training set can be generated from still images captured under the still mode, our method can enhance the resolution of the LR video sequence. In future work we plan to adopt the proposed method by modifying the problem formulation considered in this paper.

References

1) Park, S.C., Park, M.K. and Kang, M.G.: Super-resolution image reconstruction; A technical overview, *IEEE Signal Process. Mag.*, Vol.20, No.3, pp.21–36 (2003).

2) Shechtman, E., Caspi, Y. and Irani, M.: Space-time super resolution, *IEEE Trans. PAMI*, Vol.27, No.4, pp.531–545 (2005).

3) Baker, S. and Kanade, T.: Limits on super-resolution and how to break them, *IEEE Trans. PAMI*, Vol.24, No.9, pp.1167–1183 (2002).

4) Freeman, W.T., Pasztor, E.C. and Carmichael, O.T.: Learning low-level vision, *IJCV*, Vol.40, No.1, pp.25–47 (2000).

5) Sun, J., Zheng, N.-N., Tao, H. and Shum, H.-Y.: Image hallucination with primal sketch priors, *Proc. CVPR*, pp.729–736 (2003).

6) Chang, H., Yeung, D.-Y. and Xiong, Y.: Super-resolution through neighbor embedding, *Proc. CVPR*, pp.275–282 (2004).

7) Jiji, C.V. and Chaudhuri, S.: Single-frame image super-resolution through contourlet learning, *EURASIP J. on Applied Signal Processing*, Vol.2006, pp.1–11 (2006).

8) Bishop, C.M., Blake, A. and Marthi, B.: Super-resolution enhancement of video, *Proc. 9th Intl. Conf. Artificial Intell. & Statistics* (2003).

9) Kong, D., Han, M., Xu, W., Tao, H. and Gong, Y.: A conditional random field model for video super-resolution, *Proc. ICPR*, pp.619–622 (2006).

10) Nagahara, H., Hoshikawa, A., Shigemoto, T., Iwai, Y., Yachida, M. and Tanaka, H.: Dual-sensor camera for acquiring image sequences with different spatio-temporal resolution, *Proc. IEEE Int. Conf. Advanced Video and Signal based Surveillance*, pp.450–455 (2005).

11) Matsunobu, T., Nagahara, H., Iwai, Y., Yachida, M. and Tanaka, H.: Generation of high resolution video using morphing, *Proc. SICE Annual Conf.*, pp.2101–2108 (2005).

12) Watanabe, K., Iwai, Y., Nagahara, H., Yachida, M. and Suzuki, T.: Video synthesis with high spatio-temporal resolution using motion compensation and image fusion in wavelet domain, *Proc. ACCV*, pp.480–489 (2006).

13) Watanabe, K., Iwai, Y., Nagahara, H., Yachida, M. and Suzuki, T.: Video synthesis with high spatio-temporal resolution using motion compensation and spectral fusion, *IEICE Trans. Inf. & Syst.*, Vol.E89-D, No.7, pp.2186–2196 (2006).

14) Nene, S.A. and Nayer, S.K.: A simple algorithm for nearest neighbor search in high dimensions, *IEEE Trans. PAMI*, Vol.19, No.9, pp.989–1003 (1997).

15) Arya, S., Mount, D.M., Netanyahu, N.S., Silverman, R. and Wu, A.Y.: An optimal algorithm for approximate nearest neighbor searching in fixed dimensions, *J. ACM*, Vol.45, No.6, pp.891–923 (1998).

16) Mitra, P., Murthy, C.A. and Pal, S.K.: Density-based multiscale data condensation, *IEEE Trans. PAMI*, Vol.24, No.6, pp.734–747 (2002).

17) Mount, D.M. and Arya, S.: *ANN: A library for approximate nearest neighbor searching*, http://www.cs.umd.edu/˜mount/ANN/.

(Received September 29, 2008)
(Accepted October 13, 2009)
(Released December 14, 2009)

(Communicated by *Yi-Ping Hung*)

Kiyotaka Watanabe graduated from Osaka University in 2004, and received his M.E. and Ph.D. degrees from the same university in 2006 and 2009, respectively. Since 2006, he has been working at the Advanced Technology R&D Center, Mitsubishi Electric Corporation. Currently he is engaged in developing video surveillance systems. His research interests are in the fields of image processing and computer vision. He is a member of the Institute of Image Information and Television Engineers (ITE).

Yoshio Iwai graduated from Osaka University in 1992 and completed the M.S. and Doctoral programs in 1994 and 1997, respectively. He was then appointed a Research Associate at the same university, later becoming an Associate Professor. Between 2004 and 2005, he was a visiting researcher at Cambridge University. He is currently engaged in studies relating to computer vision and pattern recognition. He is a member of IEEE, IPSJ, and RSJ. He also has a D.Eng. degree.

Tetsuji Haga graduated from the Department of Control Engineering at Osaka University in 1989. In 1991, he completed the Masters program in the Graduate School of Engineering Science, Osaka University. In the same year, he joined Mitsubishi Electric Corp., initially in the Industrial Systems Research Group. Since 2002, he has been a member of the Advanced Technology R&D Center involved in research into image processing systems for security surveillance. He completed the Ph.D. program in the Graduate School of Information Science and Technology, Osaka University in 2006. He is a member of IEICE, IPSJ, IIEEJ, and IEEJ.

Koichi Takeuchi received his B.E. and M.E. degrees in Applied Physics from Osaka University in 1985 and 1987, respectively. He joined Mitsubishi Electric Corp. in 1987, and has been a member of the Advanced Technology R&D Center since 2003. His fields of research at Mitsubishi Electric are optical disc systems and IP-broadcasting systems. He is a member of the Institute of Image Information and Television Engineers (ITE).

Masahiko Yachida received his B.E. and M.Sc in Electrical Engineering, and a Ph.D. in Control Engineering, all from Osaka University in 1969, 1971, and 1976, respectively. He joined the Department of Control Engineering, Faculty of Engineering Science, Osaka University in 1971 as a Research Associate and later became an Associate Professor in the same department. He then moved to the Department of Information and Computer Science at the same university as a Professor in 1990. From 1993, he was a Professor in the Department of Systems Engineering at the same university. From 1997, he was a Professor of Systems and Human Science, Graduate School of Engineering Science, Osaka University. Since 2008 he has been a Professor in the Faculty of Information Science and Technology, Osaka Institute of Technology. He is the author of Robot Vision (Shoukoudou) which received the Ohkawa Publishing Prize, a co-author of Pattern Information Processing (Ohm-sha), and an editor of Computer Vision (Maruzen) and other books. He has been the Chairman of the Technical Committee on Computer Vision & Image Media, IPSJ and the Chairman of the Technical Committee on Pattern Recognition & Media Understanding, IEICE, Japan. His research interests are in the fields of computer vision, image processing, mobile robots and artificial intelligence.