Image Data Interpolation Based on Adaptive Triangulation

Huan-Chun Xu* ***Associate Member***, Jung-Sik Lee*, Jae-Jeong Hwang* ***Regular Members***

ABSTRACT

This paper proposes a regional feature preserving adaptive interpolation algorithm for natural images. The algorithm can be used in resolution enhancement, arbitrary rotation and other applications of still images. The basic idea is to first scan the sample image to initialize a 2D array which records the edge direction of all four-pixel squares, and then use the array to adapt the interpolation at a higher resolution based on the edge structures. A hybrid approach of switching between bilinear and triangulation-based interpolation is proposed to reduce the overall computational complexity. The experiments demonstrate our adaptive interpolation and show higher PSNR results of about max 2 dB than other traditional interpolation algorithms.

Key Words : Triangulated interpolation, Bilinear interpolation, Edge-directed interpolation

Ⅰ. Introduction

Image interpolation has wide range of application areas in many image processing systems. For digital images, interpolation is necessary to increase the sampling rate of an image signal. It is also required in any geometric transformation or warping of images.

A good interpolation algorithm should be able to recover the clear edge and suppressing the artifacts around edges. And it should be computationally efficient both in time and in memory.

Various algorithms for image interpolation have been proposed in the literature. Traditional techniques, such as pixel replication, bilinear or bicubic[1] interpolation, are based on space-invariant models. A lot of algorithms have been proposed to improve the magnification results. Iterative methods such as PDE-based schemes[2-3] and projection onto convex sets (POCS) schemes[4-5] constrain the edge continuity and find the appropriate solution through iterations. Orthogonal transform methods focus on the use of the discrete cosine transform (DCT)[6-7].

Edge-directed interpolation techniques[8-12] employ a source model and modify the interpolation to fit the source model. Other approaches[13] borrow the techniques from vector quantization and morphological filtering to facilitate the induction of high-resolution images.

Another approach is triangulation modeling. Yu et al.[14] modeled images as data dependent triangulation meshes and reconstructed images from the triangulation mesh. Their approach adapted traditional data-dependent triangulation[15] with their new cost functions and optimizations. Dan Su[16] developed a new edge-directed method for image interpolation. It gets simpler and faster than Yu's algorithm because it does not involve any cost function or repeating optimization process.

In the Dan Su's algorithm, however, only diagonal directions are used to triangulate the four-pixel square. The problem is that it cannot correspond to edges of arbitrary angle. If we try to use sub-pixel triangulation to represent arbitrary angles, it will add more complexity to the algorithm. So Dan Su's algorithm cannot achieve accurate interpolated values, if the edge is roughly extended over either horizontal or vertical direction, i.e., the triangle crosses the edge. Thus,

* School of Electronics & Information Eng., Kunsan National University ({leejs, hwang}@kunsan.ac.kr)
논문번호 : KICS2007-02-055, 접수일자 : 2007년 2월 11일, 최종논문접수일자 : 2007년 8월 1일

we need to develop more correct classification algorithm to formulate or triangulate the edge structure.

The rest of this paper is organized as follows: Section 2 presents the traditional linear interpolation algorithms and triangulation interpolation. In Section 3, our proposed algorithm is discussed in three parts; first, definition of the classification of edges and description of their characteristics, second, adaptive interpolation based on data-dependent algorithm, and last, computational analysis of our algorithms. Simulation results are reported in Section 4 and some concluding remarks are made in Section 5.

Ⅱ. Interpolation of Image Data

2.1 Nearest neighbor interpolation

Nearest neighbour is the most basic and requires the least processing time of all the interpolation algorithms because it only considers one pixel－the closest one to interpolated point. This has the effect of simply making each pixel bigger.

2.2 Bilinear interpolation

Bilinear interpolation considers the closest 2*2 neighborhood of known pixel values surrounding the unknown pixel. It then takes a weighted average of these 4 pixels to arrive at its final interpolated value. This result is much smoother looking images than nearest neighbor.

It requires three linear interpolations. The four nearest pixels are northwest, northeast, southwest, and southeast.

The operations are shown as follows:

$$F(x,y) \approx F(),0)(1-x)(1-y)+F(1,0)x(1-y) + F(0,1)(1-x)y+F(1,1)xy \quad (1)$$

Actually in our case, the distance between each two near pixels is equal so we can get the more compact form (Eq.2):

$$F(x,y)=0.25^*]F(0,0)+F(1,0)+F(0,1)+F(1,1)] \quad (2)$$

2.3 Bi-cubic algorithm

Bi-cubic goes one step beyond bilinear by considering the closest 4*4 neighborhood of known pixels－for total of 16 pixels. Since these are at various distances from the unknown pixel, closer pixels are given a higher weighting in the calculation. Bi-cubic produces noticeably sharper images than the previous two methods. It is a standard in many image editing programs, printer drivers and in-camera interpolation. The bi-cubic interpolation is calculated as Eq.3:

$$\sum_{i=0}^{3}\sum_{j=0}^{3} a_{ij}x^i y^j \quad (3)$$

The procedure used to calculate the coefficients a_{ij} depends on the interpolated data source properties.

2.4 Triangulation

This algorithm fits the finest triangular mesh to the source pixels. The “image mesh” is completely regular except that the diagonals are locally selected to run in the same general direction as any visible edge. To generate a new image, possibly at higher resolution, the target pixels are located in the source mesh. We can evaluate each target pixel from the triangle in which it sits. It is interpolated using only the information from the three triangle vertices. For this reason the high-resolution image can has the edges sharp and the smooth areas smooth.

Ⅲ. New Data-Dependent Interpolation

3.1 Classification of edge

It has been recognized that taking edge information into account will improve the interpolated images’ quality and it is known that the human visual system makes significant use of edge. If the detail is trivialness the complexity of edge is higher. The more facts which can be influence the image quality should be considered. Always the obvious and large edge can be recorded in a sampling, so the analysis of the sampling pixel can give us more edge information.

In our algorithm we only focus on a small area of image X (only include 2*2 pixels), we can use this small group to get 3*3 pixels in image Y which include 4 sampling pixels at the corner, 1 middle pixel in the middle of group and 4 side pixels between the each two sampling pixels. First we need to classify the different edges in the small area. We can get judgment intuitively like follows (Fig. 1):

Flat area: In fact, each image must contain a lot of flat areas such as background or big object. In these areas the intensity of the image is located in a small range of values. We can compare the difference between the maximum and minimum pixel values and the threshold which we set at first. If the difference is lower than threshold we assign the group is a flat area.

Horizontal or vertical edge: That means a boundary combined by a set of connected pixels that are located at an orthogonal step transition in gray level.

Diagonal edge: We consider the case that there is an edge passing between a square of four pixels. If this edge cuts off one corner, one pixel will have a value substantially different to the other three. We call this pixel the outlier and we can get a triangle which is consisted of other three pixels.

Crossover case: It is the situation that both of diagonal position pixels are bigger or smaller than others.

In our case, we designed a filter to classify the down-sampling pixels. Firstly we can find the maximum and minimum pixels and record their position. Secondly we calculate the gap between the max and min pixels. If it is lower than the threshold which we set before, we call it flat area. Otherwise, we get the half value between maximum and minimum pixels, compare the other two pixels, higher or lower than half value, and record their position. Then we accord the records to classify each group.

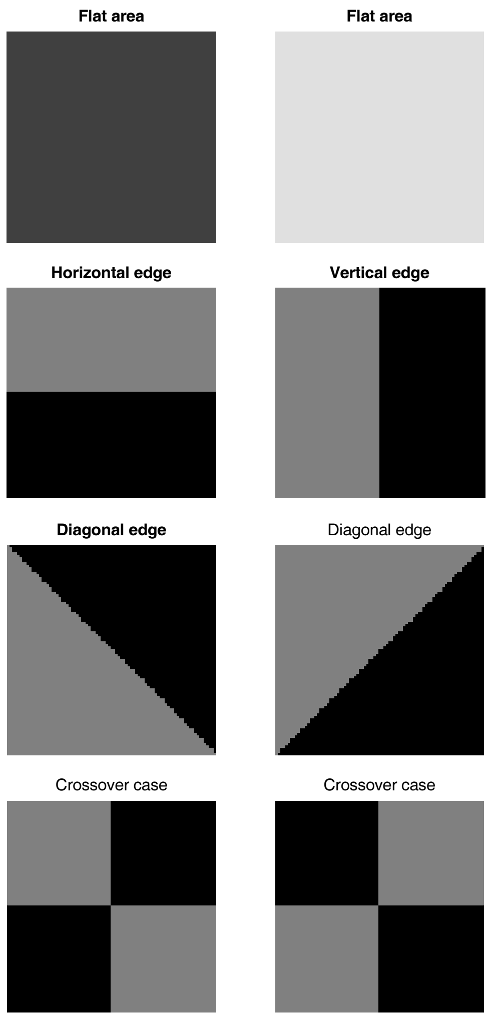

Fig. 1. Structural edges in a 2*2 region

3.2 Middle point interpolation

Now we have recorded the each group edge's information, according it, it is easy to interpolate the middle pixel by bilinear algorithm in flat area. If the edge is roughly either horizontal or vertical it is similar to bilinear interpolation in theses cases.

If the diagonal is to correspond to the edge in the image, the diagonal should be the one which does not connect to the outlying pixel value, the one most different to the other three. We can use the triangulate interpolation to get the middle pixel.

The crossover is more complex because it always denotes that the edge is too narrow tobe detected. So we think about the situations near this group to identify the edge's type. Always the crossover case is denoted as a diagonal edge, like a bridge astride a river. So we only need to find which two pixels content the real edge and use linear interpolation to get a middle pixel.

Edge structure and reference pixels are illustrated in Fig. 2.

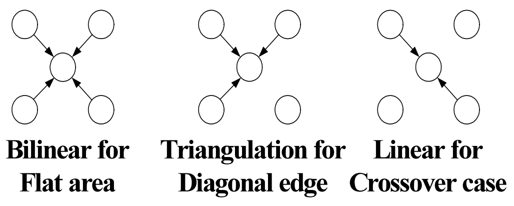

Fig. 2. Reference pixels for interpolation.

3.3 Side point interpolation

In horizontal and vertical case, it is easy to interpolate the two side pixels which have the same direction as edge type. And other pixel we can use bilinear or triangulation to interpolate it.

The diagonal case, for the side pixel falling in a flat triangle we use the three pixels which are near this pixel to triangulate it. For the side pixel falling out of a flat triangle we can take a think about adjacent group's situation. If still can't be completed accord the preceding principle, we use the linear interpolation base on the nearest two sampling pixel.

The crossover is as similar as diagonal case, we need think about the others groups' situation and adopt the different methods to get it.

3.4 Complexity analysis

Now we analyses the complexity of our algorithm. Suppose the image I has width and height m, so the number of pixels is $n = m^2$. The number of possible edge types is $(m-1)\times(m-1)$. In our implementation, we use a table to record the orientation of the edge type in each square. There are seven directions for each square; we use 3 bits to store this information. The total memory requirement for our algorithm is $3\times(m-1)^2 \approx 3n$. The memory requirement $3n$ is linear with the number of pixels n.

In our method, each classification requires sorting process in a square which need at most six comparisons. Then we calculate the gap between them to compare the threshold that we can set firstly. If the edge type is flat, we only add one subtraction and one comparison. The computation of that is $(m-1)\times(m-1)\times 8 \approx 8n$. In other cases, it will add two other comparisons. The computation of that is almost $(m-1)\times(m-1)\times 10 \approx 10n$. It is still linear with image size n.

The next interpolation step is to interpolate unknown pixels. We can use the linear and triangulate interpolation. The computation of triangulation is simpler than bi-cubic and bilinear. The triangulation only needs two additions and one division. It is easy to see that triangle interpolation has the same complexity as linear interpolation which is complexity with n. totally, our algorithm have a time complexity of $O(n)$.

Our algorithm is efficient in both memory and time, and is suitable for handling large images with a linear dependency on the image size.

Ⅳ. Simulation and Results

As mentioned in the introduction, we implemented several interpolation methods. The image from bilinear interpolation and bi-cubic interpolation were produced from Matlab 6 built-in functions. We also used the Matlab and our own graphics library to implement our methods.

Grayscale imageswere processed exactly as already described. We have used photographic images Lenna as our benchmark images. The new adaptive interpolation is compared with two conventional linear interpolation methods: bilinear and bi-cubic. The low-resolution image is obtained by direct downsampling the original image by a factor of two along each dimension (Fig. 3).

Table 1. Comparison of Interpolation performance.

Image / Method	Lenna (dB)	Peppers (dB)	Rose (dB)	Alpha (dB)
Bilinear	26.940	29.010	37.118	17.896
Bi-cubic	26.941	28.916	37.243	18.227
Triangulation	25.106	26.242	34.975	17.432
Adaptive interpolation	27.179	29.231	37.373	18.446

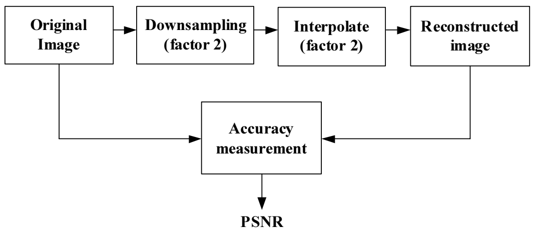

Fig. 3. Simulation flow diagram to test the adaptive interpolation.

Performance comparison is measured by the objective assessment PSNR between original and interpolated images. The original images are shown in Fig. 4.

The reconstructed image 'Lenna' is shown in Fig. 5. The detail of image 'Lenna' is considered in Fig. 6. By inspection of the reconstructed images it may be seen that the edges are reconstructed more sharply by our operator.

We have compared the image 'Peppers', the results are shown in Fig. 7. Interpolation results for 'Peppers' image by a) Bilinear interpolation, b) Bi-cubic interpolation, c) Triangulation interpolation and d) Proposed adaptive interpolation.

The reconstruction of image 'Rose' is compared in Fig. 8. Interpolation results for 'Rose' image by a) Bilinear interpolation, b) Bi-cubic interpolation, c) Triangulation interpolation and d) Proposed adaptive interpolation. The reconstruction of image 'A alphabet' is shown in Fig. 9. Interpolation results for 'A alphabet' image by a) Bilinear interpolation, b) Bi-cubic interpolation, c) Triangulation interpolation and d) Proposed adaptive interpolation.

Also an objective measure of the interpolation shows the good performances of the proposed approach for our interpolator, the bilinear, the bicubic and triangulation. We have tested four images to derive the PSNR results as shown in Table 1.

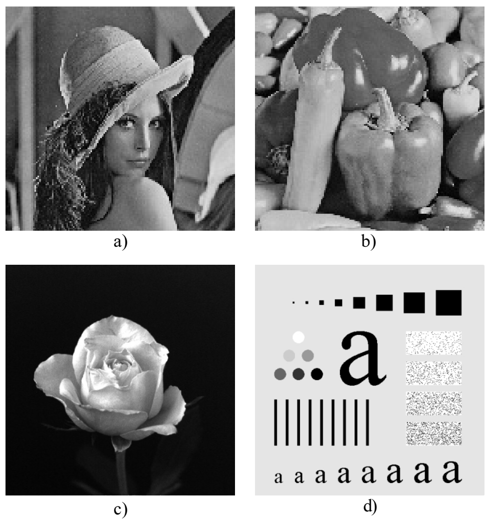

Fig. 4. Original images. a) Lenna. b) Peppers. c) Rose. d) A alphabet.

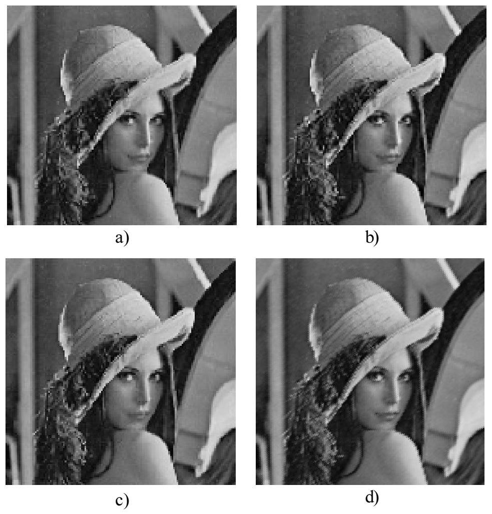

Fig. 5. Interpolation results for 'Lenna' image by a) Bilinear interpolation, b) Bi-cubic interpolation, c) Triangulation interpolation and d) Proposed adaptive interpolation.

V. Conclusions

In this paper, we present an adaptive interpolation algorithm based on data-dependent. The interpolation is adapted by the edge detection in a small group and the character of each edge type.

The linear and triangulation is proposed to reduce the computational complexity. In the simulation, the results show us new adaptive interpolation demonstrates significant improvement over traditional interpolation and triangulation interpolation on visual quality of the interpolated image.

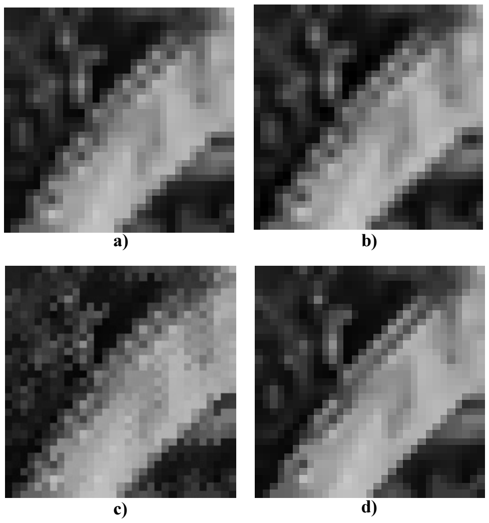

Fig. 6. Enlarged parts of the reconstructed ‘Lenna’ image. a) Bilinear interpolation. b) Bi-cubic interpolation. c) Triangulation interpolation. d) Adaptive interpolation.

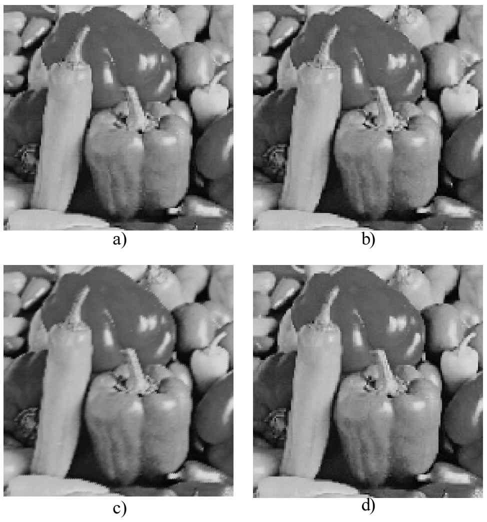

Fig. 7. Interpolation results for ‘Peppers’ image by a) Bilinear interpolation, b) Bi-cubic interpolation, c) Triangulation interpolation and d) Proposed adaptive interpolation.

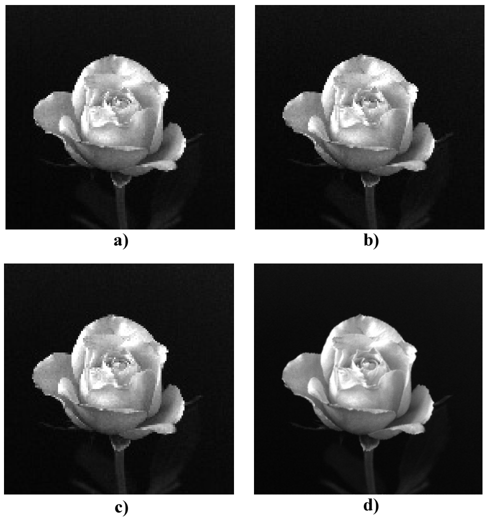

Fig. 8. Interpolation results for ‘Rose’ image by a) Bilinear interpolation, b) Bi-cubic interpolation, c) Triangulation interpolation and d) Proposed adaptive interpolation.

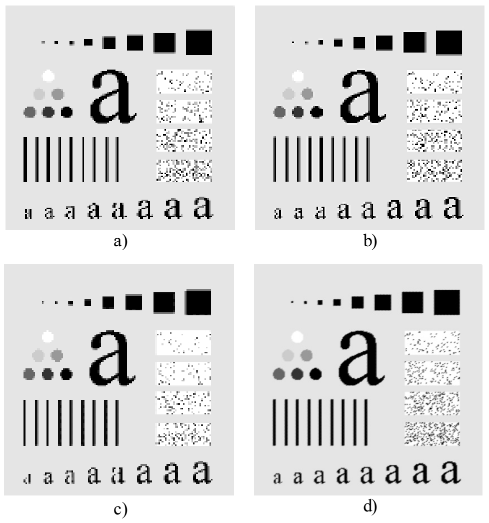

Fig. 9. Interpolation results for ‘A alphabet’ image by a) Bilinear interpolation, b) Bi-cubic interpolation, c) Triangulation interpolation and d) Proposed adaptive interpolation.

References

〔1〕 R. Keys, “Cubic Convolution Interpolation for Digital Image Processing,” IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 29, no. 6, pp. 1153-1160, June 1981.

〔2〕 B. Ayazifar and J. S. Lim, “Pel-adaptive model-based interpolation of spatially subsampled

images," in Proc. IEEE Int. Conf. Accoustics, Speech, Signal Processing, vol. 3, pp. 181-184, 1992.
〔3〕 B. S. Morse and D. Schwartzwald, "Isophote-based interpolation," in Proc. IEEE Int. Conf. Image Processing, vol. 3, pp.227-231, 1998.
〔4〕 K. Ratakonda and N. Ahuja, "POCS based adaptive image magnification," in Proc. IEEE Int. Conf. Image Processing, vol. 3, pp. 203-207, 1998.
〔5〕 D. Calle and A. Montanvert, "Superresolution inducing of an image," in Proc. IEEE Int. Conf. Image Processing, vol. 3, pp. 232-235, 1998.
〔6〕 S. A. Martucci, "Image Resizing in the Discrete Cosine Transform Domain," in Proc. Int. Conf. Image Processing, vol. 2, pp. 244-247, 1995.
〔7〕 E. Shinbori and M. Takagi, "High Quality Image Magnification Applying the Gerchberg-Papoulis Iterative Algorithm with DCT," Systems and Computers in Japan, vol. 25, No. 6, pp. 80-90, 1994.
〔8〕 S. Carrato, G. Ramponi and S. Marsi, "A Simple Edge-Sensitive Image Interpolation Filter," Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing, vol. 3 , pp. 711-714, 1996.
〔9〕 K. Jensen and D. Anastassiou, "Subpixel edge localization and theinterpolation of still images," IEEE Trans. on Image Processing, vol. 4, pp. 285-295, Mar. 1995.
〔10〕 J. Allebach and P. W. Wong, "Edge-directed in-terpolation," in Proc. IEEE Int. Conf. Image Processing, vol. 3, pp.707-710, 1996.
〔11〕 X. Li and M. T. Orchard, "New Edge-Directed Interpolation," in Proc. IEEE Int. Conf. Image Processing, vol. 2, pp. 311-314, 2000.
〔12〕 S. Battiato, G. Gallo, F. Stanco, "A lo-cally-adaptive zooming algorithm for digital im-ages,"Image and Vision computing, vol. 20, no. 11, pp. 805-812, Sep. 2002.
〔13〕 D. A. Floencio and R. W. Schafer, "Post-sampling aliasing control for natural images,"in Proc. IEEE Int. Conf. Acoustics, Speddch, Signal Processing, vol. 2, pp. 893-896, 1995.
〔14〕 X. Yu, B. Morse, T. W. Sederberg, "Image Reconstruction Using Data-Dependent Triangulation", IEEE Computer Graphics and Applications, vol. 21, no. 3, pp. 62-68, 2001.
〔15〕 N. Dyn, D. Levin, S. Rippa, "Data Dependent Triangulations for Piecewise Linear Interpolation," IMA Journal of Numerical Analysis, Institute of Mathematics and its Applications, vol.10, pp. 127-154, 1990.
〔16〕 D. Su and P. Willis, "Image Interpolation by Pixel Level Data-Dependent Triangulation" in Computer Graphics Forum, pp. 1-13, Jan. 2004.

Huan-Chun Xu 준회원

2005년: 천진 재경대학교 졸업
2007년: 군산대학교 전자정보공학부 공학석사
<관심분야> 영상처리, 영상복원

이 정 식 (Jung-Sik Lee) 정회원

1983년: 한양대학교 전자통신과 공학사
1990년: Florida Institute of Tech., Elec. Eng. M.S.
1996년: Florida Institute of Tech., Elec. Eng. Ph.D.
2004년-2005년: Univ. Of South Florida 방문 교수
1997년-현재: 군산대 전자정보공학부 부교수
<관심분야> 적응신호처리, 통신이론, 신경망

황 재 정 (Jae-Jeong Hwang) 정회원

1992년, 전북대학교 대학원 전자공학과 공학박사
1993년, 미국 텍사스주립대 객원교수
1992년 - 현재, 군산대학교 전자정보공학부 교수
2003년, 호주 모나쉬대학교 객원교수
저서: Techniques and standards for image, video, and audio coding, Prentice Hall, 1996.