Huang H, Zang Y, Rosin PL *et al.* Edge-aware level set diffusion and bilateral filtering reconstruction for image magnification. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 24(4): 734–744 July 2009

Edge-Aware Level Set Diffusion and Bilateral Filtering Reconstruction for Image Magnification

Hua Huang[1] (黄　华), *Senior Member, CCF, Member, IEEE*, Yu Zang[1] (臧　彧), Paul L. Rosin[2] and Chun Qi[1] (齐　春), *Senior Member, CCF*

[1] *School of Electronics and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China*

[2] *School of Computer Science, Cardiff University, Cardiff, U.K.*

E-mail: {huanghua, qichun}@xjtu.edu.cn; zangyu7@126.com; Paul.Rosin@cs.cf.ac.uk

Received July 5, 2008; revised April 9, 2009.

Abstract In this paper we propose an image magnification reconstruction method. In recent years many interpolation algorithms have been proposed for image magnification, but all of them have defects to some degree, such as *jaggies* and blurring. To solve these problems, we propose applying post-processing which consists of edge-aware *level set diffusion* and *bilateral filtering.* After the initial interpolation, the contours of the image are identified. Next, edge-aware level set diffusion is applied to these significant contours to remove the jaggies, followed by bilateral filtering at the same locations to reduce the blurring created by the initial interpolation and level set diffusion. These processes produce sharp contours without jaggies and preserve the details of the image. Results show that the overall RMS error of our method barely increases while the contour smoothness and sharpness are substantially improved.

Keywords computer application, image magnification reconstruction, edge-aware level set diffusion, bilateral filtering

1 Introduction

The goal of image magnification is to rescale a low resolution image to a high resolution image in order to improve its visual appearance. Image magnification is used as part of image enhancement. In many cases, although high resolution images are desirable, due to various hardware or cost limitations they are not available. So image magnification is widely applied in many areas such as satellite remote sensing[1], medical image analysis[2] and forensic analysis[3]. In recent years much research has also gone into attempting to improve the quality of image magnification[4,5].

Conventional function-based interpolation methods treat the image as a low-dimensional surface, then find a function to approximate this surface, usually bilinear or bicubic. This approach can produce good results in the low-frequency part of the image but may introduce some errors in the high-frequency contents, and consequently will often show artifacts at the image edges. These artifacts are aligned to the original low resolution pixels and prevent the image contours being as smooth as they should be, so called *jaggies.* They appear as a staircase pattern, and are most visible along high contrast diagonal edges. Although a variety of filtering approaches exists to minimize this visual artifact, but none has been applied selectively only to those edges that would otherwise appear jagged[6]. Moreover, in addition to the problem of spurious high-frequencies, interpolation errors can also result in loss of high-frequency information, leading to blurring.

To solve these problems, we propose an edge-aware level set diffusion and bilateral filtering-based image magnification reconstruction method. The first step is to perform standard interpolation, we use the bicubic method. Next, the significant contours are identified. Then level set diffusion is applied to these significant contours to remove the jaggies, followed by bilateral filtering at the same location to reduce the blur problem caused by the initial interpolation and level set diffusion. The results show that our method can produce better visual effects while the overall RMS error barely increases.

This paper is organized as follows. In Section 2 we describe some related work in this area. In the first part of Section 3 we describe how to extract the contours of the image, in the second part we use it to direct level set diffusion, followed in the third part by an explanation of how to use bilateral filtering to get sharp contours, then in the fourth part we perform experiments to fix the thresholds. In Section 4 we will show our results and compare them with some other methods. The

Regular Paper

This work is supported by the National Natural Science Foundation of China under Grant Nos. 60703003 and 60641002.

conclusions are given in Section 5.

2 Related Work

Since conventional interpolation methods have various defects, many kinds of methods had been proposed in recent years to improve the quality of the image magnification, each with its own distinctive methodology and limitation. In this section, we discuss some typical techniques by dividing them into groups of common features.

Reconstruction-based methods try to modify the image after initial interpolation to improve the visual effect. Back-projection[5] minimizes the error by an iterative procedure. In each step, the current high resolution image is back-projected to the original image resolution to determine the reconstruction error, and the error image is used to correct the current high resolution image. However, since edge information is ignored, this process is prone to artifacts such as jaggies and ringing effect. This problem was corrected by [7] in which bilateral back-projection is proposed. They first extracted the salient image edges, and applied a soft edge smoothness prior to synthesis a high resolution image with smooth and sharp edges. Next, a bilateral back-projection-based post-processing step is employed. Bilateral filtering was also applied to the error image, guided by the edge map. Such an approach is sensitive to the quality of the edge detection stage, and the iterative procedure is costly in time. After an initial interpolation, [4] proposes the level set reconstruction to smooth the contour and remove the jaggies, but this may destroy some image details, and moreover does not eliminate the blurring introduced by the initial interpolation.

Edge-detection-based methods use the edge information to control interpolation. The edge-aware concept was first proposed in [8], which fits smooth subpixel edges to the image and prevents interpolation across the edges, but they did not specify how to identify relevant edges. [9] proposed a method to improve the subjective quality of the interpolated image by tuning the interpolation coefficients to match an arbitrarily-oriented step edge. But since the interpolation weights are estimated from flat regions as well, edges are still blurred. [10] gives an edge-based image decomposition method which can produce resolution independent vectorization results, but it mainly focuses on cartoon images.

Learning-based methods[11,12] have been widely developed in recent years. They use a set of training images to direct image magnification and can produce good results, but the common weak point of these methods is that they strongly rely on the set of training images, so they cannot guarantee a good result for all new images.

Some other methods have been used in image magnification. For example, by using 8 extra bits to store the location of some important boundaries, images can be enlarged while keeping the edges infinitely sharp[13]. But this extra information is obtained from the original high resolution (HR) image, and in image magnification we usually do not have the HR image, so this limits its applicability.

Content aware image and video resizing[14–16] can resize the video including magnification. However, these methods produce fake but nice looking results by using the idea of repetition of unimportant areas.

3 Methods

Our work is based on the assumptions that:

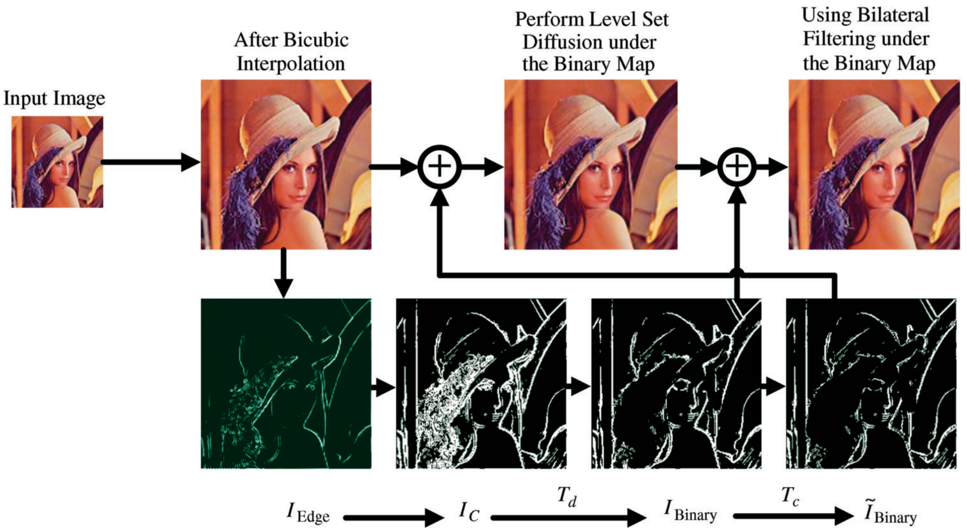

Input Image
After Bicubic Interpolation
Perform Level Set Diffusion under the Binary Map
Using Bilateral Filtering under the Binary Map
I_{Edge}
I_C
T_d
I_{Binary}
T_c
$\tilde{I}_{\mathrm{Binary}}$

Fig.1. Framework overview. The two system parameters are denoted as T_d and T_c, the remaining steps are of all automatic processes.

1) the human visual system is sensitive to edges, so improving the quality of edges can make a significant improvement on the perceived quality of the resulting image;

2) the initial interpolation errors are mainly at the significant edge contours of the image, so we just need to do the reconstruction at these places.

Considering the defects of the initial interpolation and the assumption above, we can summarize our approach as follows:

1) create an initial interpolation image using standard methods;

2) from the interpolated image, extract the edges and separate them into areas of textural detail and larger scale significant contours;

3) use level set diffusion to smooth the contours to remove the jaggies;

4) use bilateral filtering at the contours to reduce the blurring introduced by interpolation and the level set smoothing to get sharp edges.

Fig.1 shows an overview of our framework.

3.1 Edge Extraction

Denote the input image after bicubic interpolation as I_H. To extract the initial edges we first compute approximations to the image derivatives (see Fig.2(a)) using the central difference scheme which can be written as:

$$\frac{\partial I_H}{\partial x} = (I(x-h, y) - I(x+h, y))/2h, \tag{1}$$

$$\frac{\partial I_H}{\partial y} = (I(x, y-h) - I(x, y+h))/2h, \tag{2}$$

where h is the spatial step size, here it is set to 1. Edge magnitudes are then calculated as

$$I_{\mathrm{Edge}} = \frac{255}{\max(I_{\mathrm{Edge}})}\sqrt{\left(\frac{\partial I_H}{\partial x}\right)^2 + \left(\frac{\partial I_H}{\partial y}\right)^2}, \tag{3}$$

where $255/\max(I_{\mathrm{Edge}})$ is the scaling coefficient.

Other more sophisticated edge detection algorithms, such as Canny operator are available. However, Canny's non-maximal suppression step for thinning the edges tends to produce sets of edges that do not cover all the jaggies. Therefore we prefer to use the above simple edge detector, which also has the advantage of computational efficiency.

Having computed edge magnitudes the edge map needs to be binarised. This is best performed automatically so that the threshold is dynamically adapted to the image content. We use the moment-preserving method for threshold selection automatic[17]. This scheme selects an optimal threshold such that the binarised image preserves the first three moments of the input image.

The set of edges produced after thresholding is too sparse for our purpose. In particular, ignoring spatial information during thresholding means that edge contours can be fragmented due to local fluctuations in edge magnitude, even if they are quite minor. Therefore Canny's hysteresis edge thresholding method is applied to overcome this problem. The moment-preserving thresholding is first used to provide an upper threshold T_{high}. Next a lower threshold T_{low} is determined; we follow common practise and set $T_{\mathrm{low}} = T_{\mathrm{high}}/2$. Edges are now thresholded via the following steps to produce the binary edge map I_C:

1) if $I_{\mathrm{Edge}}(x, y) > T_{\mathrm{high}}$, then $I_C(x, y) = \text{white}$;

2) if $I_{\mathrm{Edge}}(x, y) < T_{\mathrm{low}}$, then $I_C(x, y) = \text{black}$;

3) if $T_{\mathrm{low}} \leqslant I_{\mathrm{Edge}}(x, y) < T_{\mathrm{high}}$ and $I_C(x_i, y_i) = \text{white}$, where (x_i, y_i) are locations in the 8-connected neighborhood of (x, y), then $I_C(x, y) = \text{white}$;

4) step 3) is repeated until convergence;

5) set remaining unassigned $I_C(x, y) = \text{black}$.

The above steps can produce more complete and connected edge information (see Fig.2(c)) than using a single threshold while still removing most spurious edges. The retaining edges can be seen to fall into one of two categories: textural details and significant edge contours. Our next goal is to classify the edges into the above two categories. The basic idea is to use the edge density in the binary edge map for this purpose.

To create the binary map I_{Binary}, take I_C as input, and for each white point in I_C, we set *count* equal to the number of the white points in the window centred at the pixel. In all the experiments shown in this paper

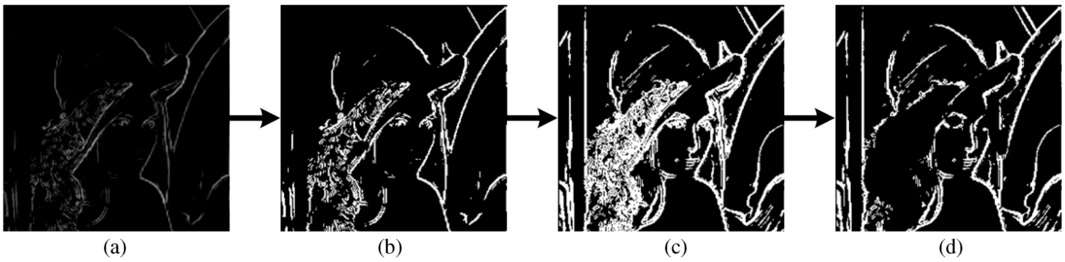


Fig.2. Edge extraction process. (a) Edge magnitudes. (b) Edges after binarised. (c) Edges after connected. (d) Contours of the image.

the window size was set to 21 × 21. Then we have the density:

$$D(x, y) = count(x, y)/A, \tag{4}$$

where A is the area of the window. For pixels on image contours we would expect D to have a low value compared to pixels in the areas with textural detail. Then we can produce the binary map I_{Binary} by D and the density-associated threshold T_d:

$$I_{\text{Binary}}(x, y) = \begin{cases} \text{black}, & I_C(x, y) = \text{black} \\ & \text{or } D(x, y) \geqslant T_d; \\ \text{white}, & \text{otherwise}. \end{cases} \tag{5}$$

Using the equation above, we can identify the details and the contours of the image, thereby preserving the detail information (see Fig.2(d)). Fig.2 shows our edge extraction process.

3.2 Edge-Aware Level Set Diffusion

The level set method was first devised by Osher and Sethian[18,19] and is used widely[20]. In image processing level set equation[4] can be written as (6):

$$I_{\text{Flow}} = F\|\nabla I\| = -curv(I)\|\nabla I\|, \tag{6}$$

where F is the speed of the movement of the curve in its normal direction and I is the input image. By using the curvature $-curv(I)$ to replace F, curves will diffuse at a rate proportional to their curvature, i.e., areas with high curvature diffuse more quickly than those with low curvature. The curvature is defined as

$$\begin{aligned} curv(I) &= div(\nabla I/\|\nabla I\|) \\ &= -\frac{I_x^2 I_{yy} - 2I_x I_y I_{xy} + I_y^2 I_{xx}}{(I_x^2 + I_y^2)^{3/2}}. \end{aligned} \tag{7}$$

Substituting this into (6) and calculating $\|\nabla I\|$ as $\sqrt{I_x^2 + I_y^2}$ we get the desired diffusion:

$$I_{\text{Flow}} = \frac{I_x^2 I_{yy} - 2I_x I_y I_{xy} + I_y^2 I_{xx}}{I_x^2 + I_y^2} \tag{8}$$

and the iterative procedure can be written as:

$$I_{n+1} = I_n + \alpha I_{\text{Flow}} \tag{9}$$

where n is the iteration number. Experiments show that setting $n = 8$ is sufficient for the diffusion process to converge to a stable state. α is the factor of diffusion magnitude, for all the experiments in this paper we set $\alpha = 1$, and we implement I_x, I_{xx}, etc. using the central difference scheme:

$$I_x(x, y) = (I(x - h, y) - I(x + h, y))/2h, \tag{10}$$

$$I_{xx}(x, y) = (I(x + h, y) + I(x - h, y) - 2I(x, y))/h^2, \tag{11}$$

where h is the spatial step, here it is set to 1.

Using (9) jaggies are smoothed and artifacts are diminished. But standard level set diffusion does not only smooth the jaggies, it smoothes some details of the image too (see Fig.3(b)). To avoid over-smoothing we must introduce some constraints. [4] proposed anchor constraint to preserve the fidelity of the original image. The constraint can be represented as follows:

$$I_{\text{Flow}} = \begin{cases} 0, & \text{for original pixels;} \\ -curv(I)\|I\|, & \text{otherwise;} \end{cases} \tag{12}$$

which ensures that the original low resolution pixel information is preserved.

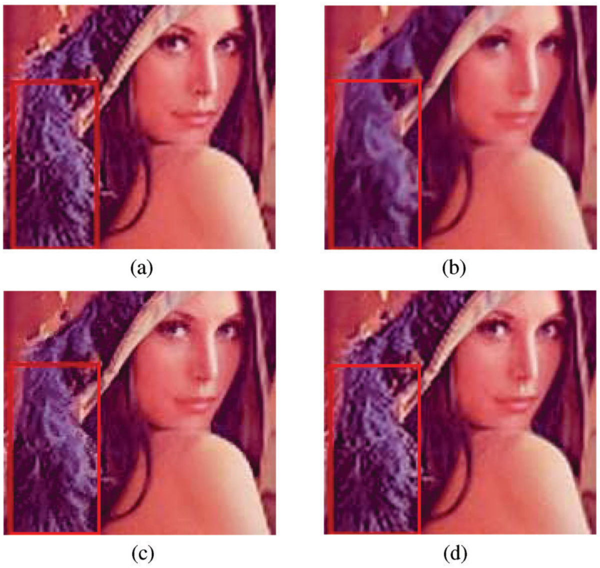


Fig.3. Lena image magnified by a factor of 3. (a) Bicubic interpolation. (b) Unconstrained level set diffusion. (c) Anchor constrained level set diffusion. (d) Our approach.

However, this method causes two problems:

1) anchor constraint preserves the original pixel information but the jaggies are not totally removed;

2) although anchor constraint is used, level set diffusion is still performed over the whole image, so it will tend to corrupt the image in highly detailed (textured) areas.

To solve these problems we constrain the application of level set diffusion in a different way. Note that the different features have different behaviors after performing level set diffusion: the contours may become smooth, but the area of textural details and the corners (i.e., the points with high curvature) are likely to be degraded. This suggests that level set diffusion should just be applied to the contours and not to textural details or corners in the image. This approach has two advantages:

1) smooth the image contours sufficiently;

2) preserve the textural details and the corner features of the image.

We have already extracted the edges and generated a binary map to identify the contours in Subsection 3.1, but this does not specially identify corners and avoid including them in the binary map. This is now done in the following stage. The curvature need only be calculated at positions specified by the previous binary map $I_{\rm Binary}$. Then by choosing a threshold T_c and using it to modify $I_{\rm Binary}$ slightly, we can generate the new binary map $\widetilde{I}_{\rm Binary}$:

if $I_{\rm Binary}(x,y) = $ black then $\widetilde{I}_{\rm Binary}(x,y) = $ black;

if $I_{\rm Binary}(x,y) = $ white, then

$$\widetilde{I}_{\rm Binary}(x,y) = \begin{cases} \text{black}, & curv(I(x,y)) \geqslant T_c; \\ \text{white}, & curv(I(x,y)) < T_c; \end{cases} \tag{13}$$

where $curv(I(x,y))$ is defined by (7) and calculated from the interpolated image.

Now we perform level set diffusion only at the white points in $\widetilde{I}_{\rm Binary}$ to smooth the contours while preserving the details. Our method is compared with others in Fig.3: (a) is the bicubic interpolation which suffers the worst jaggies and blurry edges; (b) is the unconstrained level set diffusion, the jaggies disappear but the image becomes very blurred; (c) is the level set reconstruction using anchor constraint, preserving the fidelity of the original image but still destroying some details such as the feather (highlighted by the rectangle), eyes and so on; (d) is our result, smoothing the edge jaggies and also preserving the image details.

3.3 Bilateral Filtering for Blur Removal

Although the jaggies in the image have been successfully removed, due to interpolation errors, there is still blurring across the image contours that has not been corrected by level set diffusion. Therefore, it is not enough to reconstruct an image using only level set diffusion. Figs. 4(a) and 4(b) show the change of contours in above two steps.

Bilateral filtering[21] is an effective method for smoothing images while preserving their edges. It was improved by [22], [23] and has been widely applied in many fields[24], such as adaptive image smoothing[25], image and video extraction[26], image denoising[27]. Not only does it perform anisotropic smoothing, but we can also use this technique to sharpen the blurred contours.

The characteristic of bilateral filtering is that it not only uses the distance between the central pixel and a distant pixel to determine its influence on the averaging process, but also considers the similarity between the pixels' properties. For an image I bilateral filtering is defined as:

$$H(\widehat{x}, \sigma_c, \sigma_s) = \frac{1}{K}\int I(x)c(x,\widehat{x})s(I(x), I(\widehat{x}))\mathrm{d}x \tag{14}$$

where $c(x,\widehat{x})$ represents the spatial relationship between the two pixels in the image and $s(I(x), I(\widehat{x}))$ represents the feature relationship between the two pixels. K represents the normalizing coefficient which can be defined as:

$$K = \int c(x,\widehat{x})s(I(x), I(\widehat{x}))\mathrm{d}x. \tag{15}$$

A popular choice for $c(x,\widehat{x})$ and $s(I(x), I(\widehat{x}))$ is:

$$c(x,\widehat{x}) = e^{-\frac{1}{2}(\frac{\|x-\widehat{x}\|^2}{\sigma_c^2})}, \tag{16}$$

$$s(I(x), I(\widehat{x})) = e^{-\frac{1}{2}(\frac{\|I(x)-I(\widehat{x})\|^2}{\sigma_s^2})}. \tag{17}$$

where σ_c is related to the blur radius and σ_s determines how contrasts will be preserved or blurred. Since different images are insensitive to these two parameters, we set them as the constant values $\sigma_c = 5.0$ and $\sigma_s = 15.7$ throughout the paper.

After bilateral filtering, pixels around edges will be averaged in opposite directions (Fig.5 shows this process). Therefore, we can use this method of image reconstruction in order to make the edges sharper (see Fig.4(c)).

Bilateral filtering can produce sharp contours but will also smooth some details, so we must introduce some constraints to prevent blurring being applied to non-edge areas. Now the problem model can be written as follows:

$$I_{\rm Output} = F(H(I, \sigma_c, \sigma_s), \nabla I, D), \tag{18}$$

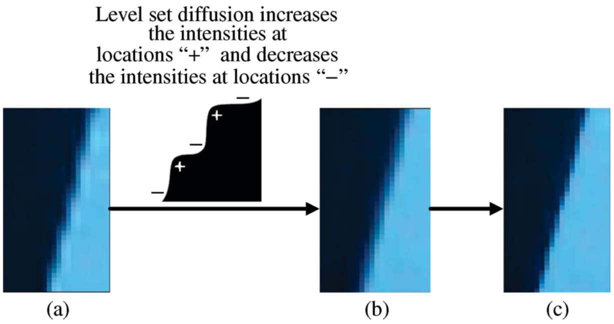

Level set diffusion increases the intensities at locations "+" and decreases the intensities at locations "–"
+
–
+
–
–
(a)
(b)
(c)

Fig.4. Edges change effect. (a) Edges after bicubic interpolation, which are both jaggy and blurry. (b) Edges after level set diffusion, where jaggies are removed but are still blurry. (c) Edges after bilateral filtering, which are sharper and clarified.

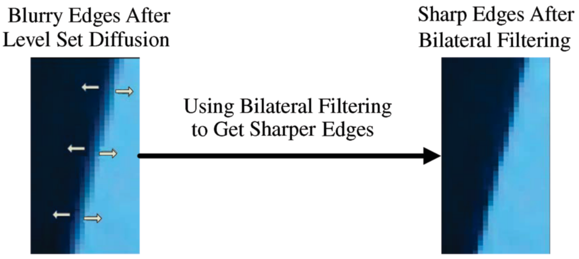


Fig.5. Edges sharpened by bilateral filtering.

where $H(\bullet)$ is the bilateral filtering operator, ∇I is the gradient of the input image and D represents the edge density defined in Subsection 3.1. There are two feasible cases of function F.

1) Performing bilateral filtering continuously in the whole image and introducing a weight to control its intensity. Assume that the input image is I and the image after bilateral filtering is $\widetilde{I}$, we can set the output image to:

$$I_{\text{Output}} = \alpha I + (1-\alpha)\widetilde{I}, \quad \text{where } \alpha = G(\nabla I, D) \quad (19)$$

and G is some function to be defined to control the amount of filtering.

2) Just apply bilateral filtering to some parts of the image, specifically to significant contours where jaggies and blurring are likely to occur. Therefore, we first extract the edges of the image and identify the significant contours after removing the textural details, and then use bilateral filtering at just those contours, so as to produce sharp contours whilst preserving the image details.

Since it is difficult to find a suitable function G to describe the relation among α, ∇I and D, we adopt the second scheme and use the edge extraction method described in Subsection 3.1. This step concludes our whole reconstruction process.

Some other traditional sharpening filters[28] can also produce good results. Fig.6(b) and Fig.7(b) are filtered by Photoshop's unsharp masking (USM). Edges are visibly enhanced compared with the original image, see Fig.6(a) and Fig.7(a). The rectangles show the magnified edges. But compared with constrained bilateral filtering (see Fig.6(c) and Fig.7(c)), the USM filter has two defects: 1) suitable parameters need to be chosen for different images for the best effect while bilateral filtering can produce similar results for different images using fixed parameters; 2) USM filtering can enhance edges by amplifying their magnitude, but cannot make them sharp, while bilateral filtering performs well in this respect. Moreover, if bilateral filtering is applied as a pre-processing step before the USM sharpening, the benefits of both approaches are combined to produce the best effects (see Fig.6(d) and Fig.7(d)).

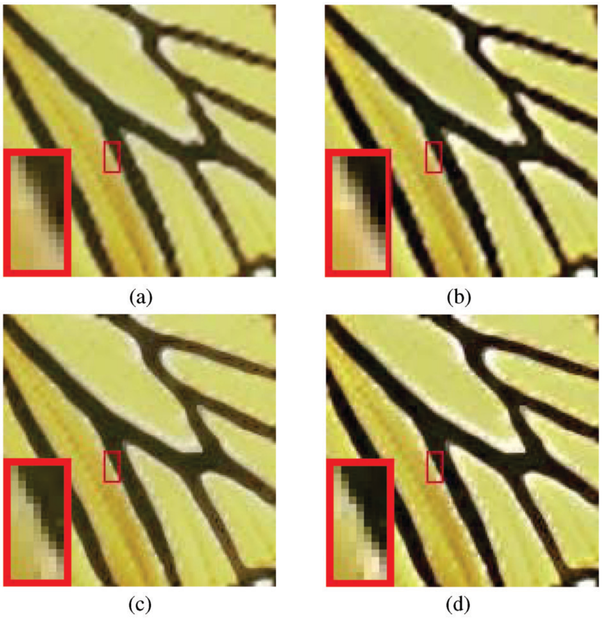


Fig.6. Monarch image using different sharpening filters with parameters. (a) Without sharp filter. (b) USM sharp filtering (amount = 65%, radius = 7.0, threshold = 15). (c) Bilateral filtering (σ_s = 15.7, σ_c = 5.0). (d) USM sharp after bilateral filtering.

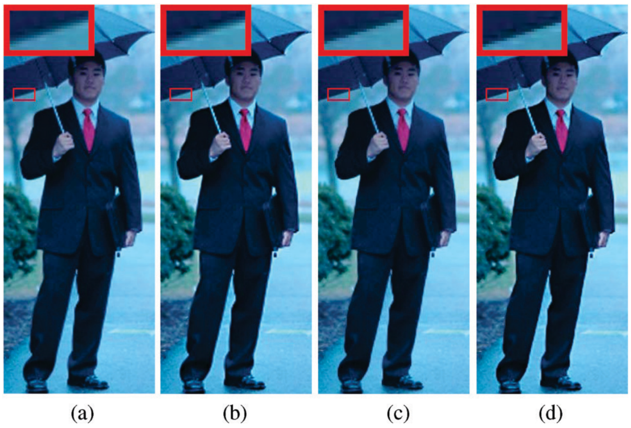


Fig.7. Man image using different sharpening filters with parameters. (a) Without sharp filter. (b) USM sharp filtering (amount = 56%, radius = 4.7, threshold = 13). (c) Bilateral filtering (σ_s = 15.7, σ_c = 5.0). (d) USM sharp after bilateral filtering.

Color Images. Our method can also be extended to color images. In *Lab* color space the human visual system is most sensitive to the luminance channel (L channel). Therefore first we split the image into *Lab* color space, and then magnify the image by performing the initial interpolation independently in all the three channels. After that, our reconstruction post-processing is only applied to the L channel, which is then merged with the interpolated *ab* channels and transformed back to RGB color space. In this way, we can get similar results to performing our work on RGB channels separately but require less computation.

3.4 Fixed Threshold Selection

Since it is laborious for the user to choose the thresholds T_d and T_c, our approach is to estimate appropriate values from training data. In this Subsection we use 10 sets of 100 images that were randomly selected from various sources. Our methodology is then to find threshold values that optimize the quality of results obtained by our method. This requires a scheme to measure image quality, and for that we use the Structural Similarity Image Measurement (SSIM)[29] which is a state of the art method. To improve simple measures such as RMS error, SSIM attempts to incorporate structural and contrast information so as to provide a measure that is perceptually more meaningful. For each threshold the average SSIM value is calculated over 100 images, and the optimal threshold was selected (as described below). This process was repeated over each set of 100 images to check the stability of the threshold estimation. To measure the errors we first take a high-resolution image, reduce it by a factor *f*, then enlarge the low resolution version by the same factor using various methods. These are then compared against the original according to the SSIM.

Because T_c only affects a small percentage of pixels, the SSIM value is not sensitive to changes in this parameter. However, T_c can still have a significant impact on the perceptual impression of the result, since corners are important locations in the image. If T_c is chosen too small, then many non-corners are likely to be removed from the binary map, so preventing level set diffusion operating on many contours. If it is chosen too large, then corner points will not be preserved. Given the difficulty in automatically quantifying the effect of modifying T_c we determine an appropriate value experimentally. For all the results reported in this paper we set T_c to an intermediate value of 13.0.

The Data1 in Fig.8 shows that the SSIM value monotonically decreases with increasing T_d. The value barely decreases at first since there exist visible jaggies. Although the reconstruction work may bring some extra error, it also removes more jaggies and blurring. Then as T_d increases, the reconstruction increasingly affects details which do not contain jaggies. Consequently, this increases the reconstruction error, so from the point $T_d = 0.352$ the SSIM value decreases more rapidly. We also draw the SSIM evaluation curve of two other image sets (see Data2 and Data3 in Fig.8), from the results we can see the SSIM evaluation curve of each set shares the similar general shape excluding some slight local differences. This suggests that an appropriate threshold should be at the bend point of the graph since this is where the largest amount of artifacts can be removed without significantly degrading the quality of the image (i.e., without reducing the SSIM value). We find this point through the following step which is effective even if the graph is noisy or the bend is not sharp[30]. First fit a straight line to the endpoints of the graph. Then find the point on the curve with the maximum perpendicular distance $D_{\max}$ to the line. Following this idea, the threshold in Fig.8 was set to $T_d = 0.352$. Also we chose another 9 independent sets of 100 images and computed the parameters for each set as before. The results were 0.339, 0.343, 0.331, 0.327, 0.357, 0.341, 0.313, 0.360, and 0.338. From the data we can see most of the values lie in a small range [0.320∼0.350], which shows the experimental method is robust and consistent. The average value $T_d = 0.340$ was selected as the fixed threshold, which was further confirmed to be reasonable by visualizing the results it produced on images not included in the training sets of 100 images. Of course, these threshold values for T_c and T_d are not optimal for every image, and can be adjusted by the user if so desired.

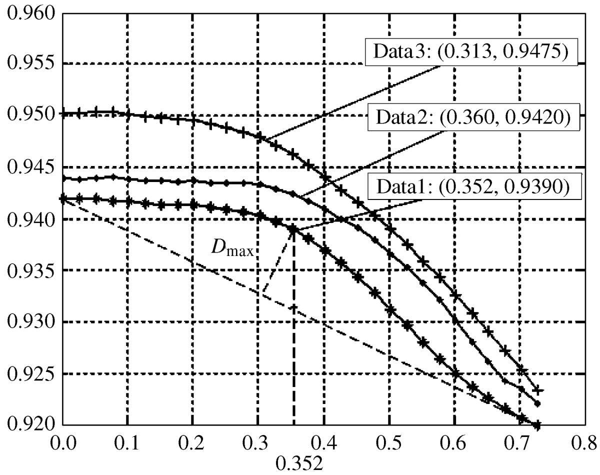


Fig.8. SSIM evaluation over 100 images.

4 Experimental Results

In this section we will show some results of the key steps and compare them with other typical methods. Fig.9 shows the edge detection results using Canny operator (see Fig.9(a)) and our method (see Fig.9(b)). Canny edges cannot cover the entire jaggies region while our edges seem quite different. Moreover, our method is fully automatic and does not require any thresholds to be provided.

Fig.10 shows the results of the detail preserving step for different values of threshold T_d. High values of T_d preserve more details.

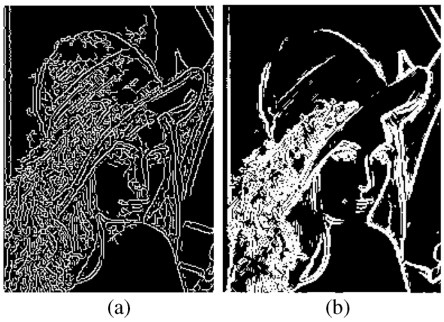


Fig.9. Edge detection results. (a) Canny edges with thresholds 50, 25. (b) Our method after moment-preserving and hysteresis thresholding.

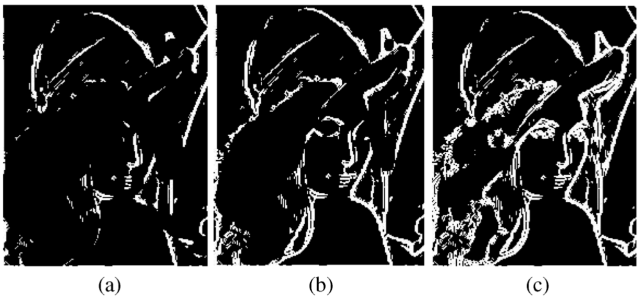


Fig.10. Different levels of detail preservation. (a) $T_d = 0.295$. (b) $T_d = 0.408$. (c) $T_d = 0.567$.

Fig.11 compares the results of magnifying the Lena image by the factor of 4 using different methods. Fig.11(a) is the result using simple pixel replication which suffers the worst jaggies and loss of many significant features. Fig.11(b) is bicubic interpolation, the result is closer to the original high resolution image, but has created some jaggies and blurring at the edges. Fig.11(c) shows level set reconstruction. The image contours become smooth along the contours but still blurry across the contours and some details like hair and eyes had also been destroyed. Fig.11(d) shows our result, which has removed the jaggies and has sharper contours while preserving the details. Fig.11(e) shows the result using the fixed threshold values for T_d and T_c which has a similar quality with the result obtained by manually optimizing the thresholds.

In Fig.12 we can see since the anchor constraint had been used in conventional level set diffusion (Fig.12(d)) some jaggies still exist and the image seems still blurry. In contrast, with our methods (Fig.12(e)) the contour smoothness and sharpness are both visually and quantitatively improved. The result of using the fixed thresholds is shown in Fig.12(f), in this image it does not work so well. In Fig.13, level set diffusion (Fig.12(b)) destroys some details like the face, the shoes etc. Our method (Fig.12(c)) avoids this and can produce sharper contours, the results of the fixed thresholds are comparable to the best thresholds selected by hand.

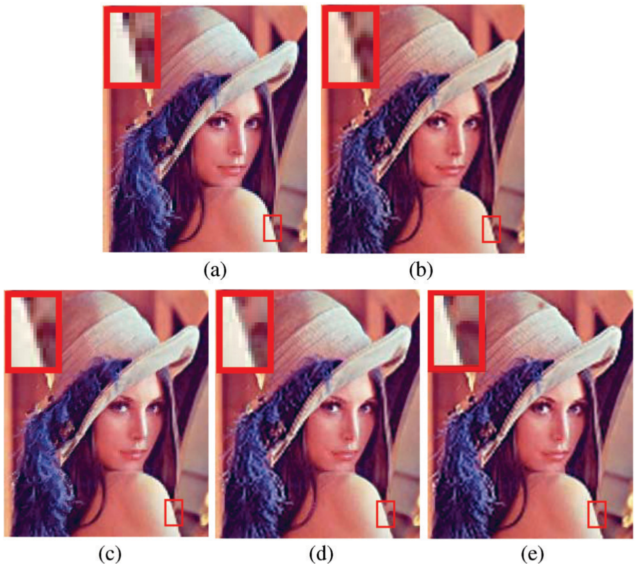


Fig.11. Lena image magnified by the factor of 4. (a) Replication. (b) Bicubic. (c) Level set reconstruction. (d) Our method with the thresholds $T_d = 0.408$, $T_c = 9.3$, $\sigma_s = 15.7$, $\sigma_c = 5.0$. (e) Result with fixed threshold $T_d = 0.340$, $T_c = 13.0$.

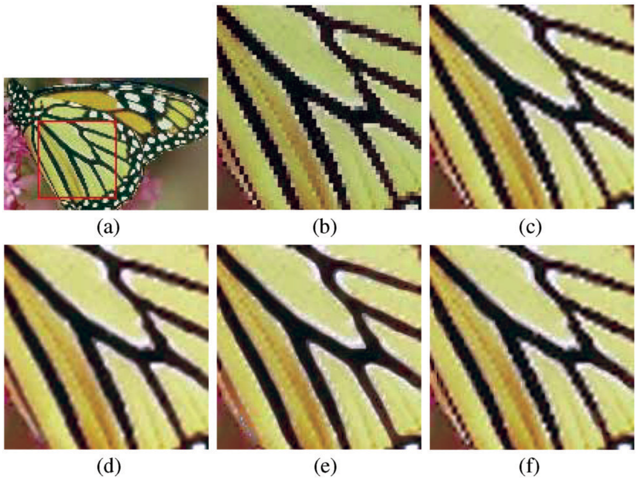


Fig.12. Monarch magnified by the factor of 3. (a) Input image. (b) Replication. (c) Bicubic. (d) Level set reconstruction. (e) Our method with the thresholds $T_d = 0.680$, $T_c = 10.0$, $\sigma_s = 15.7$, $\sigma_c = 5.0$. (f) Result with fixed threshold $T_d = 0.340$, $T_c = 13.0$.

Error Measurement. In Table 1 we give the error measurements between our approach and the original image, these differences are measured by the RMS error and also by the SSIM. From the table we can see that using the RMSE and SSIM measures for comparison, our method, no matter whether thresholds are set manually for each image or the fixed default values are used, is almost the same as the initial interpolation and level set diffusion. Since our reconstruction, when constrained by the thresholds, tends to perform on the jaggies but not on true details, it will not incur much extra error if the thresholds are set properly.

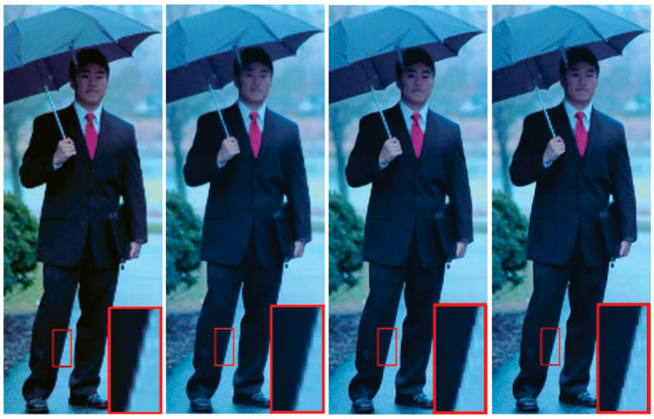


Fig.13. Man image magnified by the factor of 3. (a) Bicubic. (b) Level set reconstruction. (c) Our method with the threshold $T_d = 0.317$, $T_c = 13.0$, $\sigma_s = 15.7$, $\sigma_c = 5.0$. (d) Result with fixed threshold $T_d = 0.340$, $T_c = 13.0$, $\sigma_c = 5.0$.

Table 1. RMSE and SSIM for Different Methods

	Methods	RMSE	SSIM
Lena	Bicubic	9.4792	0.9233
	Level Set	9.5177	0.9197
	Our Method ($T_d = 0.408/T_c = 9.3$)	9.5543	0.9141
	Fixed Threshold ($T_d = 0.347/T_c = 13.0$)	9.5431	0.9133
Man	Bicubic	11.5137	0.9448
	Level Set	12.0588	0.9375
	Our Method ($T_d = 0.317/T_c = 13.0$)	13.2047	0.9318
	Fixed Threshold ($T_d = 0.347/T_c = 13.0$)	14.0319	0.9273

We note that neither RMS error nor SSIM correctly rank the magnification methods in correspondence with visual perception, so we additionally performed human testing. Results were generated by applying several methods to the test images. These were presented randomly to 20 subjects (university students are not associated with the research) who were asked to rate every result. The scores were allowed to range from 1 (worst) to 10 (best). Table 2 shows the average scores for each result. For all the three images, our approach with the manual threshold was considered to produce the best quality by most of the users, and the scores are all above 9.0. The fixed threshold results perform well for images Lena and Man, and the scores are just a little low than manual threshold, ranked 2. But for image Monarch, the fixed thresholds are significantly different from the manual threshold, so got lower scores. Level set reconstruction results were considered better than that of the bicubic method and ranked 3 for images Monarch and Man, but since it destroyed some details of image Lena, the score is the lowest for this image. The bicubic method, due to the visible jaggies, got low scores for all the three images. Besides these three images, we also chose 7 other images at random and give an average score of the 10 images at the last column in Table 2. The results show that our method with manual thresholds got the highest score, and the fixed threshold is a little worse, but better than the other two methods. Bicubic method got the lowest score and the difference from other three results is apparent.

Table 2. User's Rank

	Lena	Man	Monarch	Avr.
Bicubic	3.7	1.1	1.3	1.74
Level Set	3.2	7.3	6.9	6.32
Our Method	9.9	9.7	10.0	9.57
Fixed Threshold	7.5	9.1	3.7	8.69

Processing Time. We implemented our work on the CPU using OpenCv. All of the tests were performed on an Intel Pentium Dual 2.2GHz with Windows XP, ATI Radeon HD2400 PRO, and 2GB memory PC. Performance value depends on the image size and the parameters. Table 3 shows the processing times with different images.

Table 3. Processing Time

	Lena	Man
	$T_d = 0.408$	$T_d = 0.317$
	$T_c = 9.8$	$T_c = 13.0$
Before Processing (ms)	110	150
After Processing (ms)	175	282
Processing Time (ms)	62	130

5 Conclusion

This paper proposes an edge-aware level set diffusion and bilateral filtering-based image magnification reconstruction method. After the initial interpolation, textural details and the significant contours of the image are identified. Then edge-aware level set diffusion is applied to these significant contours to remove the jaggies, followed by bilateral filtering at the same locations to reverse the blurring introduced by the initial interpolation and the level set diffusion processes. These processes produce sharp contours without jaggies while preserving the details of the image. The results show that the overall RMS error of our method barely increases while contour smoothness and sharpness are substantially improved.

References

[1] dell'Acqua F, Gamba P. Using image magnification techniques to improve classification of hyperspectral data. In *Proc. International Geoscience and Remote Sensing Symposium*, Toulouse, France, July 21–25, 2003, pp.737–739.

[2] Kwiatkowski J, Kwiatkowska W, Kawa K, Kania P. Using fractal coding in medical image magnification. In *Proc. Parallel Processing and Applied Mathematics*, Naleczow, Poland, September 9–12, 2002, pp.517–525.

[3] Gehani A, Reif J. Advances in digital forensics iii. In *Proc. International Federation for Information Processing*, Addis Ababa, Ethiopia, August 22–24, 2007.

[4] Morse B S, Schwartzwald D. Image magnification using level-set reconstruction. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Kauai Island of Hawaii, USA, December 11–13, 2001, pp.333–340.

[5] Irani M, Peleg S. Motion analysis for image enhancement: Resolution, occlusion, and transparency. *Journal of Visual Communication and Image Representation*, 1993, 4(4): 324–335.

[6] Naiman A C. Jagged edges: When is filtering needed? *ACM Trans. Graphics*, 1998, 17(4): 238–258.

[7] Dai S, Han M, Wu Y, Gong Y. Bilateral back-projection for single image super resolution. In *Proc. IEEE International Conference on Multimedia and Expo*, Beijing, China, July 2–5, 2007, pp.1039–1042.

[8] Allebach J, Wong P W. Edge-directed interpolation. In *Proc. International Conference on Image Processing*, Lausanne, Switzerland, March, 1996, pp.707–710.

[9] Li X, Orchard M T. New edge directed interpolation. In *Proc. International Conference on Image Processing*, Vancouver, Canada, September 10–13, 2000, pp.1521–1527.

[10] Zhang S H, Chen T, Zhang Y F, Hu S M, Martin R R. Vectorizing cartoon animations. *IEEE Trans. Visualization and Computer Graphics*, 2009.(to appare)

[11] Fattal R. Image upsampling via imposed edge statistics. *ACM Trans. Graphics*, 2007, 26(3): 95.

[12] Sun J, Zheng N N, Tao H, Shum H Y. Image hallucination with primal sketch priors. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Monona Terrace Convention Center Madison, Wisconsin, USA, June 16–22, 2003, pp.729–736.

[13] Tumblin J, Choudhury P. Bixels: Picture samples with sharp embedded boundaries. In *Proc. Eurographics Workshop on Rendering Techniques*, Norköping, Sweden, June 21–23, 2004, pp.255–264.

[14] Rubinstein M, Shamir A, Avidan S. Improved seam carving for video retargeting. In *Proc. ACM SIGGRAPH*, Los Angeles, California, USA, August 11–15, 2008, pp.1–9.

[15] Zhang Y F, Hu S M, Martin R R. Shrinkability maps for content-aware video resizing. *Comput. Graph. Forum*, 2008, 27(7): 1797–1804.

[16] Huang H, Fu T N, Rosin P L, Qi C. Real-time content-aware image resizing. *Science in China Series F: Information Sciences*, 2009, 52(2): 172–182.

[17] Tsai W H. Moment-preserving thresholding: A new approach. *CVGIP: Image Understanding*, 1985, 29(3): 377–393.

[18] Osher S. A level set formulation for the solution of the Dirichlet problem for Hamilton-Jacobi equations. *SIAM Journal on Mathematical Analysis*, 1993, 24(5): 1145–1152.

[19] Sethian J A. Level Set Methods: Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision and Materials Science. Cambridge University Press, 1996.

[20] Bajaj C L, Xu G L, Zhang Q. Higher-order level-set method and its application in biomolecular surfaces construction. *Journal of Computer Science and Technology*, 2008, 23(6): 1026–1036.

[21] Tomasi C, Manduchi R. Bilateral filtering for gray and color images. In *Proc. International Conference on Computer Vision*, Bombay, India, January 4–7, 1998, pp.839–846.

[22] Elad M. On the origin of the bilateral filter and ways to improve it. *IEEE Trans. Image Processing*, 2002, 11(10): 1141–1151.

[23] Buades A, Coll B, Morel J M. The staircasing effect in neighborhood filters and its solution. *IEEE Trans. Image Processing*, 2006, 15(6): 1499–1505.

[24] Paris S, Kornprobst P, Tumblin J, Durand F. A gentle introduction to bilateral filtering and its applications. In *ACM SIGGRAPH Courses*, San Diego, California, USA, August 5–9, 2007.

[25] Barash D. A fundamental relationship between bilateral filtering, adaptive smoothing, and the nonlinear diffusion equation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2002, 24(6): 844–847.

[26] Winnemöller H, Olsen S C, Gooch B. Real-time video abstraction. *ACM Trans. Graphics*, 2006, 25(3): 1221–1226.

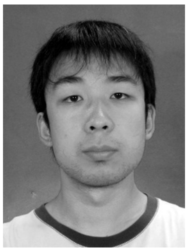
[27] Liu Y L, Wang J, Chen X, Guo Y W, Peng Q S. A robust and fast non-local means algorithm for image denoising. *Journal of Computer Science and Technology*, 2008, 23(2): 270–279.

[28] Aubert G, Kornprobst P. Mathematical Problems in Image Processing: Partial Differential Equations and the Calculus of Variations. Springer Verlag, 2002.

[29] Wang Z, Bovik A C, Sheikh H R, Simoncelli E P. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Processing*, 2004, 13(4): 600–612.

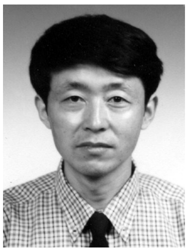
[30] Rosin P L. Unimodal thresholding. *Pattern Recognition*, 2001, 34(11): 2083–2096.

Hua Huang is an associate professor in School of Electronics and Information Engineering, Xi'an Jiaotong University, and is a senior member of CCF and a member of IEEE. He received his B.S. and Ph.D. degrees from Xi'an Jiaotong University in 1996 and 2006, respectively. His main research interests include image and video processing, pattern recognition and machine learning.



Yu Zang received the Bachelor's degree in School of Electronics and Information Engineering, Xi'an Jiaotong University in 2008. He is currently studying for his Ph.D. degree in Xi'an Jiaotong University. His main research interests include image and video processing.

Paul L. Rosin is currently working on the School of Computer Science, Cardiff University. His previous posts include lecturer at the Department of Information Systems and Computing, Brunel University London, UK, research scientist at the Institute for Remote Sensing Applications, Joint Research Centre, Ispra, Italy, and lecturer at Curtin University of Technology, Perth, Australia. His research interests include the representation, segmentation, and grouping of curves, knowledge-based vision systems, early image representations, low level image processing, machine vision approaches to remote sensing, methods for evaluation of approximations, algorithms, etc., medical and biological image analysis, mesh processing, and the analysis of shape in art and architecture.



Chun Qi received the B.S. degree in electronic engineering in 1982 from Xidian University, and Ph.D. degree in information and communication engineering in 2000 from Xi'an Jiaotong University. He is currently a professor in School of Electronics and Information Engineering, Xi'an Jiaotong University, and is a senior member of CCF. His main research interests include image processing, pattern recognition and signal processing.