Image Interpolation by Super-Resolution

Alexey Lukin, Andrey S. Krylov, Andrey Nasonov
Moscow State University, Moscow, Russia
lukin@graphics.cs.msu.ru, kryl@cs.msu.su, nasonov@comail.ru

Abstract

Term "super-resolution" is typically used for a high-resolution image produced from several low-resolution noisy observations. In this paper, we consider the problem of high-quality interpolation of a single noise-free image. Several aspects of the corresponding super-resolution algorithm are investigated: choice of regularization term, dependence of the result on initial approximation, convergence speed, and heuristics to facilitate convergence and improve the visual quality of the resulting image.

Keywords: image interpolation, super-resolution, regularization.

1. INTRODUCTION

Linear methods for image interpolation are usually constructed to deal with bandlimited signals. The interpolated one-dimensional signal is defined as: $F'(x) = \sum_{i=-\infty}^{+\infty} F(ih) K(ih - x)$, where $K(x)$ – is the interpolation filter, h is the sampling step. In a two-dimensional case, the interpolation is typically performed separately for each axis. The most popular weight functions are box filter (or nearest neighbor), tent function (or bilinear), ideal low-pass filter, Lanczos filter, Gaussian filter, and bicubic interpolation [1].

For every algorithm which is using linear interpolation there are some typical artifacts: blurriness, ringing effect, and jagged edges [6]. Reduction of one of these artifacts increases the others.

As usual, non-linear algorithms are used to scale two-dimensional images with a fixed ratio without constructing continuous image. Interpolated pixel values are calculated as a linear combination of nearest sampled values, but the main difference with the linear interpolation is the variability of coefficients which depend on surrounding pixel intensities.

The main idea of *gradient algorithms* is the fact that directed interpolation along edges results in better interpolation than non-directed linear interpolation. The direction and the intensity of an edge in a point are defined by the local gradient information.

One of these algorithms is WADI [2], which is based on the modification of bilinear interpolation. It computes the derivate along the normal to every side of a square formed by four sampled pixels and modifies coefficients of bilinear interpolation in a special way: the side with greater derivative results in smaller coefficients for points of this side.

Gradient algorithms are fast in the class of non-linear algorithms and produce better results than linear interpolation; it makes edges less jagged and more realistic.

NEDI algorithm (New Edge-Directed Interpolation) is a typical non-linear algorithm, which doubles the resolution of images [3]. It uses the concept of self-similarity. The assumption is that coefficients of linear combination used for destination pixel interpolation are the same as coefficients used for interpolation of source image pixels by pixels of the decimated source image. This algorithm provides very good interpolation quality but it is very complex, so it is often executed only in small areas with strong edges while simpler algorithms process the rest of area.

The rest of the paper is organized as follows. In section 2, we introduce the super-resolution method for image interpolation and the process of regularization. In section 3, we discuss several variants and modifications of the method and describe their influence on the resulting image with respect to visual artifacts and PSNR quality by giving the results of our experiments. Section 4 concludes the paper by summarizing variants of the super-resolution method that provide best image quality.

2. SUPER-RESOLUTION METHOD

Super-resolution method is typically used to restore a high-resolution image from several low-resolution noisy observations [4]. In this paper, we consider the interpolation of a single image. So, we will formulate the problem as

$$Ax = y, \qquad (1)$$

where x is the unknown high-resolution image (represented as a vector of pixel values), y is the known low-resolution image, and A is the downscaling operator typically consisting of decimation D following a low-pass filtering H:

$$A = DH \qquad (2)$$

The choice of the low-pass filtering operator depends on a point spread function of the imaging system that produced the low-resolution image. If the imaging system is unknown we will assume that operator H is a simple box filter.

2.1 Regularization

The equation (1) is generally ill-posed and a small change of the input vector y can cause a huge change of the resulting vector x. For the equation (1), the regularized solution is found as:

$$x = \arg\min \|Ax - y\|_n^p + \alpha F(x), \qquad (3)$$

where the first term is called as *"discrepancy"*, $F(x)$ is the *stabilizer* and $\alpha > 0$ is the *coefficient of regularization* [5].

The most popular and universal stabilizer is the Tikhonov functional. It is calculated as a grid approximation of the functional:

$$F(x) = \|\Delta x\|_2^2, \qquad (4)$$

and $n=2$, $p=2$. For each $\alpha > 0$ the solution x is correct: it is unique, defined for every y and continuously depends on y. We can write the Euler equation for this case: $(A^T A + \Delta^2)x = A^T y$.

But in this case the algorithm becomes linear because x is the solution of the system of linear equations. So, this method inherits drawbacks of linear interpolation algorithms and we need to find more adaptive stabilizer for image resampling.

We will consider Total Variation (TV) and Bilateral TV (BTV) stabilizers [4], which are working in l_1 norm ($n=1$, $p=1$):

$$TV(x) = \|\nabla x\|_1, \qquad (5)$$

where ∇x is the gradient operator (its modulus),

$$BTV(x) = \sum_{s,t=-p}^{s,t=p} \gamma^{|s|+|t|} \|x - S_x^s S_y^t x\|_1, \qquad (6)$$

where S_x^s and S_y^t are shift operators along x and y axes by s and t pixels respectively, $\gamma=0.8$.

2.2 Inverse iterations

To solve the equation (3) with a stabilizer (6) the iterative steepest-descent method can be used:

$$x_{n+1} = x_n - \beta\{H^T D^T sign(DHx - y) +$$
$$+ \alpha \sum_{s,t=-p}^{s,t=p} \gamma^{|s|+|t|} (I - S_x^{-s} S_y^{-t}) sign(x - S_x^s S_y^t x)\} \qquad (7)$$

$z=sign\ x$ is a vector with per-element applied sign function; D^T is an up-scaling operation. If D in (2) is the simplest decimation operator that takes every k-th pixel, D^T is the up-scaling operator by zero insertion. If H in (2) is a symmetric filtering, then H^T is equal to H. x_0 is the initial approximation of the high resolution image.

3. STUDY OF THE METHOD

We have used the described method to perform image enlargement by the factor of 2. The operator H has been set to box filter.

The following paragraphs describe our experiments with various modifications of the super-resolution method and obtained results. To evaluate the quality of results, we have selected several diverse test images and scaled them down using a box filter. Then we applied different variants of the super-resolution algorithm and compared the enlarged images with original high-resolution ("ground truth") copies. Thus, descriptions of noticed artifacts will be supported with PSNR measurements.

3.1 Dependence on initial approximation

The simplest form of the initial approximation high-resolution image would be a zero image. However such an approximation will take long to converge to the solution. A better choice would be the original image upscaled by some simple algorithm, such as bilinear interpolation or a gradient-directed interpolation.

We have found that using gradient-directed interpolation as initial approximation image results in slightly better quality that when using bicubic interpolation and significantly better quality than using bilinear interpolation. Bilinear interpolation used for initial approximation leaves the jagged edges artifact in the resulting image, and it cannot be completely eliminated by the following super-resolution iterations.

Even better results can be obtained using NEDI enlargement of the original image as initial approximation.

We have tested this method with different initial images: zero-filled, bilinear, bicubic, gradient-interpolated and using NEDI algorithm aimed at getting the fastest convergence of iterative method. We'd like to get a good approximation after small number of iterations. That's why using the best possible initial ap-

proximation (NEDI) helps getting better results after fixed number of iterations. Fig. 1 displays improvement of SNR (ISNR) against bilinear interpolation for the super-resolution method with different initial approximations.
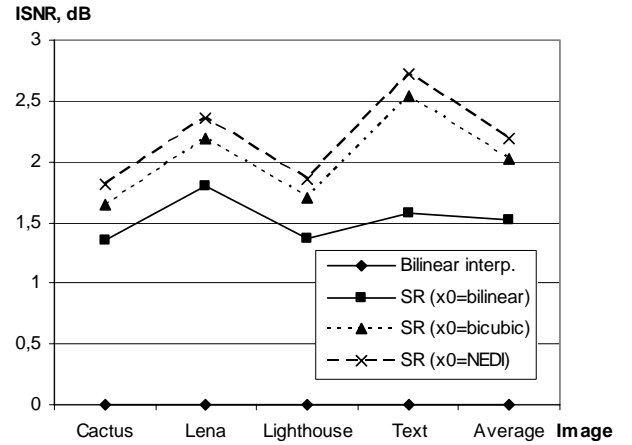

Figure 1: Dependence of SNR on initial approximation after 16 iterations.

3.2 Convergence speed

In the described iterative method, we can't reach a precision less than β, because *sign* function can only take values of -1, 0, and 1 and it is multiplied by β to form the correction term at each iteration. So, if we want to get closer to the optimal image we need to decrease the value of β and increase the number of iterations.

Also it can be noted that after a certain number of iterations with constant β, iterated images start fluctuating around the optimal image without approaching the target image closely.

To improve the convergence speed we can use the variable coefficient β. First, we have chosen the following way: a number of iterations are processed with a constant coefficient until the image starts fluctuating. When this happens, β is decreased by some fixed ratio and the process continues until the next start of fluctuations and so on. The task is to detect fluctuations and to choose the optimal ratio of β modification. We need to operate with the discrepancy to detect the fluctuations. Before the beginning of fluctuations the discrepancy has a tendency to decrease, but when fluctuations begin the discrepancy starts to randomly oscillate around some mean value. So, the following algorithm can be applied: we count a discrepancy at each iteration and when it becomes greater than on the previous iteration, the counter of oscillations is increased. If a counter is greater than a threshold, it is reset and the value of β is decreased. The greater value of the threshold results in more precise detection of the beginning of fluctuations, but makes the convergence speed lower. In practice we set the threshold value to 2 oscillations.

Finally, we need to define the divisor for β. Practically, small divisors lower the convergence speed, while big divisors result in worse quality, but there isn't any optimal intermediate value. If the divisor is big the algorithm needs more iterations before the image starts to fluctuating than when the divisor is smaller. The optimal divisor range is from 2 to 4. For any value of divisor from this range, the convergence speed and the quality are approximately the same.

We have noted that the number of iterations between updates of β for a fixed divisor depends very weakly on the source image and iteration number. So, after the first fluctuation, we may use the geometric progression instead of the piecewise constant behavior of β (fig. 2).

So, we have used the following strategy: a certain number of first iterations are processed with a constant coefficient, and after the beginning of fluctuations, the coefficient is exponentially decreased (geometrical progression). Good selection of the initial approximation image makes it possible to fix the number of iterations with some constant initial β and the method becomes independent on the source image.

Here are some graphs for different ways of changing β. The thick line is β, the thin line is the discrepancy, and the dotted line is the mean square error between the current solution x and the ground truth result. The vertical scale is logarithmic.
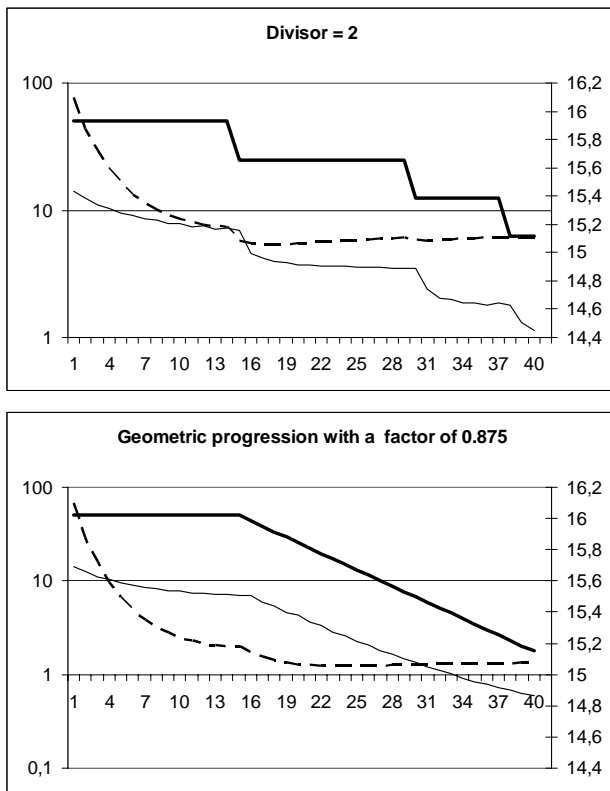

Figure 2: Variation of MSE and discrepancy depending on β.

3.3 Effects of regularization

3.3.1 Ringing suppression

Despite of excellent interpolation of edges, iterative method using TV and BTV has some artifacts: big regularization coefficient results in watercolor effect and loss of fine details, but it very strongly suppresses the ringing artifact (Gibbs phenomenon). Image becomes piecewise constant; in other words it looks "painted" with strokes of paintbrush. Small regularization amount α leaves much ringing around edges and doesn't make them sharper or less jagged. If the first approximation image was produced by a bad interpolation method, the super-resolution method won't improve it in absence of regularization.

Strong regularization using Tikhonov stabilizer results in strong blur, weak regularization leaves jagged edges.

$l1$-regularization (such as TV or BTV) is different: strong regularization sharpens the edges, trying to make the image piecewise-constant [6]. The Gibbs phenomenon is effectively suppressed.

3.3.2 Noise reduction

Many real images captured at low resolution contain noise, typically white noise. Different interpolation algorithms have different tolerance to such noise, some of them smooth the noise, and others tend to amplify it. For example, bilinear interpolation usually smoothes noise (together with image contours), while bicubic or 12-tap NEDI [3] interpolations slightly amplify it (due to overshoot/ringing property of their resampling filters).

Super-resolution algorithm produces even sharper images and the high-frequency component of noise is typically amplified. However the process of regularization helps preventing noise amplification by minimizing the total variation of the resulting image. If the strength of regularization (value of α) is increased, the noise is suppressed, while the sharpness of image contours is preserved, see [6] for examples.

3.4 Effects of changing the internal upsampling algorithm

The whole iterative method of super-resolution (7) can be viewed as several simple steps:

1. Downscale the current approximation image.
2. Compare it with the original low-resolution image and truncate the difference using the *sign* function.
3. Enlarge the difference.
4. Add the (amplified) enlarged difference to the current approximation.
5. Add the (amplified) regularization term to the current approximation.
6. Go to 1 until convergence.

The next improvement of the iterative method is modifying D and H operators. Originally, the upsampling $H^T D^T$ of the *sign* operator of the discrepancy $H^T D^T sign(DHx - y)$ is simple: we interpolate the discrepancy with zeros and apply the filter H^T. This is a linear method and it does not preserve edge directions, so it results in some edge jaggedness. We propose to use edge-directional interpolation (such as gradient interpolation) at this stage to reduce effect of jagged edges. The coefficients of gradient interpolation can be calculated only once, using the first approximation image x_0. The gradient interpolation is applied at each iteration to enlarge the modified difference. This modification of the original method increases both PSNR and visual quality. Fig. 3 compares PSNR for bilinear vs. gradient upsampling of discrepancy.
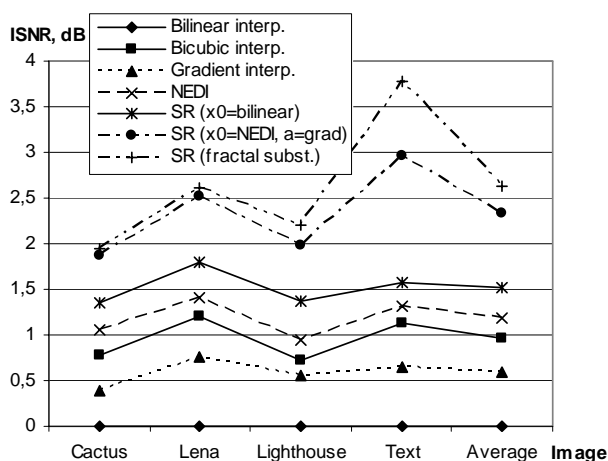
Figure 3: Dependence of SNR on discrepancy upsampling algorithm.

3.5 Future work

We are currently investigating the following modifications of the super-resolution algorithm.

The strength of regularization can be made image-adaptive. The main purposes of regularization are reduction of ringing and reduction of noise level. So, it is possible to apply more regularization in areas of high-contrast edges and in noisy areas, while at the same time preventing excessive "watercolor" artifact in other areas.

In the effort to generate plausible high-frequency content, we have developed a patch-based algorithm for substitution of high-frequency details in the current approximation x_n from the original low-resolution image y. We call this stage "fractal substitution" because substituted details are selected using a self-similarity criterion across 2 scales.

4. CONCLUSION

We have analyzed properties of super-resolution method applied to the interpolation of a single image. It was shown that modifications of the original super-resolution method given by (7) can improve the visual quality and PSNR of resulting images. The example of work of the algorithm is given on a fig. 4, more examples can be found in [6].

The careful choice of initial approximation prevents jagged edges artifact from happening. We suggest using the NEDI algorithm to calculate the initial approximation image.

The gradient-directed interpolation used for upsampling of the difference image prevents jagged edges occurring in the process of iterations.

The choice of Bilateral TV regularization (6) doesn't blur the edges while reducing the ringing artifact and limiting the noise amplification.

The adaptive way of modification of β parameter allows reducing the number of required iterations to 16-20 iterations for producing results with best PSNR.

The simulation on a 1 GHz P3 computer has shown that it takes approximately 1 minute to produce a 1-megapixel enlarged image from a quarter-megapixel source image (for 16 iterations).

Figure 4: Comparison of visual quality for bilinear interpolation (left) and the proposed super-resolution algorithm (right).

5. REFERENCES

[1]   Ken Turkowski "Filters for Common Resampling Tasks" // "Graphics gems", pp. 147-165, Academic Press Professional, Inc., 1990.

[2]   Shuai Yuan, Masahide Abe, Akira Taguchi, Masayuki Kawamata "High Accuracy WADI Image Interpolation with Local Gradient Features" // Proceedings of 2005 International Symposium on Intelligent Signal Processing and Communication Systems pp. 85-88, 2005.

[3]   J.A. Leitao, M. Zhao and G. de Haan "Content-Adaptive Video Up-Scaling for High-Definition Displays" // Proceedings of IVCP 2003, vol. 5022, January 2003.

[4]   Sina Farsiu, Dirk Robinson, Michael Elad, Peyman Milanfar "Fast and Robust Multi-Frame Super-Resolution" // IEEE Trans. On Image Processing, Vol. 13, No. 10, pp. 1327-1344, October 2004.

[5]   A. Tikhonov, V. Arsenin "Solutions of Ill-Posed Problems" // Washington DC: WH Winston, 1977.

[6]   Demo web page with image examples:
http://audio.rightmark.org/lukin/graphics/superres.htm

About authors

Andrey S. Krylov is an associated professor at the Moscow State University, Faculty of Computational Mathematics and Cybernetics.
Email: kryl@cs.msu.su

Alexey Lukin is a member of scientific staff at the Moscow State University, Faculty of Computational Mathematics and Cybernetics, member of IEEE and AES.
Email: lukin@graphics.cs.msu.su

Andrey Nasonov is a student of the Moscow State University, Faculty of Computational Mathematics and Cybernetics.
Email: nasonov@comail.ru