LEARNING SPATIALLY-VARIABLE FILTERS FOR SUPER-RESOLUTION OF TEXT

Adrian Corduneanu

Massachusetts Institute of Technology*
Computer Science and Artificial Intelligence Laboratory
Cambridge, MA 02139

John C. Platt

Microsoft Research
Redmond, WA 98052

ABSTRACT

Images magnified by standard methods display a degradation of detail, particularly noticeable in the blurry edges of text. Current super-resolution algorithms that address the lack of sharpness by filling in the image with probable details hallucinate broken outlines when applied to text. Our novel algorithm for super-resolution of text magnifies images in real-time by interpolation with a variable linear filter determined nonlinearly from the neighborhood to which it is applied. We train the mapping that defines the linear filter to specifically enhance edges of text, producing a conservative algorithm that infers the detail of magnified text. Possible applications include resizing web page layouts or other interfaces, and enhancing low resolution camera captures of text. In general, learning spatially-variable filters is applicable to other image filtering tasks.

1. INTRODUCTION

The digital world is rife with low-resolution images that must be rendered at higher resolutions, with printing of web images, browser resizing of web layouts, and rendering low resolution video modes on higher resolution LCD's being some of the examples. Mainstream image magnification by interpolation with a linear filter smoothes out important visual cues such as edges and text, raising the need for super-resolution algorithms that preserve and even introduce plausible detail.

Super-resolution in general is difficult because image magnification is naturally under-determined: at a factor of $4x$ the original can contribute with only 6% of the pixels. Magnification is at all possible only because typical images share common characteristics, and because the human visual system imposes a strong bias on what looks pleasing. For example bicubic interpolation [1] assumes smoothness, that holds most of the time, but breaks along edges. Other algorithms attempt to reduce unpleasant artifacts, such as the level-set method [2] that enhances bicubic filtering by reducing "jaggies".

*Work done while at Microsoft Research.

The higher the scaling factor, the less useful generic priors such as smoothness are — zoomed images seem to lack detail by being for instance overly smooth. Baker [3] argues that super-resolution is still possible if it is based on a set of recognition decisions on the original image. For example, some interpolation algorithms [4][5][6] localize the edges and are tuned to sharpen them.

For images other than natural pictures, and at magnification factors other than small, algorithms that are not aware of the content of the image are not likely to reproduce high-resolution detail, even if they recognize low-level generic features, such as edges. Rasterized text, icons, textures, computer-generated images, represent distinct classes of images that require distinct recognition decisions. For example one extreme algorithm that would produce perfectly looking text would be to recognize fonts with OCR, then re-render, though on non-text regions it may hallucinate text. One powerful algorithm that learns the class to be magnified is that of Freeman [7]. It adds detail to bicubic-interpolated images from patches from training data, and learns well natural images and texture, but cannot learn to magnify text without breaking the character outlines.

We propose a learnable spatially-variant filter that combines the efficiency of real-time linear interpolation with the power of learning the specific class of images to be resized. In the context of super-resolution of images that contain text, we model an optimal MSE filter trained on samples of rendered text whose parameters are the output of a non-linear edge detector [8]. The result is a spatially-variable interpolation algorithm that improves text at high resolution while performing as well as standard interpolation on the rest of the image. Being a linear filter, the method remains robust and conservative, never breaking character outlines, and performing well on false-positive and false-negative text recognition errors.

2. SPATIALLY-VARIABLE INTERPOLATION

The blueprint of the algorithm is to initially magnify the image by bilinear or bicubic interpolation [1], then post-process the result with a trained spatially-variable linear fil-

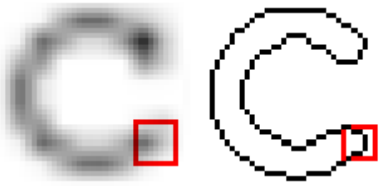

Fig. 1. Features for constructing the spatially-variable filter: a 7x7 patch of the bilinear interpolated image, and the corresponding 5x5 patch from the Canny edge image.

ter. While theoretically the two steps can be collapsed to a single pass of linear interpolation, we still need the intermediate step to construct the spatially-variable filter. The input is therefore a grayscale intensity image of values in $[0, 1]$ magnified to the same size as the desired output. Extending to color images is straightforward and we will present it in Section 3.

Let $\mathbf{x} \in [0,1]^{K^2}$ be the vectorized intensity patch of size K centered at a given point. To obtain the predicted intensity $\hat{y}$ of the point we need to determine the coefficients $\mathbf{w}$ of the linear filter:

$$\hat{y} = \mathbf{w} \cdot (\mathbf{x} - \mu) + \mathbf{x}_{\text{center}} \tag{1}$$

where μ is the mean intensity of the intensity patch and $\mathbf{x}_{\text{center}}$ is the pixel in the intensity patch that corresponds the the predicted pixel. We use this form of filter to ensure that any enhancement is independent of the overall intensity of the image.

To capture the nonlinear peculiarities of text we introduce a point-dependent nonlinear model for $\mathbf{w}$. Because we believe the edge map of the characters capture most nonlinear information in rendered text, we construct the filter from the binary Canny edge-detected [8] version of the bilinearly interpolated image. That the detected edges are also relatively noise free contributes to filters unaffected by variability in the image. We form the feature vector $\mathbf{e}$ from a patch of the edge image centered at the point. We use edge patches of size $E = 5$. Thus, $\mathbf{e}$ is of dimensionality $M = E^2$.

We obtain the coefficients of a spatially-variable linear filter from a fast-to-evaluate linear model on the feature vector:

$$\mathbf{w} = \mathbf{B}\mathbf{e}, \tag{2}$$

where $\mathbf{B}$ is a $K^2 \times M$ matrix of global parameters.

2.1. Optimal linear estimator

We estimate the model parameters $\mathbf{B}$ from pairs of images, one at low resolution, and the other the high-resolution truth. For every pair of corresponding pixels we match the high-resolution truth intensity y^j and the estimate from the low-resolution image

$$\hat{y}^j = \mathbf{B}\mathbf{e}^j \cdot (\mathbf{x}^j - \mu^j) \tag{3}$$

by minimizing the mean squared error of the training set:

$$\text{MSE}(\mathbf{B}) = \sum_{j=1}^{N} (y^j - \hat{y}^j)^2 \tag{4}$$

This criterion is quadratic in $\mathbf{B}$, hence has an analytic solution which is optimal. Note that the filter is *non-linear* in the input intensity $\mathbf{x}$, as $\mathbf{e}$ is a non-linear function of $\mathbf{x}$.

The optimal $\mathbf{B}$ can be derived by considering a new vector $\mathbf{u}^j$. First, compute the matrix resulting from an outer product between the edge image $\mathbf{e}$ (considered as a vector) and the intensity image $\mathbf{x}$ (considered as a vector):

$$u^j_{kl} = e^j_k (x^j_l - \mu^j) \tag{5}$$

. This matrix is then "flattened" into a vector of length MK^2.

The optimal solution the proceeds analogously with optimal linear filtering:

$$\mathbf{B}^{\text{optimal}} = \mathbf{C}^{-1}\mathbf{z}, \tag{6}$$

where $\mathbf{C}$ is the autocorrelation matrix over the flattened vector,

$$\mathbf{C} = \sum_j \mathbf{u}^j \mathbf{u}^{jT}, \tag{7}$$

and $\mathbf{z}$ is the cross-correlation between the flattened vector and the error,

$$\mathbf{z} = \sum_j (y^j - \hat{y}^j)\mathbf{u}. \tag{8}$$

The matrix $\mathbf{C}$ has dimension $MK^2 \times MK^2$. Thus, care must be taken in computing this matrix. Instead of computing the outer product of $\mathbf{u}$ with itself directly, we exploit the sparsity of the $\mathbf{e}$ matrix: only those submatrices of $\mathbf{C}$ corresponding to non-zero $\mathbf{e}$ are updated with the correlation submatrix $(\mathbf{x}^j - \mu^j)(\mathbf{x}^{jT} - \mu^j)$.

3. IMPLEMENTATION

Because contrast resolution is significantly more noticeable than color resolution, we extended the system to color images by applying the spatially-variable filter only to the luminance channel, and use bicubic interpolation for chroma channels. We use the YIQ color space for the experiments shown in Figures 2 and 4.

Should utmost efficiency not be a concern, one may apply the algorithm to the luminance and the chroma, or all three RGB channels.

We selected bilinear as the base interpolation instead of bicubic interpolation because the Canny edge detector picks out the bicubic ringing artifacts. Other edge detectors may be insensitive to ringing, in which case bicubic is an option.

The system is trained on $4\times$ magnification. For lower resolutions we compute the $4\times$ output then subsample it with a bilinear filter. Since the parameters take negligible memory, in practice one should instead train on a variety of magnification factors.

We took utmost care to build a training set robust to the variability of text. We sampled independently the font from 118 fonts with a bias on sans serif, the style (regular, bold, italic), the size (7pt, 9pt, 12pt, or 18pt), and sequences up to 3 glyphs generated from a statistical letter model. The high resolution truth rendered at $4\times$ has been reduced in size by nearest neighbor, box, Gaussian, bilinear, and bicubic filtering to generate a matching low resolution image.

We found that a filter size of $K = 7$ coupled with a edge feature patch of size $E = 5$ achieves good results while not compromising efficiency. Both in training and testing only points whose feature patches intersect some edge need to be considered.

4. EVALUATION

We tested the spatially-variable filter against bicubic interpolation, and the edge-restricted bicubic interpolation algorithm of Xue, et al. [4], and the learning super-resolution patch-based algorithm of Freeman [7]. The test images (Figure 3) consist of a grayscale image of text generated like the training data, for which we also have the $4\times$ high-resolution truth (*glyphs*); a typical UI capture with text and graphics (*word*); and a natural image with complex edges (*oldmill*).

The patch-based algorithm partitions the image scaled by bicubic interpolation into patches, and predicts the high-frequency detail to be added to each patch as a lookup into a large training set of seen pairs of low-frequency/ high-frequency patches. The algorithm maintains consistency by favoring neighboring details that overlap well. We have trained the patch algorithm on $100,000$ pairs from the same training set of generated glyphs. Because it hallucinates unsightly font details in natural images, we only apply it to the *glyph* test image. At the expense of efficiency, our implementation should perform better than the original [7] because the search for optimal patches is exact.

Because for the *glyphs* image we know the $4\times$ truth (from which *glyphs* was derived by various font anti-aliasing algorithms) we can measure the signal-to-noise (SNR) and peak signal-to-noise (PSNR) recovery of the original. We refrain from measuring SNR and PSNR on an image downsampled by a single downsampling algorithm, then scaled back with the test algorithms, because the choice of downsampling may introduce bias. Table 4 shows that in terms of SNR/PSNR the variable filter performs better than all considered algorithms. The results are even more significant given that most of the *glyphs* image contains white areas scaled without error.

	variable filter	**bicubic**	**edge restricted interpolation**	**patch**
SNR	8.52	7.79	7.16	5.94
PSNR	17.72	16.99	16.36	15.14

Table 1. Comparison between the spatially-variable filter, bilinear and bicubic interpolation, Xue's edge-restricted interpolation and Freeman's super-resolution on 40 test glyph images. These images were generated with the 5 training downsampling filters.

The patch-based algorithm is a low performer in regards to text. Visual inspection (Figure 3) confirms it is unsuitable for text as it breaks glyph outlines.

Edge-restricted interpolation over-emphasizes edges in the text image, producing a characteristic "cartoon" image. This is because edge-restricted interpolation ignores all information from the opposite side of the edge, producing step functions. In contrast, the learned spatially-variable filter uses all local pixel information in an optimal way, to best estimate the high-resolution image.

Visually, the spatially-variable filter learned to straighten glyph outlines and induce the thought effect of super-resolution. At the same time the algorithm is conservative enough and does not break glyph outlines. It even improves the appearance of lines, as it appears in the *word* image. The effect of super-resolution extends to natural images, because their edges become sharper and without the characteristic "jaggies".

5. CONCLUSIONS

We proposed a super-resolution algorithm based on interpolation with a spatially-variant filter that can be learned from typical images. In the case of images with text we improved on standard linear interpolation by modeling the filter as a linear function of the edge pattern, whose parameters can be trained by optimizing MSE by sparse matrix inversion. Application of the variable filter is fast and memory efficient, and it improves text outlines with minimal distortion. Edges of natural images also benefit from the method. Topics of further research include the optimization of *separable* spatially-variable filters, and feature representations other than edge detection.

6. REFERENCES

[1] Don P. Mitchell and Arun N. Netravali, "Reconstruction filters in computer graphics," *ACM SIGGRAPH*

Computer Graphics, vol. 22, no. 4, pp. 221–228, August 1988.

[2] Bryan S. Morse and Duane Schwartzwald, "Isophote-based interpolation," in *Proceedings of the IEEE International Conference on Image Processing*, 1998, vol. 3, pp. 227–231.

[3] Simon Baker and Takeo Kanade, "Limits on super-resolution and how to break them," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 9, pp. 1167–1183, September 2002.

[4] Kefu Xue, Ann Winans, and Eric Walowit, "An edge-restricted spatial interpolation algorithm," *Journal of Electronic Imaging*, vol. 1, no. 2, pp. 152–161, 1992.

[5] Jan Allebach and Ping Wah Wong, "Edge-directed interpolation," in *Proc. ICIP*, 1996, pp. 707–710.

[6] Kris Jensen and Dimitris Anastassiou, "Subpixel edge localization and the interpolation of still images," *IEEE Trans. on Image Processing*, vol. 4, no. 3, pp. 285–295, 1995.

[7] William T. Freeman, Thouis R. Jones, and Egon C. Pasztor, "Example-based super-resolution," *IEEE Computer Graphics and Applications*, vol. 22, no. 2, pp. 56–65, March/April 2002.

[8] John Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679–698, November 1986.

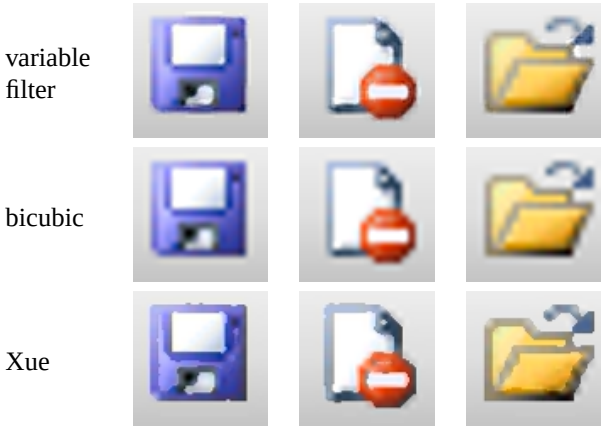

Fig. 2. $4\times$ magnification of icon images with various algorithms

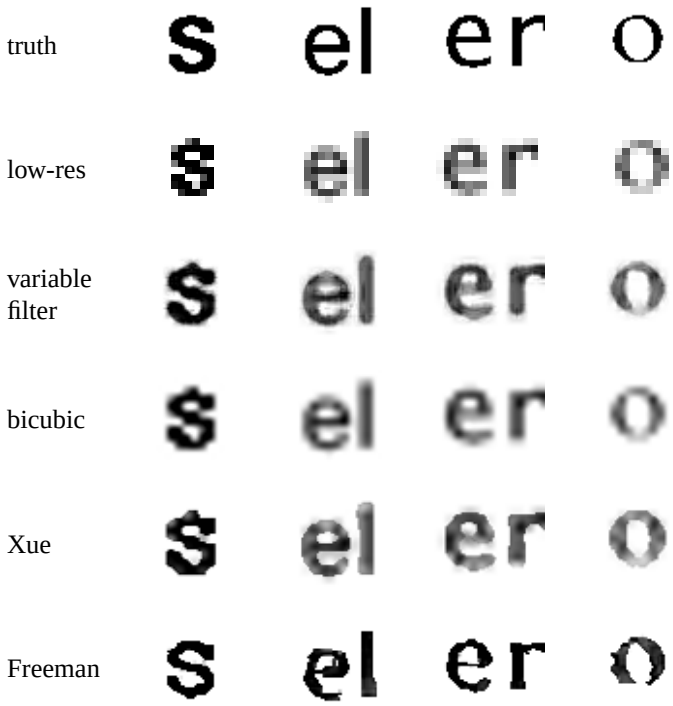

Fig. 3. $4\times$ magnification of glyph images with various algorithms

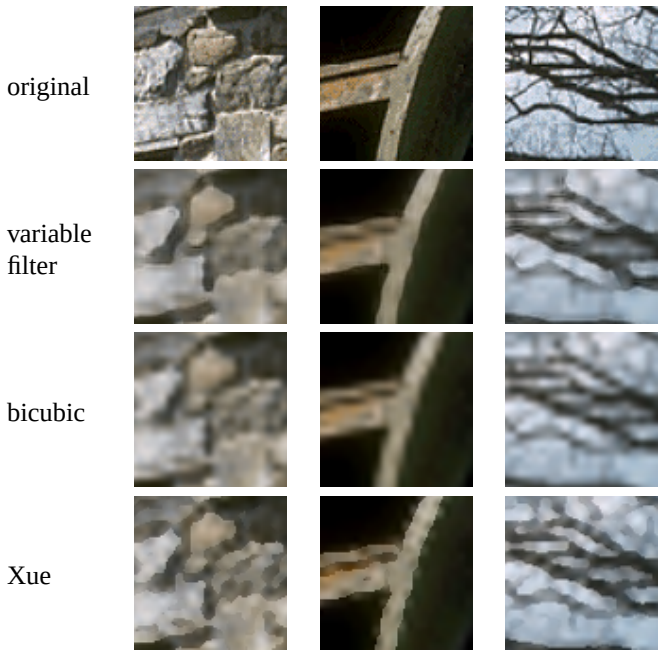

Fig. 4. $4\times$ magnification applied to a photograph subsampled with bicubic.