SINGLE-FRAME TEXT SUPER-RESOLUTION: A BAYESIAN APPROACH

Gerald Dalley[1,2], *Bill Freeman*[1,2], *Joe Marks*[1]

[1]Mitsubishi Electric Research Labs, Cambridge, MA marks@merl.com
[2]CSAIL, Massachusetts Institute of Technology, Cambridge, MA {dalleyg,billf}@csail.mit.edu

ABSTRACT

We address the problem of text super-resolution: given a single image of text scanned in at low resolution from a piece of paper, return the image that is mostly likely to be generated from a noiseless high-resolution scan of the same piece of paper. In doing so, we wish to: (1) avoid introducing artifacts in the high-resolution image such as blurry edges and rounded corners, (2) recover from quantization noise and grid-alignment effects that introduce errors in the low-resolution image, and (3) handle documents with very large glyph sets such as Japanese's Kanji. Applications for this technology include improving the display of: fax documents, low-resolution scans of archival documents, and low-resolution bitmapped fonts on high-resolution output devices.

1. INTRODUCTION

Some early work on fully automatic super-resolution of text was done by Ulichney and Troxel [1]. They use a fixed set of heuristic templates to model local continuous shape. While doing so, they verify that neighboring shape assignments are mutually compatible and make local straight-line approximations.

Several pieces of more-recent work on super-resolution [2, 3, 4] and compression [5] have centered on inferring shape by clustering glyphs seen in a document. For a document containing Latin characters, Hobby and Ho [3] generate nearly 300 glyph clusters. Ideally, each cluster contains all instances of a single glyph. During the clustering process, a high-resolution model of each glyph is generated by averaging polygonal shape approximations. Some care is taken to try to preserve sharp corners [2]. Bern and Goldberg expand on Hobby and Ho's approach by modeling the scanning process probabilistically and using slightly different algorithms at each step.

Thouin and Chang [6] use nonlinear optimization on a grayscale input image to minimize a Bimodal Smoothness Average (BSA) score. To be bimodal, the high-resolution image should be black and white with few gray pixels. To be considered smooth, its locally-estimated second derivatives should be small. For the "average" measure to be small, the average of high-resolution pixels should be close to the value of the corresponding pixel in the low-resolution image. This method is 1-2 orders of magnitude slower than cubic-spline interpolation, and works well in areas away from sharp corners and thin curves. Thouin, Du, and Chang also experimented with a Markov Random Field formulation, with results of similar quality [7].

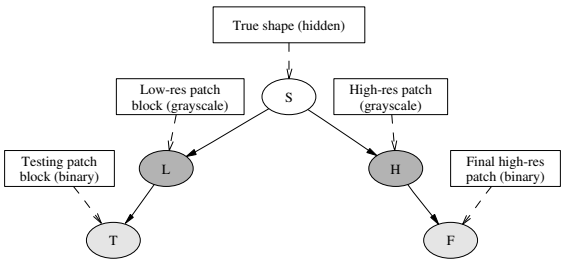

Fig. 1. Bayesian Network Model

Kim presents a time-efficient method that is extensible to large glyph sets at low resolution [8]. He begins by obtaining a training set consisting of a pair of registered black-and-white images at both low and high resolution. This pair of images is then scanned to build a database that indicates which high-res patch should be output given an input low-res patch. When using this database, any time an input low-res patch occurs that was not present in the training image, a weighted k-nearest-neighbor search is performed to estimate the best high-res output patch. Kim uses this approach for zooming noiseless 300dpi, 12pt. Times text to 600dpi, for zooming handwritten text, and for several nonzooming image-processing tasks. He also presents error bounds, confidence intervals, and inductive bias measures.

We employ a training-based method, similar to Kim, but adopt a full-Bayesian approach with an explicit noise model. To handle noise and large glyph sets while limiting the training set size, we utilize grayscale training images. A side effect of our approach is that we trade off training time so that super-resolution can be done via simple table lookup, without any nearest-neighbor searches at runtime. We perform most of our experiments with fax-resolution (200dpi) inputs.

0-7803-8554-3/04/$20.00 ©2004 IEEE.

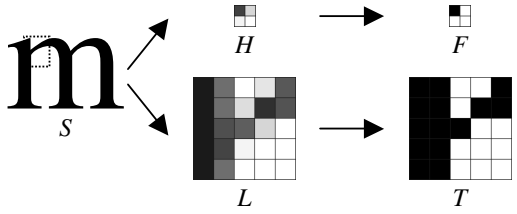

Fig. 2. Pictorial representation of our Bayesian framework. The patch of interest for the low-res images is highlighted with a dotted line on the large letter 'm'. The true shape of the patch is the hidden state S. It has been spatially quantized onto 2×2 high-res and 5×5 low-res image windows, resulting in H and L, respectively. Note that because the high- and low-resolution processes are independent when conditioned on S, the amount of area each window covers may be different, as shown in this figure. H and L are then quantized in intensity to produce the bilevel images F and T. The task is to infer the most likely F from T.

2. METHOD

We pose the problem in the Bayesian framework depicted in Figs. 1 and 2. This network is for a pair of image patches from a high-res and a low-res image. For example, in the low-res image we might be analyzing 5×5 patches to be able to produce a single pixel in the high-res image. S is the true shape of the image and is a hidden node, meaning that we never know and will not even try to estimate what that shape is. We will discuss the other nodes of the network in the following sections.

2.1. Training

During training, we are given pairs of grayscale low- and high-resolution image patches (L and H) that have been deterministically generated from the true glyph shape (S). As a simplification, we assume that when the underlying shape (S) is scanned to a binary image (named T for "test" or F for "final") during testing, the probability that a particular pixel will be quantized to be white is equal to the grayscale value of that pixel in the training image (L or H). We assume that the quantization of each pixel is independent of all the others. A more complete model might consider shape rotation, shape placement relative to the imaging grid, nonuniform placement of individual imaging sensors, and sensor noise.

To be more precise,

$$P(T|L=l) = \prod_{i=1}^{|L|} \left(l_i \delta(1,T_i) + (1-l_i)\,\delta(0,T_i)\right) \quad (1)$$

$$P(F|H=h) = \prod_{j=1}^{|F|} \left(h_j \delta(1,F_j) + (1-h_j)\,\delta(0,F_j)\right) \quad (2)$$

where $\delta(\cdot,\cdot)$ is the Kronecker delta function, $|L|$ is the number of pixels in a low-res patch, l_i is the grayscale value of patch l at pixel location i, T_i is the binary pixel value at location i, $|F|$ is the number of pixels in a high-res patch, h_j is the grayscale value of patch h at pixel location j, and F_j is the binary pixel value at location j. As indicated in Fig. 1, we assume that the quantization processes producing F and T are independent.

Ultimately, we wish to estimate $P(F|T)$:

$$P(F|T) = \int_S \frac{P(T,F|S)\cdot P(S)}{P(T)} \quad (3)$$

$$= \int_S \int_L \int_H \frac{P(T|L)P(F|H)P(L,H,S)}{P(T)} \quad (4)$$

Since rendering to grayscale is a deterministic process, the joint PDF of L, H, and S is a delta function. To estimate $P(F|T)$, we then need:

$$\hat{P}(F|T) = \frac{1}{N\cdot\hat{P}(T)} \sum_{s=1}^{N} P\Big(T|L=l^{(s)}\Big)\, P\Big(F|H=h^{(s)}\Big) \quad (5)$$

where s indexes the training samples, N is the number of training samples, $\left(l^{(s)}, h^{(s)}\right)$ is a training sample pair, and we estimate $P(T)$ as follows:

$$\hat{P}(T) = \frac{1}{N} \sum_{s=1}^{N} P\left(T|L=l^{(s)}\right) \quad (6)$$

2.2. Complexity Reduction

As stated, our training procedure is $O\left(|H|\cdot 2^{|L|}\right)$ in time and size. To reduce the time complexity, consider the case where $l_i = 0$. When training, any database entries corresponding to $t_i = 1$ need not be visited because $P(t|L = l) = 0$. In a typical training situation, most pixels are pure black or white, so only a very few of the 2^{25} entries need to be visited per training sample. Instead of enumerating all possible values of t, we can instead begin with the most likely t by thresholding L. We then recursively flip individual pixel values in t until $P(t|L = l) \approx 0$. If we only allow up to n pixels to be flipped (with respect to the most likely t) and $n \leq \frac{1}{2}|L|$, our worst-case time complexity reduces to $O\left({}_{|L|}C_n \cdot |H|\right)$ per training sample.

If we employ the preceding approximation, we find that most of the possible low-res patches are never processed in our training sets. In our largest database, 87% of the possible database entries have $\hat{P}(T) = 0$ and thus could be eliminated by using a more space-efficient data structure than a

large linear array. Additionally, by taking advantage of the independence of pixels in a high-resolution patch and partitioning the database, we can reduce the amount of RAM required to $O\left(2^{|L|}\right)$ [9].

2.3. Super-Resolution

Once the training database has been compiled, the basic super-resolution process becomes simple. The most likely binary pixel values in the output image are chosen using the local $\hat{P}(F|T)$ estimate computed from the corresponding low-res patch. To improve the quality of the results, $\hat{P}(F|T)$ may be output instead, and treating the probabilities as gray-scale values, image-processing operations such as smoothing and sharpening may be performed before creating the final bilevel output image.

To handle cases where the test image contains patches that are very different from what was seen in the training set, we create another super-resolution image by simply replicating pixels in the low-resolution image. We then take a weighted average of the $\hat{P}(F|T)$ image and this image. In our experiments, we set the weight of the pixel-replicated image to 0.01. The weight of each pixel in the $\hat{P}(F|T)$ image is $\hat{P}(T) \cdot N$, where $\hat{P}(T) \cdot N \gg 0.01$ for nearly all pixels in a typical image.

3. EXPERIMENTS

3.1. Training

For our training, we used a total of six synthetic image sets, all with paired 200dpi and 400dpi images. Each image in the datasets named *qc12* and *qc* contains a single instance of 93 glyphs that can be typed on a standard US keyboard. Multiple images with slight x- and/or y- offsets are used to introduce grid-quantization effects. The *qc12* glyphs are all 12pt., Bitstream Cyberbit. *qc* also includes 8, 10, and 14pt glyphs.

ec and *et* both contain images of a 1-page English document in order to simulate the effects of matching the character occurrence frequency. They use 12pt., Cyberbit and Times New Roman fonts, respectively.

ia and *ic* are similar to *qc12*, except they use 20,992 glyphs from the Unicode Chinese, Japanese, and Korean (CJK) Ideographs range (U+4E00–U+9FA5). *ia* uses the Arial Unicode MS typeface while *ic* uses the Cyberbit face.

Using these datasets, we trained databases for use with Kim's method as well as ours. For both, we used 5×5 low-res patches and 2×2 high-res patches. See [9] for pointers to the online datasets and more comprehensive results.

3.2. Testing

For qualitative testing we took actual scans of an English document. For our quantitative tests, we used a pessimistic, but reproducible scheme. We took the high-res text renderings and box-filtered, downsampled, gamma-corrected, and Gaussian-blurred (radius$= 3$, $\sigma = 0.5$) them. We then did a biased coin flip on each pixel to decide whether it should be white or black. The bias is decided by the blurred image's gray value. We then super-resolved the corrupted images to produce grayscale estimates of the original. For an error metric, we use the mean-squared error of the output images compared to the original high-res rendering.

The *ec200* image is the first page of a 200dpi rendering of an English document using 12pt. Bitstream Cyberbit glyphs (1628×2131 pixels at low resolution). *ec150* and *ec300* are 150dpi and 300dpi renderings of the same document, respectively. *jc200* is a 200dpi rendering of the first page of a Japanese document written in Kanji using 12pt. Bitstream Cyberbit glyphs. *ja200* and *js200* use the same document as *jc200*, but with the Arial Unicode MS and SimSun typefaces instead. Note that Arial, SimSun, and Cyberbit have different letter and line spacings, so these three images do not all contain the same numbers of glyphs.

3.3. Results

In Fig. 4, we show several examples of super-resolution output. The top four rows are from images scanned in from paper. In the *sepa* images, the Bayesian method is able to better recover from stray pixels at the bottom of the letter 'p' and missing pixels on the left edge of its descender. In the scanned Kanji images, our method is also better able to recover from noise in the bottom line of the left character and the middle portion of the right character. The bottom row of the figure shows typical results for the much noisier synthetic data we used. Note that the blocky regions in the Bayesian results are indicative of regions where the low-weight pixel-replicated estimate dominates.

Fig. 3 summarizes the results from our quantitative experiments. In nearly all experiments, the Bayesian method yields lower error rates than Kim's or pixel-replication, especially when the training and test glyphs are similar. In the *ja200+ic* test, we perform slightly worse than pixel-replication because the Arial font in the test image has a much higher stroke weight than our Cyberbit training data. For this test however, Kim's method yields much higher errors. The *jc200+ec* and *jc200+et* experiments represent using completely different glyph sets in training and testing (Latin vs. CJK ideographs). In this situation, both Kim's and the Bayesian method improve significantly upon pixel replication, though Kim's method is marginally better in MSE terms. In all other cases, the Bayesian method produces lower errors. We also observed a $2\times$ to $5\times$ speedup in

Test Image	Training Set	Pixel-Repl.	Kim's Method	Bayesian Method
ec200	*ia*	0.0641	0.0654 (-2%)	**0.0587** (8%)
ec200	*ic*	0.0641	0.0726 (-13%)	**0.0611** (4%)
ec200	*ec*	0.0641	0.0582 (9%)	**0.0522** (18%)
ec200	*et*	0.0641	0.0594 (7%)	**0.0551** (13%)
ec200	*qc*	0.0641	0.0592 (7%)	**0.0524** (18%)
ec200	*qc12*	0.0641	0.0592 (7%)	**0.0523** (18%)
ec150	*ec*	0.0795	0.0677 (14%)	**0.0652** (18%)
ec300	*ec*	0.0422	0.0383 (9%)	**0.0342** (19%)
ja200	*ia*	0.0409	0.0411 (0%)	**0.0357** (12%)
ja200	*ic*	**0.0409**	0.0488 (-19%)	0.0413 (0%)
jc200	*ia*	0.0460	0.0447 (2%)	**0.0414** (9%)
jc200	*ic*	0.0460	0.0467 (-1%)	**0.0389** (15%)
jc200	*ec*	0.0460	**0.0396** (13%)	0.0402 (12%)
jc200	*et*	0.0460	**0.0407** (11%)	0.0408 (11%)
jc200	*qc*	0.0460	0.0412 (10%)	**0.0386** (16%)
jc200	*qc12*	0.0460	0.0412 (10%)	**0.0386** (16%)
js200	*ia*	0.0532	0.0514 (3%)	**0.0479** (9%)
js200	*ic*	0.0532	0.0540 (-1%)	**0.0452** (15%)

Fig. 3. **Results Summary:** We report the mean-squared error (see Section 3.2) for each super-resolution method. For reference purposes, we also include the MSE for simple pixel-replication. The numbers in parentheses are the percentage error reduction relative to pixel-replication. The best result in each row is shown in **bold**. Please refer to Section 3 for a description of the test images and training sets.

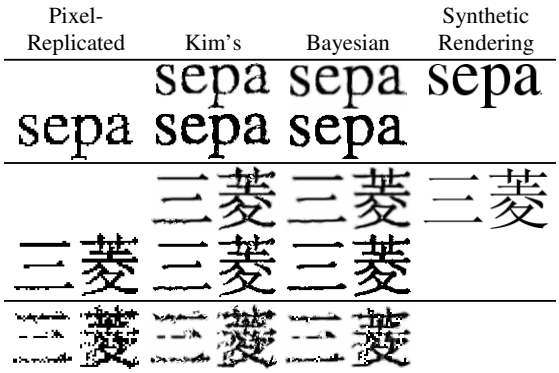

Fig. 4. **Selected Results:** These are small clippings from three tests. We used Kim's method and our Bayesian method for super-resolution, with 5×5 low-res patches and non-overlapping 2×2 high-res patches. The input to the first two rows is from an English document, using 12pt. Times New Roman, scanned in at 200dpi. The next two rows are from a Japanese (Kanji) document, using 12pt. SimSun, also scanned at 200dpi. The final row is the same as the previous pair, except the data were generated by rendering the image with synthetic noise and the Bitstream Cyberbit typeface was used. The bilevel images (2^{nd} and 4^{th} rows) were created by manually selecting a threshold on the corresponding grayscale images.

computing time for super-resolution versus Kim's method.

4. CONCLUSIONS

We have developed a full-Bayesian formulation to the task of performing super-resolution on binary text images. This method uses synthetic grayscale training data and an explicit noise model to limit the training set size in the face of noise and very large glyph sets. Our results show improvements in both English and Kanji test data in low-noise qualitative tests of real data as well as quantitative high-noise synthetic tests.

5. REFERENCES

[1] Robert A. Ulichney and Donald E. Troxel, "Scaling binary images with the telescoping template," *PAMI*, vol. 4, no. 3, pp. 331–335, 1982.

[2] John D. Hobby and Henry S. Baird, "Degraded character image restoration," in *Proceedings of the Fifth Annual Symposium on Document Analysis and Image Retrieval*, 1996, pp. 233–245.

[3] J. Hobby and T. K. Ho, "Enhancing degraded document images via bitmap clustering and averaging," in *Proc. of the 4th Int'l Conference on Document Analysis and Recognition*, Ulm, Germany, August 1997, pp. 394–400.

[4] Marshall Bern and David Goldberg, "Scanner-model-based document image improvement," in *International Conference on Image Processing*. IEEE, Sept. 2000, vol. 2, pp. 582–585.

[5] Peter B. Mark and Stuart M. Shieber, "Method and apparatus for compression of images," United States Patent 5,303,313, April 12 1994.

[6] Paul D. Thouin and Chein-I Chang, "A method for restoration of low-resolution document images," in *International Journal on Document Analysis and Recognition*. 2000, number 2, pp. 200–210, Springer-Verlag.

[7] Paul D. Thouin, Yingzi Du, and Chein-I Chang, "Low resolution expansion of gray scale text images using gibbs-markov random field model," in *Symposium on Document Image Understanding Technology*, Sheraton Columbia Hotel, Columbia, Maryland, April 2001.

[8] Hae Yong Kim, "Binary operator design by k-nearest neighbor learning with application to image resolution increasing," in *International Journal Imaging Systems and Technology*, 2000, vol. 11, pp. 331–339.

[9] Gerald Dalley, Bill Freeman, and Joe Marks, "Single-frame text super-resolution: A bayesian approach," Technical Report TR2004-059, Mitsubishi Electric Research Labs, 2004.