

Figure 2. The most intuitive methods to solve the super-resolution problem on an icon 2(a) could be to use a lookup table, as in the scalex method first methods 2(b), or use a threshold on a bicubic interpolation 2(c). Note: the threshold has been manually performed under Photoshop by giving a substantial hint on the colors to threshold, as well as on the zones that should be considered as gradients or uniform.

The super-resolution method we present allows for magnification of computer icons by any factor. It does not require multiple views or a database of training examples. This work lays the basis for an easier post-processing of GUIs and fonts for the visually impaired. For example, icon vectorization can be used for better color conversions (e.g., increasing the color variance), more accurate magnification (e.g., on normal or resolution-independent display), and emphasized high-frequency features.

2. Devised Method

One could argue that for such artificial images as bitmap icons, a bicubic interpolation followed by an efficient thresholding would be sufficient (cf Figure 2(c)). This efficiency would need to take into account the way those icons are created and therefore perceived. Icon artists use software that involves two windows: one with the nearest-neighbor magnified icon, and another one with the icon at its final size. The modus operandi is also fairly simple: the zoomed pixels of the first window are modified one by one until the result in the second window is satisfactory: clear, unambiguous and free of noise. Our goal is therefore to clarify those perception cues, tacitly used in the icon artwork, but well-known in psychophysics and vision.

Our approach first focuses (in Section 3) on characterizing the interactions between the pixels in the low-resolution icon. The method we propose is based on simple local models of color perception which are globalized using a Markov Random Field model. The MRF determines how the human visual system links or unlinks pixels, while computing the perceived color of each pixel at the same time.

In Section 4) we consider the problem of generating the icon at any higher resolution. For this purpose, we define regions of influence around each low-resolution pixel. Using the linkage and the perceived colors we have computed through the MRF, the boundaries of those regions as well their color content can be defined. Active contours are finally smooth those boundaries while keeping the strength of the linkage/separation between those regions.

Experimental results are presented in Section 5, and Section 6 discusses future work and provides concluding remarks.

3. Modeling Local Low-resolution Pixel Relationships

Examining the pixel interactions in the low-resolution (input) icon is a necessary step before interpolating it to a larger size and we propose a Markov Random Field (MRF) model to make the interactions between the icon pixels explicit. We will first present the structure of our MRF in Section 3.1. Then, using perception models, we will make explicit its different potential functions in Sections 3.2 and 3.3. Finally, we will solve it in Section 3.4.

Let us formulate the problem as follows: for a given pixel color Y_i , we would like to determine the perceived color X_i . A complete model should take into account the *simultaneous lightness/color contrast* [23], the *lightness/chromatic adaptation* [22] and some geometric effects (e.g. spatial frequencies [2, 13]). We propose to operationalize these factors by considering the following parameters that influence the perceived color X_i s.

- the color of the pixel Y_i and the colors of its 8 surrounding pixels (Figure 3(a))
- X_i 's relationship to each of its 8 neighbors, which can be *uniform* (their colors are perceived to be the same), *gradient* (the colors are perceived as closed but different), or *unlinked* (the colors are perceived as too different) (Figure 3(b))
- The weight of the linkage between X_i and its neighbors, which is a notion described in Figure 3(c)

Figure 4 depicts, the different factors influencing the perceived color for one node. The hidden nodes $H_{i \rightarrow j}$ characterize the type of link between X_i and X_j (uniform, gradient or unlinked) and its weight, which can be different for $H_{i \rightarrow j}$ versus $H_{j \rightarrow i}$. As we will see in Section 3.4, the strength of the linkage can be inferred only from the X_i s and their types of linkage. This allows us to use the simpler MRF shown in Figure 5 in which $H_{i,j}$ is now a discrete variable indicating only the type of link.

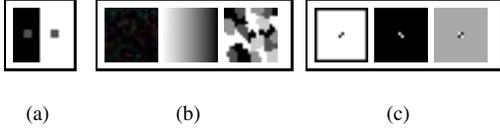


Figure 3. 3(a) Due to differences in their surrounding neighborhood, the two central squares are perceived as lighter or darker while they actually are the same gray. 3(b) Left: The square looks uniformly gray despite containing 1% additive Gaussian noise. Center: The image appears smooth while it is formed by a discrete staircase. Right: even though the zones are small, the 4 grays are different enough to be perceived distinctly. 3(c) All three images contain the same 2×2 central checkerboard block. Left: We perceive a black form elongated at 45° (which can be a line, ellipse, or peanut shape depending on one's interpretation). Center: With a black surrounding region we perceive a white form elongated at 135° . Right: With a gray surrounding region, neither organization is preferred.

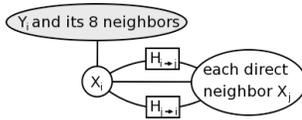


Figure 4. MRF for inferring X_i , the perceived color of the i th pixel.

3.1. Joint distributions of our model

In this work, we adopt the convention of using uppercase to refer to random variables and lowercase to their occurrence. The joint distribution of an MRF can be written as follows:

$$p(x) = \frac{1}{Z} \prod_{C \in \mathcal{C}} \psi_C(x_C)$$

where \mathcal{C} is the set of maximal cliques in the graph, $\psi_C(x_C)$ is a potential function on the clique C , and Z is the normalizing constant. If we look at Figure 5, we are presented with two different kinds of maximal cliques:

- As we are working in 8-connectivity, we have a maximum clique linking four X_i s forming a square. Therefore, for a given X_i , we have four different such cliques. We denote them $C_4^j(X_i)$, $1 \leq j \leq 4$, and their corresponding potential function ψ_4 .
- Another kind of maximal clique is the one linking two neighboring X_i and X_j with their $H_{i,j}$. We now have eight of those cliques for each X_i ; we use the notation $C_3^j(X_i)$, $1 \leq j \leq 8$ and ψ_3 for its potential function.

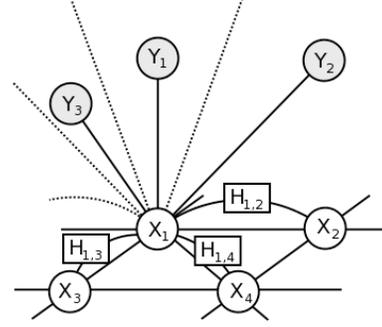


Figure 5. MRF model chosen to find the X_i s.

We denote by n_i the set formed by the pixel i and its eight neighbors, and by $y(n_i)$ the actual colors of the pixels in n_i . We can therefore rewrite the MRF as:

$$p(x) \propto \prod_{x_i} \left(\phi(x_i; y(n_i)) \prod_{j=1}^4 \psi_4(C_4^j(x_i)) \prod_{j=1}^8 \psi_3(C_3^j(x_i)) \right)$$

As edges are common to some cliques, we instead write the joint distribution in the following way:

$$p(x) \propto \prod_{x_i} \left(\phi(x_i; y(n_i)) \prod_{j=1}^8 \psi(x_i, x_j, h_{i,j}) \right)$$

This formulation is more intuitive as the ϕ s are responsible for the luminance and hue effects as shown in Figure 3(a), and the ψ s are responsible for spatial interactions between pixels as shown in Figure 3(b). We describe the structure of these two functions in the following two sections.

3.2. Color Perception

As seen in the first two pictures in Figure 3(b), the neighbors of a pixel influence its perceived color in two ways:

1. **Lightness Induction.** Lightness induction consists of two effects. *Lightness assimilation* where an almost black pixel surrounded by pure black pixels will be perceived as completely black and *simultaneous lightness contrast* where a light gray pixel surrounded by black pixels will be perceived as lighter [27].
2. **Hue Induction.** Similar to lightness induction, hue induction also consists of two effects. The key difference here is that while lightness induction effects only depend on the luminance of the neighborhood, hue induction effects depend on the hue as well as the lightness of the neighboring pixels. In the case of *Hue assimilation* the perceived hue of a pixel surrounded by pixels with similar hue is shifted toward the average

hue of the neighborhood. *Simultaneous Hue Contrast*, on the other hand, shifts the perceived hue of a pixel towards the complement color of its neighborhood when it is surrounded by pixels of contrasting hue.

While easy to describe, these effects are quite hard to implement in a quantitative manner. Over the years, a number of models have been developed which capture these effects to varying degrees of accuracy [8, 19]. Since our interest is restricted to icons which will be displayed on a computer screen, an approximate estimate of these effects will suffice. Our approximation is inspired by the work on *Equivalent Background* [7, 16]. In the following analysis, we will use the CIELab color space [25]. L stands for luminance and has values in $[0, 255]$, a is the red-green channel with values in $[-128, 127]$, and b is the yellow-blue channel with values in the same range.

We will now describe the function $\phi(x_i; y(n_i))$. Given a pixel i with actual color y_i , we begin by estimating the “shift vector” y'_i such that the distance of the perceived color x_i from the line segment $y_i y'_i$ is distributed normally. y'_i represents the color direction towards which y_i is shifted when perceived and is computed as following. For the j^{th} pixel with color y_j in the neighborhood of the pixel i we compute

$$L_j^{shift} = \begin{cases} 0 & \text{if } |L_i - L_j| \leq L_{t1} \\ L_j - L_i & \text{if } L_{t1} < |L_i - L_j| \leq L_{t2} \\ L_i - L_j & \text{if } L_{t2} < |L_i - L_j| \end{cases}$$

We similarly define a_j^{shift} and b_j^{shift} . The components of the color vector $y'_i = (L'_i, a'_i, b'_i)$ are then defined as

$$L'_i = L_i + \mu_L \frac{1}{\#n_i} \sum_j L_j^{shift}$$

and similarly for a'_i and b'_i . The constants $L_{t1}, L_{t2}, a_{t1}, a_{t2}, b_{t1}, b_{t2}, \mu_L, \mu_a$ and μ_b are empirically determined quantities, independent of the image being considered.

3.3. Spatial Interaction

As mentioned earlier, we consider three kinds of pairwise interactions. The spatial interaction term $\psi(x_i, x_j, h_{i,j})$ can now be defined for the three cases as follows:

1. **Uniform.** In this case the distance in color space between the perceived colors x_i and x_j is small enough for them to be considered the same, i.e., $\|x_i - x_j\| \leq \lambda_0$. Psychophysics tests indicate that the human eye cannot discern the difference between two colors which are less than one unit distance apart in the

CIELab color space [14]. To account for the error due to the finite size of each pixel and brightness variations across display screens, we allow for a higher value of λ_0 ($=20$). The spatial interaction term $\psi(x_i, x_j, h_{i,j} = \text{uniform})$ is then defined as

$$\begin{cases} \mathcal{N}(\|x_i - x_j\|; 0, \sigma_0^2) & \text{if } \|x_i - x_j\| \leq \lambda_0 \\ 0 & \text{otherwise} \end{cases}$$

2. **Gradient.** This is the case when the color change as we move from pixel i to pixel j is smooth enough for it to appear continuous. This is the case when $\lambda_1 \leq \|x_i - x_j\| \leq \lambda_2$. Note that the interval $[\lambda_1, \lambda_2]$, that we took as $[15; 35]$ may overlap with the interval $[0, \lambda_0]$ as there is no sharp boundary between the two phenomena. The corresponding spatial interaction $\psi(x_i, x_j, h_{i,j} = \text{gradient})$ term is given by

$$\begin{cases} \mathcal{N}(\|x_i - x_j\|; \mu_1, \sigma_1^2) & \text{if } \lambda_1 \leq \|x_i - x_j\| \leq \mu_1 \\ 1 & \text{if } \mu_1 \leq \|x_i - x_j\| \leq \mu_2 \\ \mathcal{N}(\|x_i - x_j\|; \mu_2, \sigma_2^2) & \text{if } \mu_2 < \|x_i - x_j\| \leq \lambda_2 \\ 0 & \text{otherwise} \end{cases}$$

3. **Unlinked.** There is the case when there is no spatial interaction between pixels i and j because their perceived colors are too far apart. The joint potential function therefore is independent of their color and constant:

$$\psi(x_i, x_j, h_{i,j} = \text{unlinked}) = 1$$

3.4. Solving the MRF

In order to improve the convergence and accelerate the solution of the MRF, we attempt to restrict the number of possible values for each $H_{i,j}$. For each pair of neighboring pixels i and j , we compute d_{min} and d_{max} , the minimum and maximum distance between the perceived colors x_i and x_j such that both $\phi(x_i; y(n_i))$ and $\phi(x_j; y(n_j))$ are above a certain threshold. We then check for the intersection between the intervals $[d_{min}, d_{max}]$ with $[0, \lambda_0]$ and $[\lambda_1, \lambda_2]$, which correspond to the pixels i and j being part of a uniform or a gradient region respectively. In most cases the number of possible values for $H_{i,j}$ is reduced from 3 to 1 or 2. Next, we initialize the x_i values by: $x_i^{ini} = \frac{1}{2}(y_i + y'_i)$, which is the expected value of the potential function $\phi(x_i; y(n_i))$.

We tried several methods of solution for the MRF: belief propagation [30], simulated annealing [18], and Metropolis sampling [21]. We found empirically that there were few local minima, each yielding perceptually very similar results. This arises from the simplicity of our model and the limited number of possible perceived colors for a pixel. In all of our experiments, we used simulated annealing.

4. Constructing the High-resolution image

As our ultimate goal is to obtain a zoomed version of an icon at any size, we need an appropriate descriptor that is scale independent, i.e. vectorized. So far, the method of Section 3 provides us with the perceived colors of the pixels as well as the interactions at stake. Therefore, the inside of a uniform/gradient region formed by linked pixels can be determined by basic interpolation. The only problem is to determine, in a scale independent way, the boundary between unlinked pixels.

To determine these boundaries we build a “zone of influence” around each pixel, defined by the interactions between a pixel and its neighbors. Then, a snake smoothes these boundaries while trying to stay as much as possible under equal influence of the surrounding pixels. This snake representation is necessary for the following two reasons:

- it enables a vectorized parametrization.
- it restores the interaction between pixels as shown in Figure 6.

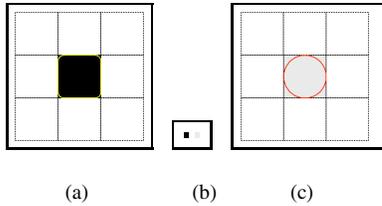


Figure 6. The boundary between a black pixel and its white neighbors is better defined than between a lighter pixel and its white neighbors 6(b). Therefore, the snake in 6(a) sticks more to the boundary than in 6(c) where the smoothness is more enforced.

4.1. The influence zones

As the boundaries of the influence zones are smoothed by the snake, we can afford to use a simple model. We choose to consider the following model of influence: a low resolution pixel p_i influences the points with local polar coordinates (r, θ) with Gaussian intensity:

$$I_i(r, \theta) = \exp\left(-\frac{r^2}{2 \cdot \sigma(\theta)}\right)$$

$\sigma(\theta)$ is computed in the following way:

- find out p_j , the direct neighbor of p_i such that the segment linking the centers of p_i and p_j is closest to (r, θ)
- define $\sigma(\theta)$ as:

$$\sigma(\theta) = \beta \frac{\|(L_i, a_i, b_i)^\top - (L_j, a_j, b_j)^\top\|}{1 + \#\{p_k \text{ such that } p_k \text{ and } p_i \text{ are linked}\}}$$

where $(L_i, a_i, b_i)^\top$ is the color of the pixel p_i and β is a normalization factor (we took $\beta = 50$ as it gives values of σ fulfilling the later requirements).

This form of σ obviously takes 8 values (one for each neighbor). We therefore linearly smooth $\sigma(\theta)$ to have a continuous influence $I_i(r, \theta)$.

Let us consider the case of a point right between two low-resolution pixels. If σ is small for both, then the two influences will be small for the considered point: it will therefore be easy for the snake to float around if it wants an equal influence. On the contrary, if σ is high, the snake will be stuck in a valley and will therefore not be able to bend a lot.

The form of $\sigma(\theta)$ is justified by the two effects we want to reproduce:

- the connection strength shown in Figure 3(c) which is inversely influenced by the number of connected pixels.
- the intensity of the color difference between unlinked pixels, as shown in Figure 6.

It is easy to compute analytically the boundaries of equal influence between unlinked pixels but, in the meantime, the obtained curve is jagged. That is why we need to smooth it while conserving the connection to the data: active contours [17] help us for this purpose.

4.2. Vectorial snake

As mentioned before, our first step is to compute analytically the boundaries between unlinked pixels. These different boundaries are going to intersect each other at certain points (where 3 or 4 or unlinked pixels are equally influential) and/or begin at others. Those points can easily be found and they will form the endpoints of our snakes. Using a conventional snake, we find the best compromise between its smoothness and the influences of the pixels it separates. We compute the curve $\mathbf{v}(s)$ that verifies the following properties:

- $\mathbf{v}(0)$ and $\mathbf{v}(1)$ are the extremities of the snake we compute at first
- $s \mapsto \mathbf{v}(s)$ minimizes:

$$\int_0^1 A \left(\frac{d\mathbf{v}}{ds}\right)^2 + B \left(\frac{d^2\mathbf{v}}{ds^2}\right)^2 + |I_1(\mathbf{v}(s)) - I_2(\mathbf{v}(s))| ds$$

where I_1 and I_2 are the influences of the two low-resolution pixels the snake is trying to separate.

4.3. The case of the anti-aliasing zones

So far, during the whole process, we have tried to minimize the need for interpretation of specific structures, e.g., imposing that pixels aligned at 45° have to form a line.

There is one case, however, that needs special attention: anti-aliasing zones. They are one-pixel wide zones located between uniform/gradient zones. They are very local gradients incorporated into the icon design in order to make the edges appear less jaggy.

Until now, those zones have not been defined, though we have defined gradient and uniform zones. We can therefore deduce the anti-aliasing zones and treat them appropriately.

Each anti-aliasing pixel is located on the boundary between several (usually two) zones: its color is therefore a weighted mean of the neighboring “non anti-aliasing” pixels. The influence around it can therefore be represented by a mixture of distributions: one for each of the “non anti-aliasing” neighboring pixels.

5. Results

In Figure 7, we show our results on three different icons, resized to 5x. Each example required 4 minutes of computation in Matlab on a 2.8 GHz Pentium 4. The four columns show respectively: the original icon, the nearest neighbor interpolation, the cubic interpolation and our method. The snake gives normal results on the first two icons as they contain strong corners. In Figure 8, we show the importance of smoothing in our model: even if we can change its smoothness, the snake still gets stuck in some valley and therefore cares more about the attachment to the data than to the smoothness (this is especially noticeable with the eye of the penguin that remains sharp).

6. Conclusions and Future Work

In conclusion, we have addressed the problem of super-resolution for color bitmap icons. So far, the method provides encouraging results and we intend to improve the model by first incorporating the anti-aliasing connections in the MRF and by solving the original un-simplified MRF. The snake model also needs to be improved in order to catch discontinuities. Moreover, by the specificity of the problem, this method can not be used with natural images, but we could apply it to more specific problems, e.g. font recovery as shown in the promising result in Figure 6. Finally, we plan to make the parametrization of the icon, compatible with the SVG HTML [12] vector format or the Cairo graphics vector library [9] in order to have it available to a wide range of platforms and therefore applications.

Acknowledgements

This work was funded by the UCSD division of Calit2, the California Institute for Telecommunications and Information Technology, NSF-CAREER #0448615, DOE/LLNL contract no. W-7405-ENG-48 (subcontracts B542001 and B547328), and the Alfred P. Sloan Fellowship. The authors would also like to thank S. Agarwal, K. Branson, S. Dasgupta and A. Rabinovich for helpful discussions.



Figure 7. Results of our proposed method (last column): 1st row: the wheels are sharp on the edge and smooth inside. 2nd row: the Mozilla spikes are rendered, as is the shading, and the eyes and teeth look brighter, but the edges are noisy. 3rd row: the S appears as one connected component without holes, and the blue segments are well separated, but some noise appears on their boundaries



Figure 8. As a first step towards vectorization, the boundaries of the uniform zones have been smoothed with a spline. From left to right, the images are: the original icon, the nearest neighbor, the cubic interpolation, our interpolation without and with the spline

References

- [1] B. Bascle, A. Blake, and A. Zisserman. Motion deblurring and super-resolution from an image sequence. In *Proc. 4th Europ. Conf. Comput. Vision*, pages II:573–582, 1996.
- [2] F. Campbell and J. Kulikowski. Orientation selectivity of the human visual system. *J. Physiol.*, 187:437–445, 1966.
- [3] D. Capel and A. Zisserman. Super-resolution enhancement of text image sequences. In *Proc. Int'l. Conf. Pattern Recog.*, pages Vol I: 600–605, 2000.
- [4] D. Capel and A. Zisserman. Super-resolution from multiple views using learnt image models. In *Proc. IEEE Conf. Comput. Vision and Pattern Recognition*, pages II:627–634, 2001.
- [5] S. Chaudhuri. *Super-Resolution Imaging*. Kluwer, September 2001.
- [6] F. J. Cortijo, S. Villena, R. Molina, and A. Katsaggelos. Bayesian super-resolution of text image sequences from low resolution observations. *IEEE International Symposium on Signal Processing and its Applications*, 2003.
- [7] M. Fairchild. A victory for equivalent background – on average. In *IST/SID 7th Color Imaging Conference*, volume 82, pages 87–92, Scottsdale, 1999.
- [8] M. Fairchild and G. M. Johnson. Meet icam: A next-generation color appearance model. *Color Imaging Conference*, pages 33–38, 2002.
- [9] freedesktop.org. <http://www.freedesktop.org/Cairo/>.
- [10] W. Freeman, E. Pasztor, and O. Carmichael. Learning low-level vision. In *Proc. 7th Int'l. Conf. Computer Vision*, pages 1182–1189, 1999.
- [11] W. T. Freeman, T. R. Jones, and E. C. Pasztor. Example-based super-resolution. *IEEE Computer Graphics and Applications*, 2001.
- [12] S. V. Graphics. <http://www.w3.org/TR/SVG/>.
- [13] L. Harmon and B. Julesz. Masking in visual recognition : effects of two-dimensional filtered noise. *Science*, 180:1194–1197, 1973.
- [14] R. Hunt. *The Reproduction of Colour*. Wiley, 1967.
- [15] M. Irani and S. Peleg. Motion analysis for image enhancement: Resolution, occlusion, and transparency. *JVCIR*, 4, 1993.
- [16] G. Jando, T. Agostini, A. Galmonte, and N. Bruno. Measuring surface achromatic color: Toward a common measure for increments and decrements. In *Behavior Research Methods Instruments and Computers*, volume 35, pages 70–81, Scottsdale, 2003.
- [17] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *IJCV*, 1(4):321–331, January 1988.
- [18] S. Kirkpatrick, J. C.D. Gelatt, and M. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, May 1983.
- [19] E. Land and J. McCann. Lightness and retinex theory. *J. Opt. Soc. Am.*, 61(1):1–11, 1971.
- [20] A. Mazzoleni, 2001. <http://scale2x.sourceforge.net/index.html>.
- [21] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller. Equations of state calculations by fast computing machines. *Journal of Chemical Physics*, 21:1087–1091, 1953.
- [22] J. Mollon and P. Polden. Post-receptoral adaptation. *Vision Research*, 19:35–40, 1979.
- [23] S. Palmer. *Vision Science: Photons to Phenomenology*. MIT Press, 1999.
- [24] E. Peli, J. Kim, Y. Yitzhaky, R. Goldstein, and R. Woods. Wideband enhancement of television images for people with visual impairments. *JOSA-A*, 21(6):937–950, June 2004.
- [25] W. Poirson. Appearance of colored patterns: Pattern-color separability. *J. Opt. Soc. Am.*, 1993.
- [26] G. Ramanarayanan, K. Bala, and B. Walter. Feature-based textures. *Proc. of the Eurographics Symposium on Rendering*, pages 265–274, 2004.
- [27] R. Shapley and R. C. Reid. Contrast and assimilation in the perception of brightness. In *National Academy of Sciences of the USA*, volume 82, pages 5983–5986, 1985.
- [28] M. Stepin, 2001-2002. <http://www.hiend3d.com/smartfit.html>.
- [29] J. Tumblin and P. Choudhury. Bixels: Picture samples with sharp embedded boundaries. *Proc. of the Eurographics Symposium on Rendering*, pages 186–196, 2004.
- [30] J. Yedidia, W. Freeman, and Y. Weiss. Bethe free energy, kikuchi approximations, and belief propagation algorithms. In *SCTV01*, pages xx–yy, 2001.