An Adaptable k-Nearest Neighbors Algorithm for MMSE Image Interpolation

Citation: Ni, K.S., and T.Q. Nguyen. "An Adaptable k -Nearest Neighbors Algorithm for MMSE Image Interpolation." Image Processing, IEEE Transactions on 18.9 (2009): 1976-1987. © Copyright 2009 IEEE

As Published: http://dx.doi.org/10.1109/tip.2009.2023706

Publisher: Institute of Electrical and Electronics Engineers

Persistent URL: http://hdl.handle.net/1721.1/60036

Version: Final published version: final published article, as it appeared in a journal, conference proceedings, or other formally published context

An Adaptable k-Nearest Neighbors Algorithm for MMSE Image Interpolation

Karl S. Ni, *Member, IEEE*, and Truong Q. Nguyen, *Fellow, IEEE*

***Abstract*—We propose an image interpolation algorithm that is nonparametric and learning-based, primarily using an adaptive k-nearest neighbor algorithm with global considerations through Markov random fields. The empirical nature of the proposed algorithm ensures image results that are data-driven and, hence, reflect "real-world" images well, given enough training data. The proposed algorithm operates on a local window using a dynamic k-nearest neighbor algorithm, where k differs from pixel to pixel: small for test points with highly relevant neighbors and large otherwise. Based on the neighbors that the adaptable k provides and their corresponding relevance measures, a weighted minimum mean squared error solution determines implicitly defined filters specific to low-resolution image content without yielding to the limitations of insufficient training. Additionally, global optimization via single pass Markov approximations, similar to cited nearest neighbor algorithms, provides additional weighting for filter generation. The approach is justified in using a sufficient quantity of training per test point and takes advantage of image properties. For in-depth analysis, we compare to existing methods and draw parallels between intuitive concepts including classification and ideas introduced by other nearest neighbor algorithms by explaining manifolds in low and high dimensions.**

***Index Terms*—Classification, embedding, interpolation, nearest neighbor, superresolution.**

I. Introduction

IMAGE interpolation relates to methods of constructing new image detail from a discrete set of known points resulting in a high-resolution image. The problem is ill-posed, and the quality of the solution is usually considered subjectively, focusing on edges, texture, and clarity of content. Such properties can be generated in a number of ways, but to obtain them, new or assumed information must be introduced. The information can come in many forms, including but not limited to assumptions on pixel properties [1]–[3], frequency properties [4], [5], a set of low-resolution, shifted images [6]–[8], or a training set in statistical and machine learning [9]–[13]. In our work, it is the final category in the list, and the proposed algorithm is concerned with its application in a sliding-window approach.

A common interpolative technique (not particularly specific to images) is static filtering [14]–[17]. The cited approaches operate on the incorrect assumption (which is widely acknowledged as such due to the diversity of image content) that relationships between local low and high-resolution content can be described by a single convolutional kernel. Rather, their complexity usually justifies their usage, and though numerical results are most likely incorrect, the visual quality does not overtly reflect it. More generally, the human visual system (HVS) is forgiving of estimation errors from reasonable linear, spatial domain filters, meaning that in terms of human perception, the actual process is *approximated* fairly well. This observation is fundamental to the proposed algorithm. In light of using content specific linear filtering devices, when errors do occur, the damage of estimation errors due to insufficient training appears perceptually mitigated.

Instead of a static interpolation process for all image content, a logical improvement [1], [3], [9], [13], [18] would be to adapt filters that are optimized for certain content. These algorithms balance specificity with estimation error by benefitting from the knowledge that for localized linear approximations, reasonable solutions will not significantly distort the image. We are interested in adapting optimal filters that are chosen from a training set by variants on the k-Nearest Neighbor (k-NN) algorithm.

Nearest neighbor algorithms are attractive because they are easy to implement, nonparametric, and learning-based. Such attributes contribute to image results that reflect natural images very well should the training set be well-chosen. The use of nearest neighbor algorithms [10], [11] for interpolation is not new. Because [10] and [11] are limited to information from a single neighbor, the details of which will be further reviewed in Section II, we propose to use a combination of information gathered from k neighbors. The hope is to build the capability of adapting and scaling to different training sets while simultaneously improving image quality according to training set size.

k-NN determines k training points that are closely related to an input vector through an appropriate similarity metric. For image interpolation, once the relevant training samples are found, filters are specially tailored to determine high-resolution values after identifying low-resolution content. The crux is to achieve specificity with regard to image content without any loss of generalization of application. That is, how detailed can we make an image look while still making it look good for any other image?

The answer to this question is intimately related to the quantity of training samples used per reconstruction filter. In images,

Manuscript received October 17, 2008; revised April 24, 2009. First published May 26, 2009; current version published August 14, 2009. This work was supported in part by a grant from Qualcomm, Inc. This work was originally submitted while K. Ni was with the University of California at San Diego. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Dimitri Van De Ville.

K. Ni was with the University of California at San Diego, La Jolla, CA 92093 USA. He is now with Massachusetts Institute of Technology Lincoln Laboratories, Lexington, MA 02421 USA (e-mail: karl_ni@cal.berkeley.edu).

T. Q. Nguyen is with the Video Processing Laboratory, Electrical and Computer Engineering Department, University of California at San Diego, La Jolla, CA 92093 USA (e-mail: nguyent@ece.ucsd.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2009.2023706

more training points per filter, i.e., k large, equals more generality, meaning that errors and variations due to the training set are diminished. Alternatively, fewer training points per filter, i.e., k small, equals more specificity, meaning that the image reconstruction is clearer and more detailed. As will be explained in later sections, these issues naturally depend on the size and quality of the overall training set, but it is reasonable to conclude that to accommodate all possible test points, k must be variable.

This work proposes an adaptable k for the k-NN algorithm with special application to image interpolation. The remainder of this paper explores the potential of the algorithm and comparisons to related approaches. Section II reviews previously proposed algorithms in superresolution, interpolation, and statistics in Section II-A as well as basic concepts in Section II-B. Section III provides the major details for the proposed algorithm, describing a variable k in k-NN used for superresolution. Then, Section IV describes a weighted, single-pass Markovian-like network that considers neighboring patches. Finally, Section V analyzes image results, compares to state of the art techniques, and discusses implications of our use of k-NN in image interpolation.

II. Recent Work and Background

It is well known from image restoration theory that image interpolation is an ill-posed problem. Aside from typical spline methods, several approaches interpolate in other domains, including the Discrete Cosine Transform (DCT) domain [4], [5], Discrete Wavelet Transform (DWT) domain [19], [20], and Fourier Domain [21], [22]. These methods perform some type of zero-padding in higher frequency slots, which by taking the inverse transform, results in a spatially larger image. However, instead of interpolation the result contains more characteristics of re-*scaling*, where higher resolution information is not added and edges and texture are not elucidated.

Algorithms that concentrate on particular image attributes often preserve some type of regularity [23]–[25] including a measure that is tailored specifically to edges [1], [2]. In particular, [24] uses properties in the decay of wavelet coefficients to predict unknown coefficients at higher resolution subbands. Meanwhile, [1] uses a low-resolution correlation matrix as an approximation to obtain a high-resolution image filter based on an assumption of geometric duality. Although simple to implement, the covariance matrix is still low-resolution, and the value added is usually inadequate for complicated textures, often causing an effect similar to aliasing. In fact, in terms of generating resolution, all of these algorithms are inferior to learning-based methods where the quantity of additional information may be exceedingly large so that the method is able to enhance all types of image content.

A. Statistical Learning Methods for Superresolution

In learning-based methods, high-resolution information is provided *a priori* in the form of a training set, where relationships are inferred to generalize to unknown data points. Previously submitted contributions in the area involve highly nonlinear approaches [26]–[30]. Although results are very good, the complexity involved in these papers cause interpolation to be unwieldy, and it seems unnecessary to involve the heavy machinery associated with the support vector regression (SVR), the central development in the approaches. Another statistical learning algorithm involves outdated neural network concepts [31], [32] with limited degrees of success in interpolation in terms of visual quality.

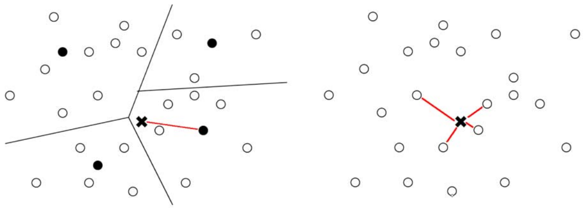

(a) Segmented domain for 2-D low-res patch in classification (b) Representation of domain for 2-D low-resolution patch in k-NN

Fig. 1. Classification-based image interpolation (a) divides the domain into several sections, and builds a representative filter at the location of the filled in circles for each of the classes. k-NN-based image interpolation (b) forgoes the classification during training and grabs the k nearest neighbors, in this case $k = 4$. With a similarity metric shown in red and the test point depicted as $\mathbf{x}$, the black dot, or the mean/centroid of a cluster, in (a) is not as related to the test point as the four nearest training points in (b). If image clusters are unknown, which they typically are, then k-NN creates much better filters for test points that border on boundaries.

The most successful algorithms[1] include classification-based adaptive filtering in [9], globalized nearest neighbor in [10], and k-NN regression with novel weighting vectors in [11]. In [9], the problem is formulated by using filters, chosen with local spatial information as the feature space. Such filters provide content-based interpolation under the assumption that image patches are distributed as Gaussian mixtures, the parameters of which can be found through the expectation maximization (EM) algorithm [33]. Classification-based reconstruction methods are partitioned according to image content, and problem complexity decreases per test point, which yields good performance. Because the domain is partitioned into several different classes, the idea of content based filtering can be improved by using more immediate points that are more relevant (seen in Fig. 1).

William Freeman's example-based superresolution [10], which draws from candidate nearest neighbors and chooses the best neighbor by a Markov network, is a good instance of nearest neighbor interpolation. Instead of a single nearest neighbor, which depending on the training set may not be sufficient, manifold learning through neighbor embedding in [11] offers a way to consider k neighbors, incorporating more information to the solution. The method tends to have an easier time coming up with local representations, but recent developments [34] show that the underlying assumption in [11] of isometry, at least for Euclidean distances, between low-resolution neighbors and high-resolution neighbors is inherently false. In other words, the distance metric used by [11] in low-resolution does not correspond equally to their high-resolution counterparts, and so the weights used at low-resolution are inappropriate for high-resolution construction.

Despite access to limited information in [10] and incorrect assumptions of isometry in [11], the concepts are very relevant

[1] Our implementations of statistical learning algorithms used inordinate amounts of training data resulting high-quality image results. Therefore, the information algorithm available to these algorithms is considerable.

and are worthy of elaboration. Among the differences between the proposed algorithm and [10], [11], besides using k neighbors over a single neighbor, the proposed algorithm takes advantage of the special properties of linear filtering for image interpolation. As linear filtering has long been an image interpolating tool in image and video processing, its use is appropriate for our application.

B. Review of k-Nearest Neighbor for Regression

The k-nearest neighbor [35] rule is among the simplest statistical learning tools in density estimation, classification, and regression. Trivial to train and easy to code, the nonparametric algorithm is surprisingly competitive and fairly robust to errors given good cross-validation procedures.

Let Ω be a training set of N input-output pairs. Then

$$\Omega = \{(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2), \ldots, (\mathbf{x}_N, \mathbf{y}_N)\}. \tag{1}$$

For the problem specifically relating to image interpolation, $\mathbf{x}_i$ is comprised of the $i\text{th}$ low-resolution image patch in the training set. Likewise, $\mathbf{y}_i$ defines the $i\text{th}$ high-resolution image patch. During runtime, where we denote runtime values with subscript t, the adaptation of k-NN determines the high-resolution image patch $\mathbf{y}_t$ from a single low-resolution image patch $\mathbf{x}_t$. For mathematical reasons, it is easier to represent image patches $\mathbf{x}$ and $\mathbf{y}$ as vectors instead of square patches. Therefore, in subsequent derivations $\mathbf{x}$ and $\mathbf{y}$ are both vectors that have been rearranged from image blocks into a single column.

The typical k-NN estimate for regression [36], [37] at the test point $\mathbf{x}_t$ is given as

$$\hat{\mathbf{y}} = g(\mathbf{x}_t) = \frac{1}{k}\sum_{i=1}^{N} W_i(\mathbf{x}, \Omega)\mathbf{y}_i \tag{2}$$

where $W_i \in \{0, 1\}$ depending on whether or not $\mathbf{x}_i$ is among the k nearest neighbors of $\mathbf{x}_t$. (To be seen in later sections, instead of g, a linear transformation G performs the task of $\mathbf{y}_t = G\mathbf{x}_t$.)

Naturally, the definition of (2) could be extended by not necessarily limiting W_i to 0 or 1, but rather the constraint $\sum_{i=1}^{N} W_i = k$. In fact, there are several common weighting schemes, ranging from posterior probability like expressions [38] to iteratively determined convex solutions [39], all functions of distances or weights that can used to minimize some criterion as in [11].

An extensive study on error rates for regression-based k-NN estimates was analyzed in [40], where it was conceded that for (x, y) jointly normal,[2] under the squared-error loss case, the unconditional, large sample risk R as $N \longrightarrow \infty$ of the k-NN estimate satisfies

$$R_N^{(k)} = \left(1 + \frac{1}{k} + \frac{\sigma_1^2}{\sigma_2^2} E\left[x_t - \frac{1}{k}\sum_{i=1}^{k} x_{i,N}\right]^2\right) R^*. \tag{3}$$

Here, R^* is the Bayes risk (minimum expected loss), $x_i \in \Omega$, and parameters σ_1 and σ_2 are variance parameters in probability distribution functions (PDF's) $f(y)$ and $f(y|x)$. The tradeoff in (3) would like to keep $1/k$ term to limit erroneous reconstruction while simultaneously favoring a larger k to keep the final term in (3) small. This type of tradeoff is common in k-NN problems, and provides much need for cross validation as will be seen even for adaptable k values.

Of course, it is not always the case that N is large and for these problems, the risk R_{NN} of nearest neighbor (where $k = 1$) is usually smaller than the risk of k-NN [40], but overall, when N is large, k-NN is invariably the rule of choice. Actually, when N is large and the dimensionality of $\mathbf{x}_i$, d, is small, k-NN is almost always preferable or at least competitive among other estimation techniques such as SVR, where SVR performs well when N is small and d is large. (SVR is more of a generalizing technique.) This is fairly intuitive because we would like to blanket the entire domain with samples, possible only with N large, and easier if d were sufficiently small.

It is this last scenario that tends to be the case with the potential of today's computing power. As memory for computing tasks increases and computing time for higher complexity routines decreases, motivation for various k-NN problems (obviously, not limited to interpolation and superresolution) is justified due to the capability of supporting large N values. The following sections offer a basic but original k-NN algorithm while referencing existing k-NN approaches.

III. Adapted k for Unknown Test Points

The traditional view of nearest neighbor algorithms [38] for density estimation is mathematically expressed as

$$\hat{f}(\mathbf{x}) = \frac{1}{N\sigma}\sum_{i=1}^{N} K\left(\frac{\mathbf{x} - \mathbf{x}_i}{\sigma}\right) \tag{4}$$

where K represents a kernel that integrates to one, e.g., Gaussian or radial basis functions (RBF) kernels in our case. Conceptually, (4) estimates $f(\mathbf{x})$ by placing kernels around every training point in order to describe a complete picture of the probability density function (PDF).

The paradigm for our approach is just the opposite, though the final results should remain exactly the same. Instead of thinking that kernels extend around each training point, we can visualize a kernel at the test point since conceptualizing a PDF at runtime is unnecessary. Therefore, distances from the test point to the surrounding training points are evaluated by kernel depending on how far *they* are to the input rather than how far the input is to each of them.

A. Building the Optimal Interpolation Mechanism

To reiterate, kernel function K used in the proposed algorithm is the RBF, stated in (5)

$$K_{\mathbb{F}}(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{2\pi\|\Sigma\|}\exp\{-d_{\mathbb{F}}(\mathbf{x}_i, \mathbf{x}_j)\} \leq 1 \tag{5}$$

where $d_{\mathbb{F}}(\mathbf{x}_i, \mathbf{x}_j)$ is the Mahalanobis distance or weighted Euclidean distance specified by $1/2\,(\mathbf{x}_i - \mathbf{x}_j)^T\,\Sigma^{-1}\,(\mathbf{x}_i - \mathbf{x}_j)$. Unfortunately, in the absence of prior knowledge, most k-NN algorithms determine proximity through un-weighted Euclidean distances. Fortunately, we can calculate the Σ of the entire training set and use a scaled version of it, which actually

[2] [40] only treats the univariate case.

does not improve very much but is better than the un-weighted case.

As alluded to in Section II-B, there are many weighting schemes for W in (2). One family of solutions, which offers more flexibility in application, known as *locally weighted regression* (LWR) [41], [42] can replace W by a particular model class $g(\mathbf{x}, \boldsymbol{\beta})$, in which $\mathbf{y}_t$ is determined locally by functions based on how similar point $\mathbf{x}_i$ is to $\mathbf{x}_t$ [43]. Then, the task of k-NN for regression becomes estimating select parameters for reconstruction in (6)

$$\boldsymbol{\beta}^* = \arg\min_{\beta} \sum_{\mathbf{x}_i \in \mathcal{N}(\mathbf{x}_t)} d_{\mathbb{R}}(g(\mathbf{x}_i, \boldsymbol{\beta}), \mathbf{y}_i) K_{\mathbb{F}}(\mathbf{x}_i, \mathbf{x}_t) \quad (6)$$

where $d_{\mathbb{R}}$ is a distance metric in the range and feature space, respectively, and $\mathcal{N}(\mathbf{x}_t)$ is the neighborhood of $\mathbf{x}_t$.

On the principle that isometry is not a realistic scenario for image superresolution [34], (6) becomes a viable alternative. With LWR, the only required assumption is that linear filtering yields an excellent approximation for local image construction as opposed to assuming some kind of duality between low-resolution and high-resolution manifolds in [11]. Hence, the $g(\mathbf{x}, \boldsymbol{\beta})$ in (6) becomes the linear filter in question, which can be reduced to an MMSE filter formulation, and we can find $\mathbf{y}_t$ by

$$\mathbf{y}_t = E[\mathbf{y}_t | \mathbf{x}_t] \approx g(\mathbf{x}_t, \boldsymbol{\beta}) = G\mathbf{x}_t \quad (7)$$

where G is a $u \times d$ matrix, u being the upsizing factor, and is constructed by probability parameters β and neighboring low-resolution and high-resolution pairs.

The focus from (7) now becomes finding G, which is accomplished by slightly modifying traditional MMSE equations. Eventually, preprocessing steps such as mean-shifting or variance normalization are implemented to determine the feature space $\mathbb{F}$ for both k-NN neighbor identification and regression, but for now, the filters are created in a manner similar to [9]. To manage the data, let us assemble all neighboring low-resolution vectors $\mathbf{x}_i$ and high-resolution vectors $\mathbf{y}_i$ of the test vector $\mathbf{x}_t$ into X and Y matrices, respectively. Hence, we define X and Y matrices as a collection of $\mathbf{x}_i$ and $\mathbf{y}_i$ vertical vectors lined up horizontally.

By defining (5), we can construct a matrix P for a given neighborhood of $\mathbf{x}_t$ such that if $\mathbf{p}$ is a vector of similarity measures whose ith entry is the value $K_{\mathbb{F}}(\mathbf{x}_i, \mathbf{x}_t)$, then

$$P = \mathbf{1}^T \mathbf{p} \quad (8)$$

where k is the number of neighbors to use and $\mathbf{1}$ is a k-dimensional vector of all ones. Hence, P has dimension $d \times k$.

The purpose of P is to establish a proper weighting of point $\mathbf{x}_i \in \mathcal{N}(\mathbf{x}_t)$. Since one of the arguments to the $K_{\mathbb{F}}(\mathbf{x}_i, \mathbf{x}_j)$ in (5) is always $\mathbf{x}_t$, weighting schemes [39] usually observe the similarity between $\mathbf{x}_i$ given $\mathbf{x}_t$ as a Gaussian PDF with mean centered at $\mathbf{x}_t$ and the elements in P as how probable that neighborhood vector is relevant.

Recall that the sample autocorrelation matrix is traditionally defined as $R_{XX} = XX^T$ and cross-correlation matrix as $R_{XY} = XY^T$. The least squares filter formulation takes on the same form in our for the optimal regression, G in (9). The result is is roughly equivalent to the derivations from [9]

$$G = \left((P \cdot X)X^T\right)^{-1} \left((P \cdot X)Y^T\right). \quad (9)$$

B. Choosing the Correct k

There are certain properties in the image interpolation problem that are especially attractive; top among them is the robustness to poorly designed filters. The explanation stems from linear filters' tendency to average out image data. Moreover, weighted MMSE solutions can be interpreted as an averaging of an overdetermined system. As such, when more and more points are involved, or k is grown to be large, the sample size of X grows as well, and G will be able to accommodate a more general base of $\mathbf{x}_t$ vectors. On the other hand, maintaining large k defeats the purpose of k-NN because using smaller k ensures specialization of the resulting filter.

The idea of choosing an adaptable k compensates for nonuniformly distributed training data when the high-dimensional domain of low-resolution image patches are inconsistently scattered across feature space. For the uniformly distributed case, k can be fixed because the variance in the distances from the surrounding training points to any given input test point is limited among all possible test points. In such cases, using a fixed k will always yield roughly equivalent relevancy in training information among any two test points. As it happens, in natural image statistics, the entropy of image patches is difficult to describe and cannot be modeled in any traditional linear sense. The investigation of image patch statistics has been extensively examined in several papers such as [44] and [45],[3] where patches are discovered to be sparsely distributed with the majority of points clustered in high concentration on nonlinear manifolds.

The view of clustered high-dimensional manifolds are ingrained in most image model descriptions in some form, and additional work on the subject is abundant [44]–[47]. Literature on high-resolution image manifolds is fairly involved, and we are not concerned with their specific properties, merely exploiting the fact that they exist. Thus, the proposed algorithm should be tuned and adaptable to the distribution and where arbitrary test inputs might land. The manifestation of such an endeavor is intimately connected to the number of nearest neighbors, or k, given the input test point in a naturally distributed training set.

The goal is to find the right k for a desired tradeoff. As one may guess, $k < k^*$, where k^* is the ideal number of neighbors, overfits the training set specializing G too much, and the manifestation is a grainy and discontinuous image. Furthermore, if k were exceedingly small, $k \ll k^*$, G could become singular. This is intuitive because training points near $\mathbf{x}_t$ could be very close together causing (9) to be underdetermined. Analytically speaking, vectors in X that are too similar can mean that R_{XX} is rank deficient and thus noninvertible. This is a dilemma because while k-NN should find the most relevant data, it is designed such that the collected vectors based on $\mathbf{x}_t$ are similar to each other. Hence, though it is counterintuitive, it is important to choose a large enough neighborhood in $\mathbb{F}$ so that diversity in the $\mathcal{N}(\mathbf{x})$ exists.

[3] [45] assumes two manifolds termed "explicit" and "implicit", but the two types effectively describe the same behavior as [44]

This type of built-in error is a consequence of uneven training data collection in unsupervised learning. Similarly, uneven training data also has implications that k^* may potentially be significantly different for any two given test points. To find the optimal k^*, we introduce η such that k^*, a function of $\mathbf{x}_t$ and the training set Ω, is determined by

$$k^*(\mathbf{x},\Omega) = \text{argmin}_k \quad \sum_{i=1}^{N} W_i(\mathbf{x},\Omega,k)K(\mathbf{x}_i,\mathbf{x}) \geq \eta \quad \text{where} \quad W_i(\mathbf{x},\Omega,k) \in \{0,1\}. \tag{10}$$

The expression in (10) obtains k^* by finding the minimum number of neighbors whose sum of similarity measures exceeds a threshold η, which is obtained through cross validation. Moreover, η is a minimum bound of k since $K(\mathbf{x}_i,\mathbf{x}_t) \leq 1$ for all $\mathbf{x}_i$.

Analyzing (10) for a given $\mathbf{x}_t$, if there are only a few $\mathbf{x}_i$ with high probability of being related to it, that is $\sum_i K(\mathbf{x}_i,\mathbf{x}_t)$ is small, then the proposed algorithm will need to consider more points in hopes of generalizing well. Alternatively, if there are many $\mathbf{x}_i$ that are related to $\mathbf{x}_t$, i.e., $\sum_i K(\mathbf{x}_i,\mathbf{x}_t)$ is large, it is unnecessary to use other points where the similarity is low because the specialized filter generated by the points within $\sum_i K(\mathbf{x}_i,\mathbf{x}_t) \leq \eta$ is very likely to be accurate. Conceptually, we can visualize a ring that extends further and further depending on whether or not there are enough points inside the ring.

C. Heuristics for Insufficient Training

k-NN algorithms assume there are enough points to blanket the entire domain, providing a good density estimate of the input space. Problems then arise from insufficient training because the further the ring of values under consideration extends, the smaller the similarity values, and the less suited any additional training point is to complete the task of reaching η. In extreme cases, η may not even be reached before the entire training set is exhausted of points. Thus, we require the incorporation of a simple heuristic of limiting the maximum value of k that is allowed to be used.

We can set a limit on k^* by letting ζ be a maximum limit on k. After doing so, a generic technique, i.e., bicubic interpolation, may be used for those $\mathbf{x}_t$ that Ω does not represent well. Additionally, the maximum number of neighbors will reduce both complexity and errors. The complexity reduction should be obvious, but to see that errors have been minimized by stopping the algorithm prematurely with ζ, the algorithm effectively acknowledges that, at least for the image patch at hand, the original intention of the proposed algorithm cannot be carried out due to a less than competent training set. Therefore, for any $\mathbf{x}_t$ that k-NN is ill-equipped to manage (i.e., $k^* > \zeta$), the errors are bounded by whatever interpolation algorithm replaces k-NN.

The question now becomes finding what kind of interpolation algorithm should replace k-NN. Is there a particular type of image patch that the k-NN algorithm consistently disfavors? Moreover, based on this bias, are there certain properties of these patches that allow us to tailor a solution using this knowledge? The answer is yes on both accounts. After running several tests, we came across a peculiar reoccurring theme in generic training and testing images: texture patches never reached ζ and appeared at high quality, but edge patches often did and needed attention.

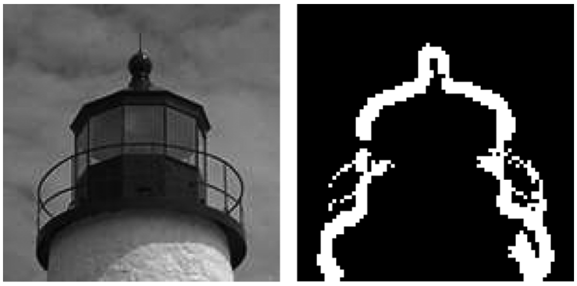

(a) Original Interpolation (b) Patches w/Insufficent Training

Fig. 2. With only 200 thousand data points, we cannot actively reconstruct many edges on the lighthouse because training patches don't occur frequently enough and there aren't close enough matches. Texture, however, *can* be, and much of the texture interpolation occurs because $k^* < \zeta$.

Using 2×10^5 training points and observing similarity measures in (8) (which are based on Euclidean distances), the texture matches usually retain similarity values of $K(\mathbf{x}_t,\mathbf{x}_i) \approx 0.93$ (out of 1.00), whereas edge matches usually satisfy $K(\mathbf{x}_t,\mathbf{x}_i) \leq 0.40$. Furthermore, in viewing a single image,[4] only a small percentage of image patches are actually edges, so accumulating relevant image patches in (10) to surpass η is even more improbable. The situation is best described in Fig. 2.

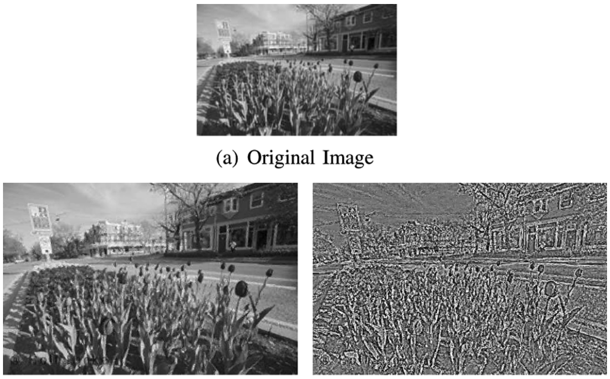
Though texture results in high peak signal to noise ratios (PSNR), to be presented in Section V, the human visual system (HVS) focuses on edges [48]. Fortunately, research into edge-oriented image filtering has been well-studied [49], [50]. In our framework, we can agglomerate a bank of edge-oriented filters that do "well-enough" when the "best" filter through k-NN is unavailable, effectively reducing the implementation to a specialized version of [9] with an added MRF improvement (see the next section, Section IV) through [51].

With enough data points, however, replacing k-NN conditioned on $k^* > \zeta$ should occur relatively few times. That is, edges may and often are well-represented in the training set, which indicates the algorithm is operating closer to capacity. Studies such as these are reserved for current and ongoing work.

IV. Using the Single-Pass Approximation to the Markov Random Field

It is widely acknowledged that local interpolation could benefit from global image information to predict high-resolution pixel values. Construction of high-resolution image detail from isolated, local low-resolution image patches (i.e., without information from adjacent patches and the image as a whole) using a single neighbor is shown in Fig. 3, where overall results are described as looking "like oatmeal" from [10]. Without information from adjacent patches, high-frequency components of an image as a whole become patchy and discontinuous. The most common remedy is to globalize the effort by using surrounding window information, where many image processing algorithms call upon Markov networks. What follows in the rest

[4] Unsupervised data collection means that the exact percentage values of patches are not known.



(a) Original Image

(b) 2× Interpolation

(c) High-Frequencies

Fig. 3. Image example follows implementation of Freeman *et al.* [10]. Superresolution results using a single neighbor with only local and individual patch information with no spatial neighbor patches. On the bottom, the high-frequencies of local reconstruction efforts look "like oatmeal."

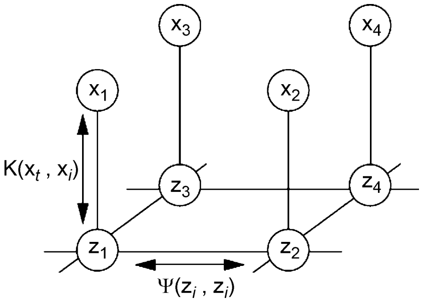
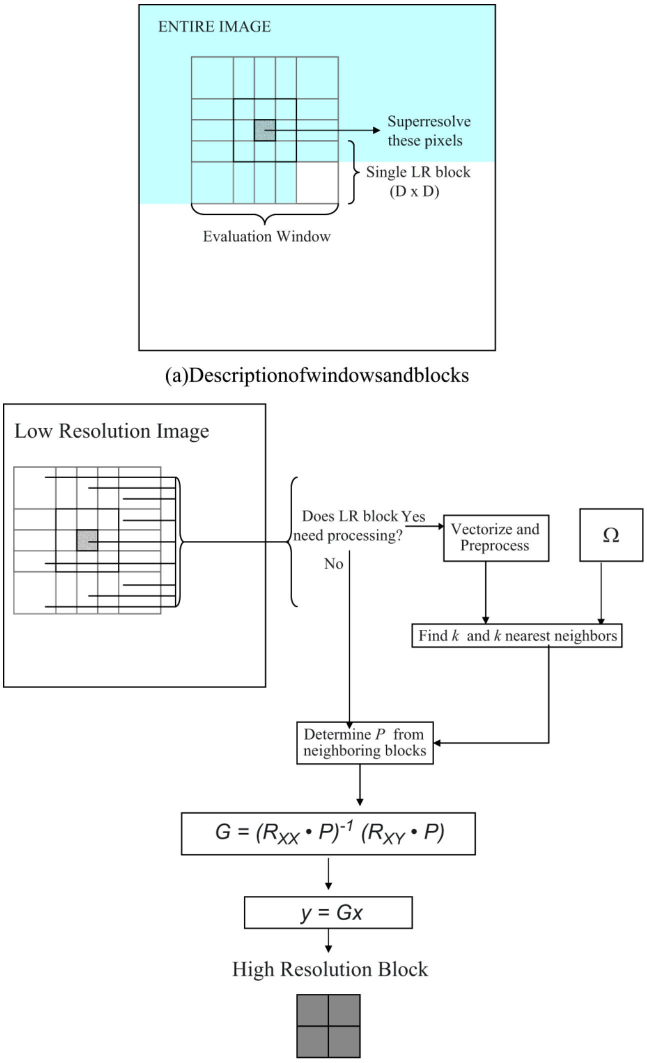


Fig. 4. This diagram is very similar to the one in [10] (the difference being terminology.) We use $K(\mathbf{x}_t, \mathbf{x}_i)$, where $\mathbf{x}_t$ is the observation and $\mathbf{x}_i \in \Omega$ in Section III to determine possible states, $\{\mathbf{z}\}$. We can use a function similar to K for Ψ to determine the compatibility between the states.


ENTIRE IMAGE
Superresolve these pixels
Single LR block (D x D)
Evaluation Window
Low Resolution Image
Does LR block need processing?
Yes
No
Vectorize and Preprocess
Ω
Find k and k nearest neighbors
Determine P from neighboring blocks
$G = (R_{XX} \bullet P)^{-1} (R_{XY} \bullet P)$
$y = Gx$
High Resolution Block

(a)Descriptionofwindowsandblocks

(b)Block diagram of algorithm

Fig. 5. Evaluation window (a) for the algorithm in which all k neighbors are known or need to be found. The center block depicts the block that is to be superresolved. The surrounding blocks supply ψ values to weight the consideration of individual neighbors. All LR blocks have or need to calculate k, and the entire algorithm (b) uses these blocks in the evaluation window to determine G and the high-resolution patch. In raster scans, for the majority of the time, the only LR block for which neighbors have not been calculated is the lower right block. Naturally, more of the blocks in the evaluation window require a search for k neighbors at the borders of an image or if implementation restrictions dependent on the available memory.

of the section is a description of a proposed implementation of a single-pass approximation to the Markov Random Field to consider neighboring areas surrounding a window and alleviate the effect of Fig. 3.

In terms of nearest neighbor algorithms for image interpolation, [10] is most like our algorithm. The differences between the proposed algorithm and [10] are subtle but significant. [10] chooses a single neighbor among a number of candidate neighbors and applies the high-resolution differences. Our algorithm chooses an "appropriate" quantity of candidate neighbors and directly determines the high-resolution content. The philosophy that leads to this arises from differences in the choice of feature space, where [10] first uses an analytic interpolation scheme, such as bicubic interpolation, and stores differences between the initial interpolation and the true high resolution patch. In contrast, the preprocessing involved in our algorithm only consists of DC subtraction and scaling. In essence, we trust the capacity of the *algorithm* to perform the approximation rather than first approximating outside the algorithm.

Despite the difference, [10] makes a good argument that globalization in terms of relating neighboring patches is necessary. Consequently, we have followed their lead by considering the usage of Markov network in modeling spatial relationships between patches. Such modeling techniques require the use of some brand of annealing process, which is usually computationally intractable for most interpolation purposes. Therefore, with the aid of [51] and [10], we have implemented a simpler-than-MRF, single-pass technique to enhance coherency from patch to patch.

Single pass algorithms include extra arguments into the decision making process that increase propensity towards one neighbor over another. Obviously, because our algorithm observes multiple neighbors per input patch, the structure of the one pass algorithm must be modified somewhat.

Given the filter construction process in (9), we can take advantage of an expression that is already designed to penalize or reward training points through a matrix P. To review conceptually, elements within P denote the importance of a particular training point. After determining the k^* values for all

TABLE I
MISCELLANEOUS PSNR COMPARISONS

METHOD	**PSNR Values**		
	Pirates	Lighthouse	Bus,4
Bicubic	28.39 dB	28.87 dB	24.55 dB
NEDI [1]	26.82 dB	27.44 dB	22.57 dB
SEL [2]	26.67 dB	27.38 dB	22.63 dB
128 Class RS [9]	28.91 dB	28.98 dB	26.48 dB
LLE [11]	21.97 dB	22.70 dB	18.04 dB
Example-Based [10]	27.28 dB	29.77 dB	25.40 dB
Fixed k-NN	29.23 dB	29.12 dB	25.42 dB
Proposed Algorithm	30.27 dB	29.81 dB	26.38 dB

image patches (or realistically, just the ones surrounding the test patch being evaluated), the logical course of action would be to reward those states that contain high values for $K(\mathbf{x}_i, \mathbf{x}_j)$ and $\Psi\left(\mathbf{z}^{(x_1,y_1)}, \mathbf{z}^{(x_2,y_2)}\right)$. Adding a scaling factor α, a very simple conditioning scheme could be

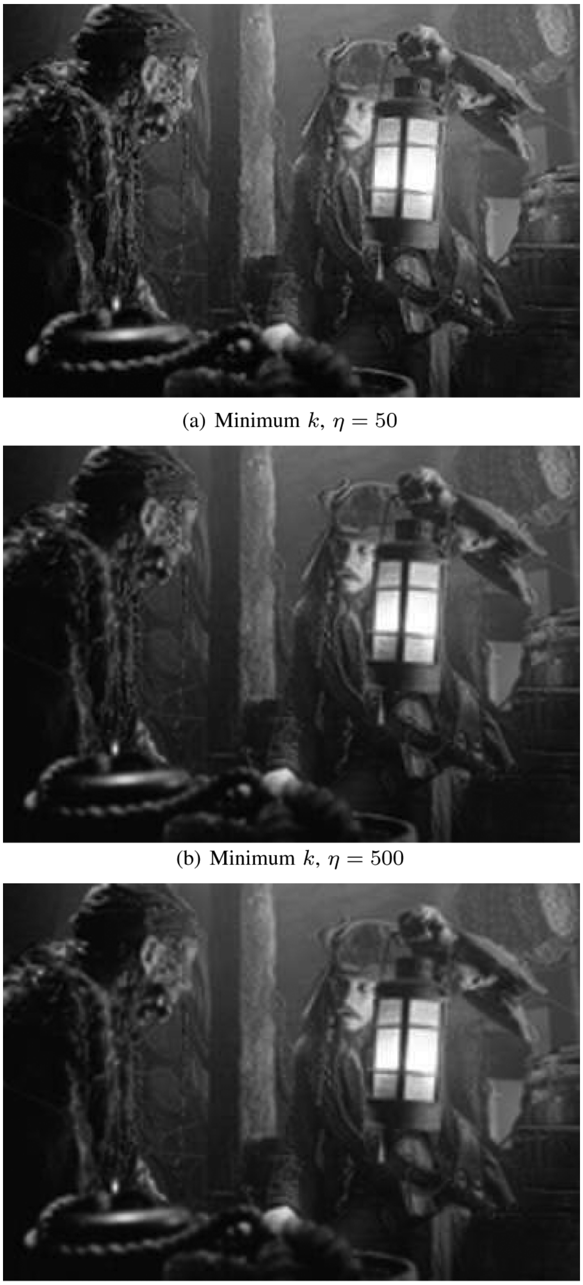
$$P_{(i,\cdot)} = K(\mathbf{x}_t, \mathbf{x}_i) + \alpha \sum_{n\in\mathcal{N}} \sum_{j} \Psi\left(\mathbf{z}_j^{(n)}, \mathbf{z}_i\right). \tag{11}$$

Here, $\mathbf{z}_j^{(n)}$ refers to the $j\mathrm{th}$ candidate state (see Fig. 4) of the $n\mathrm{th}$ low-resolution (LR) block in $\mathcal{N}$, the neighborhood of the input block. Referring to the entire algorithm in Fig. 5, blocks in $\mathcal{N}$ are adjacent to the input test block.

V. RESULTS

We define our feature space as vectorized 5 × 5, centered and normalized pixel patches. Hence, $d = 5$ in $\mathbf{x} \in \mathbb{R}^{d^2}$, and for interpolation by $u = 2$, our algorithm follows $\mathbb{R}^{d^2} \longrightarrow \mathbb{R}^{u^2}$. We have collected a diverse set of high-resolution training images at an archive available at http://calphotos.berkeley.edu, and generated the low-resolution image through MATLAB's `imresize` command. Therefore, the low-resolution training patches are generated from decimated and anti-aliasing filtered high-resolution images. With respect to Section IV, the proposed algorithm uses a relatively small influence of α in a variation of (11), and we define a kernel function, similar to K, for our choice of Ψ. The training set is chosen to have all types of textures and image content so that the highest level of generalization can occur. In addition, we have included several speedups for runtime consideration in our experimental setup (see the webpage below for specific details).

Given the occurrences of insufficient data in Fig. 2, the assumption that a very large training set is at the proposed algorithm's disposal remains (common for any k-NN algorithm). The issue is highlighted as the proposed algorithm fails to yield meaningful results when the high-resolution image to be superresolved is the only item in training. Hence, our experiments mandate a minimum of 12 to 14 large images, where resources consist of millions of image patches. (The total number of training for the set of test runs in Table I is $N =$ 4,309,914 points.) The phenomenon returns to previous explanations of determining filter coefficients and the proposed algorithm's choice of k. As it turns out, creating a nonsingular matrix is surprisingly difficult, because very often insufficient data plagues the construction effort.



(a) Minimum k, $\eta = 50$

(b) Minimum k, $\eta = 500$

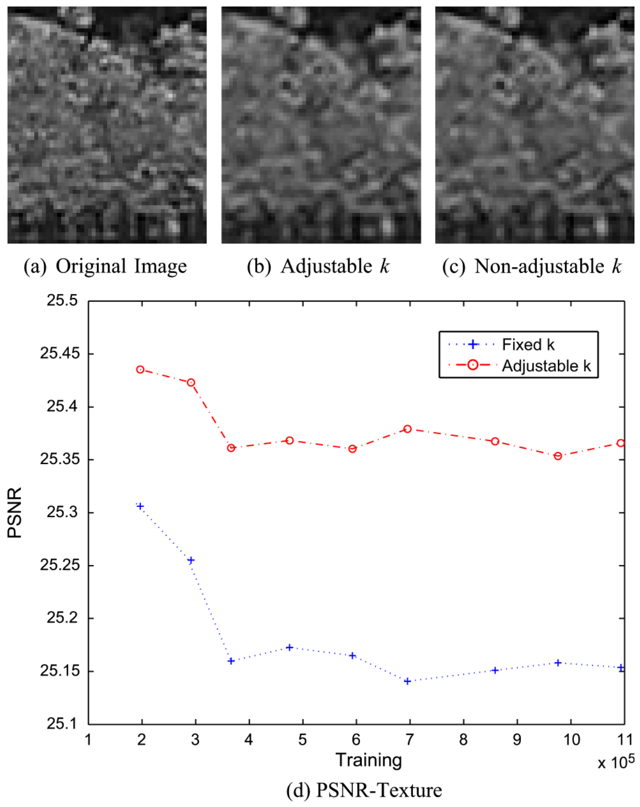
(c) Minimum k, $\eta = 5,000$

Fig. 6. Pirate images for various cross-validation values of $\eta = k_{\min}$.

A. Cross Validation and Local Content-Based Results

In terms of cross validation, there are two degrees of freedom in our experiments, σ, the bandwidth parameter of the Gaussian used to extend around the test point, or η, the minimum number of training points necessary. Between the two, it is easier to alter η because bandwidth σ is a squared exponent term and is difficult to control.

Fig. 6 depicts the role of η, where the images are placed in logarithmically increasing order of η. While Fig. 6(a) appears clearer and sharper, there appears to be quite a number of errors that afflict the construction effort. Meanwhile, Fig. 6(c) appears blurrier and more washed out, but there are fewer errors.


(a) Original Image (b) Adjustable k (c) Non-adjustable k

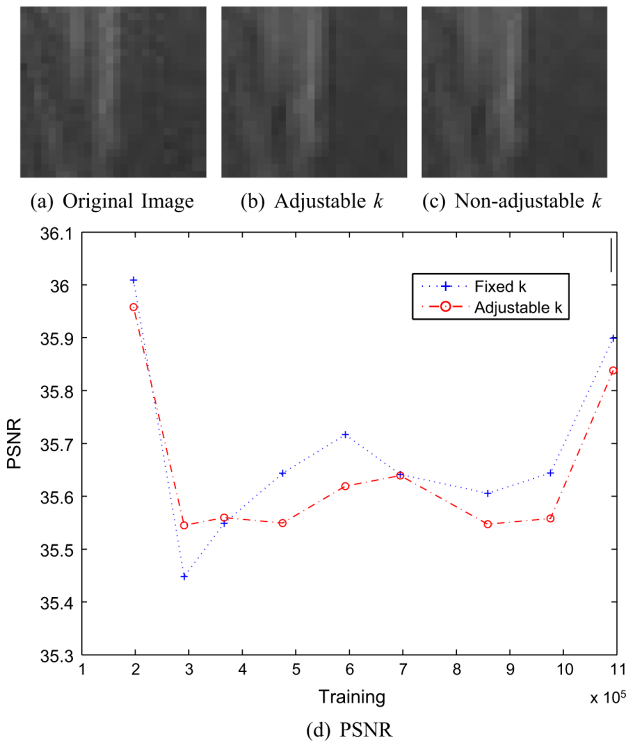
(d) PSNR-Texture

Fig. 7. Comparisons to fixed k in k-NN of texture. The proposed algorithm is consistent in superior quantitative performance for any texture that is tested.

Images are diverse entities, but for brevity and analysis purposes, they are divided into two categories in Figs. 7 and 8: texture and edges. Both figures compare the performance of a variable k versus a fixed k in nearest neighbor weighted filtering. Adjustable k PSNR results in Fig. 8(d) are somewhat lower than fixed k results. Given the behavior of k-NN in edge image patches, seen in Fig. 2, the performance is justified by some intuition. For most edge content, the amount of training that is "close" to $\mathbf{x}_t$ is insufficient, which is manifested in some ambient noise in Fig. 8(c). Recalling Section III-C, the phenomena makes sense as edges and sharp, uniform gradients in training sets are generally not represented well in terms of Euclidean L_2 distances. With the same training set, the nonadjustable k cannot pool from additional data though the neighbors it uses are not very "relevant". The adjustable k-NN algorithm will, on the other hand, attempt to obtain more data for a general result, and the image becomes softer with fewer errors in Fig. 8(b). Nevertheless, in obtaining a very general filter, the adjustable k-NN sacrifices high PSNR values because it uses a very general and broad filter that averages more than it specializes. Hence, the fixed k-NN will have higher PSNR values.

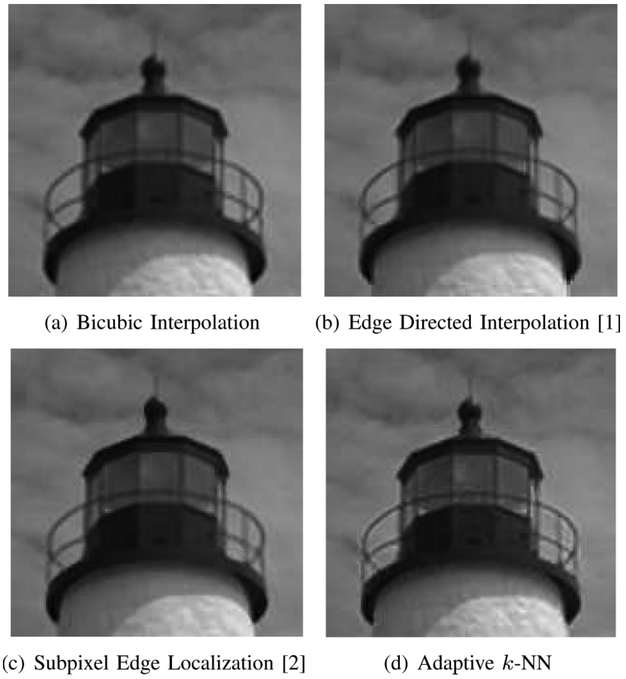
In the opposite spectrum, texture, according to Fig. 2, has a strong showing in most training sets. Therefore, the quantitative values in Fig. 7(d) reflect specific filters that are specially designed for the situation determining a quantitatively superior result.

From descriptions in Section III, values of k are directly correlated with the amount of training, N. Increasing N usually results in decreasing k, and this is verified through experimentation because k tended to stay around select values, k_{avg}, for particular N. In Figs. 9 and 10, k had a standard deviation of 59.79 on average staying around 151.76. Often $k_{\max} = 10^6$ was reached, i.e., $k^* \geq \zeta$, but it was usually either hit or miss, where


(a) Original Image (b) Adjustable k (c) Non-adjustable k

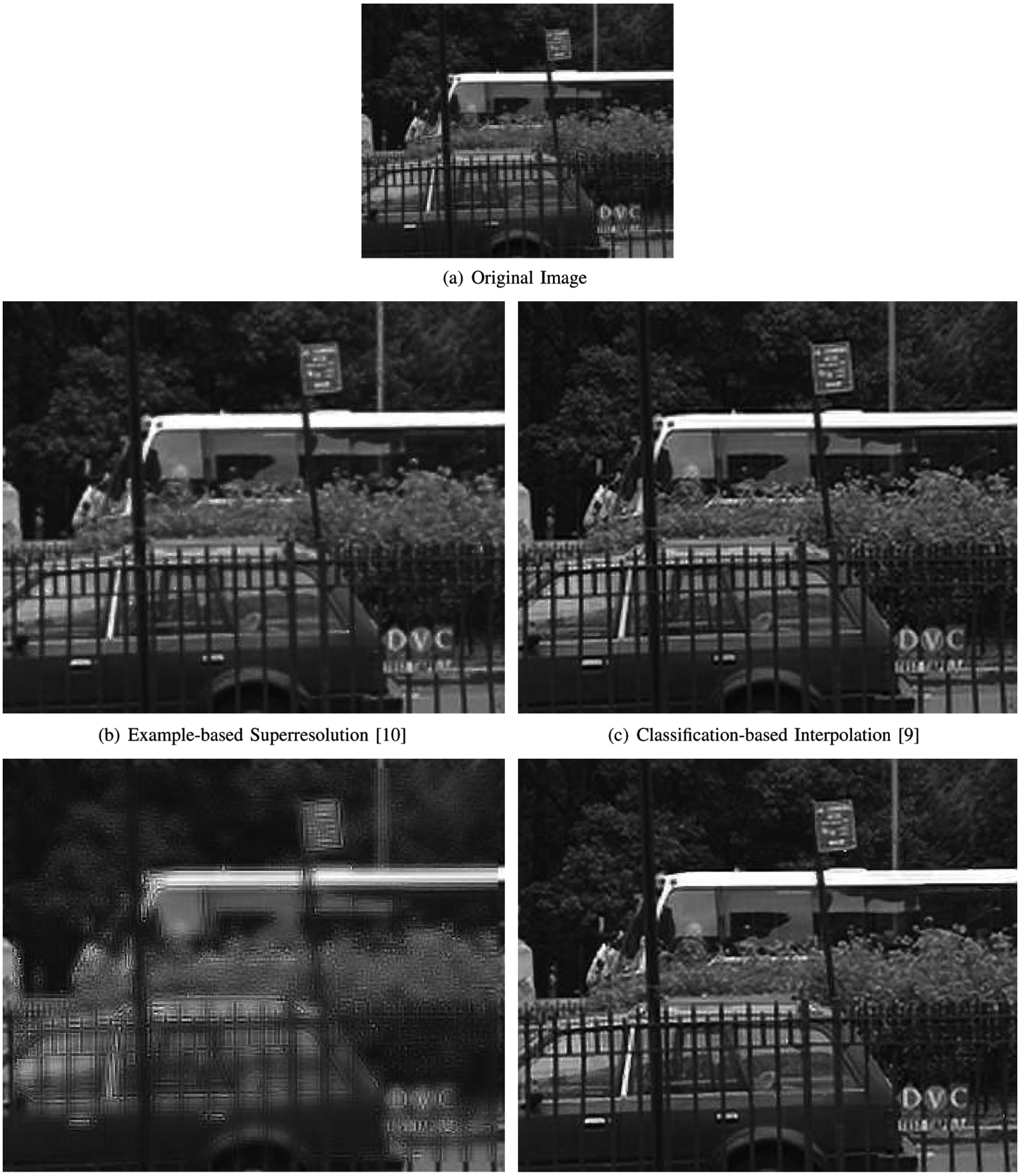
(d) PSNR

Fig. 8. Comparisons to fixed k in k-NN of edges. The proposed algorithm sacrifices some numerical performance to ensure visually pleasing results.


(a) Bicubic Interpolation (b) Edge Directed Interpolation [1]

(c) Subpixel Edge Localization [2] (d) Adaptive k-NN

Fig. 9. Comparisons to state of the art nonstatistical interpolation techniques.



(a) Original Image

(b) Example-based Superresolution [10]

(c) Classification-based Interpolation [9]

(d) Neighbor-Embedding [11]

(e) Adaptive k-NN

Fig. 10. Comparisons of the bus image for various statistical learning interpolation techniques. [11] (d) seems to work for select images only.

either k would be close to k_{avg} or ζ was reached, in which case, image values were created using bicubic interpolation.

B. *Comparisons to Nonlearning-Based Algorithmic Results*

The proposed algorithm naturally appears superior over state of the art interpolation algorithms in Fig. 9. Numerical results in Table I also highlight the proposed algorithm's advantage over nonlearning-based methods. These comparisons are probably not very fair, because the amount of information available to the proposed algorithm is greater than any of those compared to it.

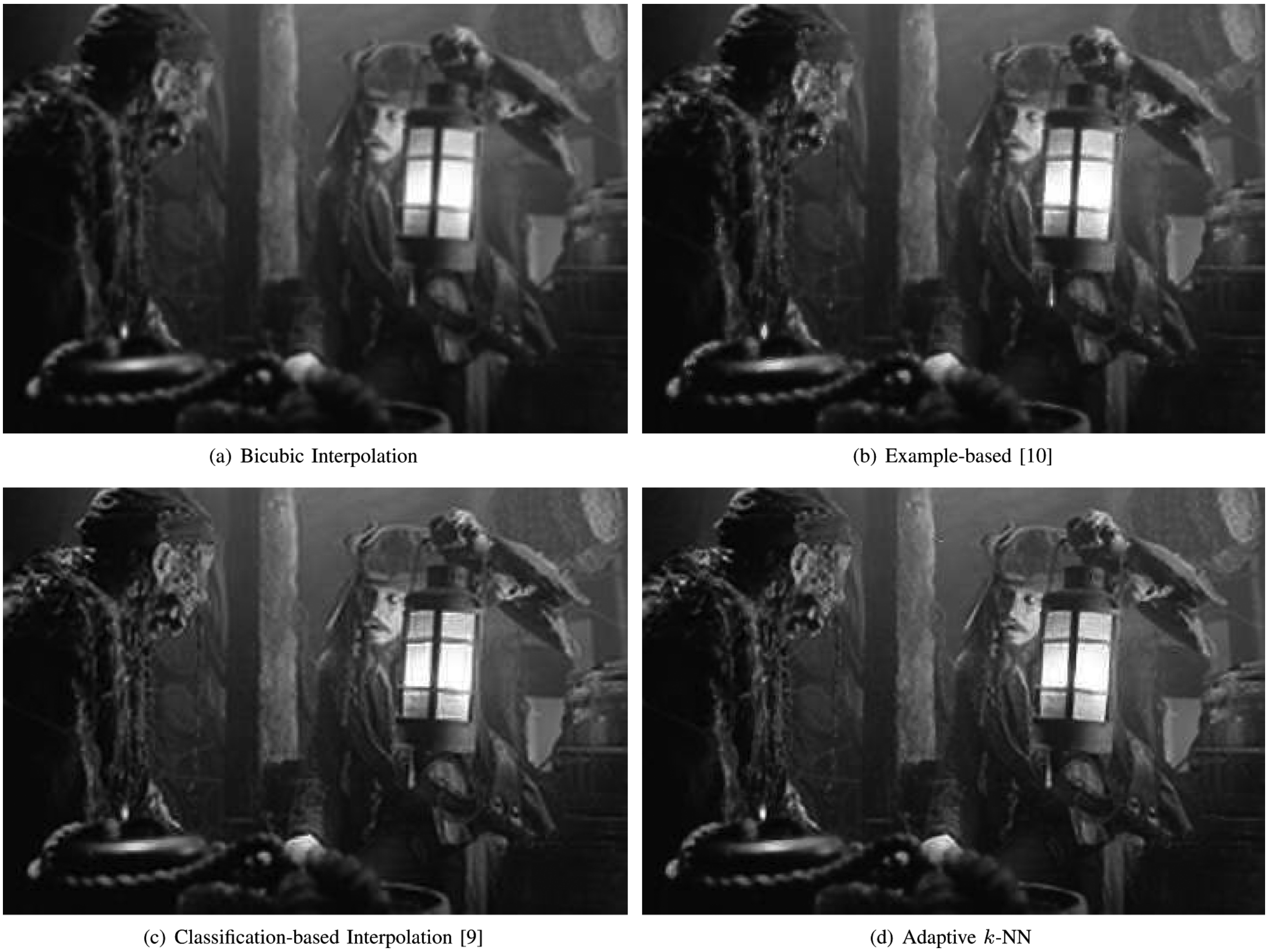
C. *Comparison to Learning-Based Algorithmic Results*

Fig. 10 shows the qualitative results of other nearest neighbor and statistical classification algorithms alongside our own, and Table I gives the quantitative results[5] for various images. Although PSNR is a standard quantitative metric image assessment, it has widely been acknowledged as a poor judge of image

[5] Peak signal-to-noise ratio (PSNR) is the widely accepted and commonly used standard of quantitatively measuring image quality.

$$\mathrm{PSNR} = 10 \cdot \log_{10} \frac{255^2}{MSE}.$$

The value 255 is used because it is the maximum image value. MSE stands for mean squared error, the standard definition.



(a) Bicubic Interpolation

(b) Example-based [10]

(c) Classification-based Interpolation [9]

(d) Adaptive k-NN

Fig. 11. Comparing pirate images. [9] seems to be oversharp in the areas such as the face, the side of the hat. While [10] concentrates on edge-continuity, Adaptive k-NN elucidates the texture of the image. Notice the barrel, the wooden pillar in the background, hat, etc.

quality in terms of the human visual system (HVS). It does, however, give us a good idea of the numerical accuracy or difference between two images, and one can conclude that the proposed algorithm estimates pixels much closer to their original high-resolution values than most other algorithms.

Figs. 10(b) and 11(b) show a decidedly edge-centric result, which can be explained by the neighborhood regularization done through Markov networks. The algorithm does especially well in areas where the original image contains a disparity between textures such as the border between the bus and the background, but gives average performance where edges are less well-defined [see the pole above the bus in Fig. 10(b)]. The same phenomenon can be seen for involved textures (see the the barrel and pillar in Fig. 11). This result is interesting, and could be a byproduct of the particularly large emphasis that Freeman places on texture regularity. More likely, Section III-C reasons that the trend could be due to an incorrect choice of a single neighbor from insufficient edge information. Figs. 10(e) and 11(d) follow this model, but here, the benefits of k rather than a single neighbor becomes apparent, significantly reducing the same artifacts that afflict [10].

Again, our algorithm does not have the luxury of an initial interpolation stage, instead filtering from scratch, and when there is insufficient training, cross validation for η and σ is difficult albeit possible. Owing to this fact, the proposed algorithm manages smaller N sizes at the lower end of $\alpha < 10$ in $\alpha \times 10^5$ worse than [10]. Yet, when N is large, it often outperforms all statistical methods, the conclusion here being that the proposed algorithm represents the inherent relationship well.

Interestingly enough, while [11] is the most similar in theory to our algorithm, the results [Fig. 10(d) and in general] did not reflect this. No experiment adjusting any parameter would yield acceptable results. Experiments in [11] trained on single images, but the feature space of [11] in those cases was an astonishing 100 dimensions, for which we would usually expect large N. Surprisingly, even with large N, the algorithm still remains at a disadvantage; our implementation[6] of [11] was never able to achieve the generalization of any of the other statistical learning algorithms. In fact, additional postprocessing was required to rid the constructed image of speckle noise from erroneous estimation. The errors may be due in part to the fact that neighboring patch information is not considered. Because the neighborhood preservation rate of the output patch is on average less than 10% [34], we can expect little continuity in the image re-

[6] This was later independently verified by the author of the original work, [34], and code from the author of [11] through e-mail correspondences.

sult. Another possible explanation attributes the errors to small k values, usually less than 40, which prevents the contribution of large amounts of information to offset estimation errors, i.e., k often did not scale proportionately with N. The proposed algorithm with $N \approx \alpha \times 10^6$ found k on average in the high hundreds, which oddly enough, applied to [11] gave even poorer visual quality.

Figs. 10(c) and 11(c) from [9] give slightly clearer results than the other statistical methods, but introduce sharpening errors in many instances. This is due in part to the number of classes involved (128) because the performance varies with different class sizes. Intuitively, the two methods should converge when the k in k-NN algorithms is excessively large and the number of classes in classification-based algorithms[7] is large as well. This is eventually true, but the way in which this happens is unexpectedly nonmonotonic. Ultimately, the tradeoff pits the number of data points per class against the number of classes, where N plays an obvious role. Current results show that as N increases (as well as the number of classes), the role of k-NN plays less, and classification based algorithms inexplicably do better. Barring this broad analysis, which composes the subject of current work, the only relevant conjecture made here is that conventional classification-based algorithms could stand to improve due their deficiency of classes.

Additional comparisons and results can be found at http://videoprocessing.ucsd.edu/~karl/k-NN/.

VI. Conclusions and Future Work

A k-NN algorithm with optimal filters and a variable k, determined by relevant training, has been proposed, tested, and compared to the state of the art. The analysis of this algorithm leads to several conclusions.

1) For small training sets, edges cannot be accurately depicted with any nearest neighbor algorithm (if the metric used is Euclidean distance).
2) For large training sets, k-NN performs especially well, both quality and quantity-wise.
3) Fast neighboring-patch approximations of a Markov Network elucidate edges and provide good continuity but sometimes hinder texture synthesis in cases where the training is limited.
4) Linear filtering is a good mask and covers up considerable estimation errors.
5) The direct application of k-NN regression with slight modifications exhibits competitive image quality and offers detailed texture.

The investigation of k-NN opens several avenues of design and analysis, which in the end, could culminate into a grand unifying theory for nonparametric statistical methods in image interpolation. More immediately though, a need for a replacement of Euclidean distance search seems pressing. [52] deals directly with this problem for image patches in one of the chapters. Because k-NN algorithms rely on searches for points with the closest distances, other solutions usually concentrate on complexity reduction and quality of search. The manifestation is some kind of feature space representation [27] or dimensionality reduction/manifold learning. The latter of the two can be done in several ways, and the concept of semi-definite embedding [53] is especially attractive.

Finally, as mentioned earlier, patch-based training images have yielded interesting properties, among them differences between large-class clustering and k-NN as well as local image distribution. While empirically touched upon in Section III-C, theoretical and statistical analysis of the domain representation remains to be explored, the topic of work underway.

[7] [9] grows a tree for training, but then prunes the tree. Comparisons for our purposes deal with overfitting by throwing more data at the problem, rationalizing the omission of the pruning process to the sheer quantity of training.

References

[1] X. Li and M. Orchard, "New edge-directed interpolation," *IEEE Trans. Image Process.*, vol. 10, pp. 1521–1527, 2001.

[2] K. Jensen and D. Anastassiou, "Subpixel edge localization and the interpolation of still images," *IEEE Trans. Image Process.*, vol. 4, pp. 285–295, 1995.

[3] V. R. Algazi, G. E. Ford, and R. Potharlanka, "Directional interpolation of images based on visual properties and rank order filtering," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, 1991, vol. 4, pp. 3005–3008.

[4] Z. Alkachouh and M. G. Bellanger, "Fast DCT-based spatial domain interpolation of blocks in images," *IEEE Trans. Image Process.*, vol. 9, pp. 729–732, 2000.

[5] J. I. Agbinya, "Interpolation using the discrete cosine transform," *Electron. Lett.*, vol. 28, no. 20, 1992.

[6] N. Nguyen, P. Milanfar, and G. Golub, "A computationally efficient superresolution image reconstruction algorithm," *IEEE Trans. Image Process.*, vol. 10, pp. 573–583, 2001.

[7] P. Vandewalle, S. Susstrunk, and M. Vetterli, "Superresolution images reconstructed from aliased images," in *Proc. SPIE/IS&T Visual Communications and Image Processing Conf.*, 2003, vol. 5150, pp. 1398–1405.

[8] S. Borman and R. L. Stevenson, "Simultaneous multi-frame MAP super-resolution video enhancement using spatio-temporal priors," in *Proc. IEEE Int. Conf. Image Processing*, Kobe, Japan, Oct. 1999, vol. 3, pp. 469–473.

[9] C. B. Atkins and C. Bouman, "Classification Based Methods in Optimal Image Interpolation," Ph.D. dissertation, Purdue Univ., West Lafayette, IN, 1998.

[10] W. T. Freeman, T. R. Jones, and E. C. Pasztor, "Example-based super-resolution," *IEEE Comput. Graph. Appl.*, vol. 22, no. 2, pp. 56–65, Mar./Apr. 2002.

[11] H. Chang, D.-Y. Yeung, and Y. Xiong, "Super-resolution through neighbor embedding," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2004, vol. 01, pp. 275–282.

[12] C. A. Dávila and B. R. Hunt, "Training of a neural network for image superresolution based on a nonlinear interpolative vector quantizer," *Appl. Opt.*, vol. 39, pp. 3473–3485, Jul. 2000.

[13] F. M. Candocia and J. C. Principe, "Super-resolution of images based on local correlations," *IEEE Trans. Neural Netw.*, vol. 10, no. 3, pp. 372–372, Mar. 1999.

[14] H. Hou and H. Andrews, "Cubic splines for image interpolation and digital filtering," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 26, 1978.

[15] M. Unser, "Splines: A perfect fit for signal and image processing," *IEEE Signal Process. Mag.*, vol. 16, no. 6, pp. 22–38, Nov. 1999.

[16] A. N. Netravali and B. G. Hasskell, *Digital Pictures: Representation, Compression and Standards*, 2nd ed. New York: Plenum, 1995.

[17] V. Kober, M. A. Unser, and L. P. Yaroslavsky, "Spline and sinc signal interpolations in image geometrical transforms," in *Proc. SPIE*, vol. 2363, pp. 152–161.

[18] S. Yang and K. Hong, "Bilateral interpolation filters for image size conversion," *Proc. Int. Conf. Image Processing*, pp. 986–989, 2005.

[19] P. Tsai and T. Archarya, "Image up-sampling using discrete wavelet transform," presented at the Joint Conf. Information Services, Advances in Intelligent Systems Research, 2006.

[20] G. Uytterhoeven, "Wavelets: Software and Applications," Ph.D. dissertation, Katholieke Univ. Leuven, Leuven, 1999.

[21] C.-M. Kao, X. Pan, M. Anastasio, and P. La Riviere, "An interpolation method using signal recovery and discrete fourier transform," presented at the IEEE Medical Imaging Conf., 1998.

[22] P. J. La Riviere and X. Pan, "Mathematical equivalence of zero-padding interpolation and circular sampling theorem interpolation with implications for direct fourier image reconstruction," presented at the SPIE Medical Imaging Conf., 1998.

[23] A. Temizel and T. Vlachos, "Image resolution upscaling in the wavelet domain using directional cycle spinning," *J. Electron. Imag.*, vol. 14, no. 4, pp. 040501–040501, 2005.

[24] W. K. Carey, D. B. Chuang, and S. S. Hemami, "Regularity-preserving image interpolation," *IEEE Trans. Image Process.*, vol. 8, no. 9, pp. 1293–1297, Sep. 1999.

[25] W. K. Carey, D. B. Chuang, and S. S. Hemami, "Regularity-preserving image interpolation," in *Proc. Int. Conf. Image Processing*, 1997, pp. 901–908.

[26] K. Ni, S. Kumar, T. Q. Nguyen, and N. Vasconcelos, "Single image superresolution based on support vector regression," presented at the Int. Conf. Acoustics, Speech, and Signal Processing, May 2006.

[27] K. Ni, S. Kumar, and T. Q. Nguyen, "Learning the kernel matrix for superresolution," presented at the IEEE Conf. Multimedia Signal Processing, Aug. 2006.

[28] K. Ni and T. Q. Nguyen, "Kernel resolution synthesis for superresolution," presented at the Int. Conf. Acoustics, Speech, and Signal Processing, 2007.

[29] K. Ni and T. Q. Nguyen, "Color image superresolution," presented at the Int. Conf. Image Processing, Sep. 2007.

[30] K. S. Ni and T. Q. Nguyen, "Image superresolution using support vector regression," *IEEE Trans. Image Process.*, to be published.

[31] C. A. Dávila and B. R. Hunt, "Training of a neural network for image superresolution based on a nonlinear interpolative vector quantizer," *Appl. Opt.*, vol. 39, pp. 3473–3485, Jul. 2000.

[32] C. Miravet and F. B. Rodriguez, "A hybrid mlp-pnn architecture for fast image superresolution," *Lecture Notes Comput. Sci.*, vol. 2714, pp. 401–401, 2003.

[33] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *J. Roy. Statist. Soc.*, vol. 39, pp. 1–38, 1977.

[34] K. Su, Q. Tian, Q. Que, N. Sebe, and J. Ma*, Neighborhood Issue in Single-frame Image Superresolution*. Los Alamitos, CA: IEEE Computer Society, 2005.

[35] K. S. Fu*, Digital Pattern Recognition*, 2nd ed. New York: Springer-Verlag, 1980.

[36] L. P. Devroye, L. Gyorfi, A. Krzyzak, and G. Lugosi, "On the strong universal consistency of nearest neighbor regression function estimates," *Ann. Statist.*, vol. 22, no. 3, pp. 1371–1385, 1994.

[37] L. P. Devroye and T. J. Wagner, "Distribution-free consistency results in nonparametric discrimination and regression function estimation," *Ann. Statist.*, vol. 8, no. 2, pp. 231–239, 1980.

[38] R. Duda, P. Hart, and D. Stork*, Pattern Classification*, 2nd ed. New York: Wiley, 2000.

[39] A. F. Atiya, "Estimating the posterior probabilities using the K-nearest neighbor rule," *Neural Comput.*, vol. 17, no. 3, pp. 731–740, 2005.

[40] T. Cover, "Estimation by the nearest neighbor rule," *IEEE Trans. Inf. Theory*, vol. IT-14, Jan. 1968.

[41] W. S. Cleveland and S. J. Delvin, "Locally weighted regression: An approach to regression analysis by local fitting," *J. Amer. Statist. Assoc.*, vol. 83, no. 403, pp. 596–610, 1988.

[42] C. G. Atkeson, A. W. Moore, and S. Schaal, "Locally weighted learning," *Artif. Intell. Rev.*, vol. 11, pp. 11–74, 1997.

[43] G. Shakhnarovich, T. Darrell, and P. Indyk*, Nearest-Neighbor Methods in Learning and Vision: Theory and Practice*. Cambridge, MA: The MIT Press, 2005.

[44] A. B. Lee, K. S. Pedersen, and D. Mumford, "The nonlinear statistics of high-contrast patches in natural images," *Int. J. Comput. Vis.*, vol. 54, no. 1–3, pp. 83–103, 2003.

[45] K. Shi and S.-C. Zhu*, Mapping Natural Image Patches by Explicit and Implicite Manifolds*, pp. 1–7, Jun. 2007.

[46] J. Huang and D. Mumford, "Statistics of natural images and models," presented at the IEEE Computer Society Conf. Computer Vision and Pattern Recognition, 1999.

[47] G. Carlsson, T. Ishkhanov, V. de Silva, and A. Zomorodian, "On the local behavior of spaces of natural images," *Int. J. Comput. Vis.*, vol. 76, no. 1, pp. 1–12, Jan. 2008.

[48] D. C. Burr, M. C. Morrone, and D. Spinelli, "Evidence for edge and bar detectors in human vision," *Vis. Res.*, vol. 4, pp. 419–431, 1989.

[49] J. Taguchi, K. Kido, and K. Sano, "Directional image filter respectively adjusted to edge and flat regions," *Syst. Comput. Jpn.*, vol. 30, no. 8, pp. 72–78, 1999.

[50] R. C. Gonzalez and R. E. Woods*, Digital Image Processing*. Boston, MA: Addison-Wesley Longman, 2001.

[51] M. Li and T. Nguyen, *Discontinuity-adaptive de-interlacing scheme using Markov random field model,* 2006.

[52] G. Shakhnarovich, "Learning Task-Specific Similarity," Ph.D. dissertation, Massachusetts Inst. Technol., Cambridge, MA, 2005.

[53] K. Q. Weinberger and L. K. Saul, "Unsupervised learning of image manifolds by semidefinite programming," *Int. J. Comput. Vis.*, vol. 70, no. 1, pp. 77–90, 2006.

Karl S. Ni (S'02–M'09) was born in Galveston, TX, in 1980. He received the B.S. degree from the University of California, Berkeley, in June 2002, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of California at San Diego, La Jolla, in Fall 2005 and Spring 2008. He is currently a research staff member at the Massachusetts Institute of Technology Lincoln Laboratories, Lexington.

His research interests include the general application of machine and statistical learning techniques to video and image postprocessing problems.

Truong Q. Nguyen (F'05) is currently a Professor at the ECE Department, University of California at San Diego, La Jolla. His research interests are video processing algorithms and their efficient implementation. He is the coauthor (with Prof. G. Strang) of the popular textbook *Wavelets and Filter Banks* (Wellesley-Cambridge, 1997) and the author of several Matlab-based toolboxes on image compression, electrocardiogram compression and filter bank design. He has authored over 200 publications.

Prof. Nguyen received the IEEE TRANSACTIONS ON SIGNAL PROCESSING Paper Award (Image and Multidimensional Processing area) for the paper he co-wrote with Prof. P. P. Vaidyanathan on linear-phase perfect-reconstruction filter banks (1992). He received the National Science Foundation Career Award in 1995 and is currently the Series Editor (Digital Signal Processing) for Academic Press. He served as an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING (1994–1996), the IEEE SIGNAL PROCESSING LETTERS (2001–2003), the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS (1996–1997, 2001–2004), and the IEEE TRANSACTIONS ON IMAGE PROCESSING (2004–2005).