COLOR IMAGE SUPERRESOLUTION BASED ON A STOCHASTIC COMBINATIONAL CLASSIFICATION-REGRESSION ALGORITHM

Karl S. Ni, Truong Q. Nguyen

ECE Dept, UCSD, La Jolla, CA 92093-0407
http://videoprocessing.ucsd.edu/

ABSTRACT

The proposed algorithm in this work provides superresolution for color images by using a learning based technique that utilizes both generative and discriminant approaches. The combination of the two approaches is designed with a stochastic classification-regression framework where a color image patch is first classified by its content, and then, based on the class of the patch, a learned regression provides the optimal solution. For good generalization, the classification portion of the algorithm determines the probability that the image patch is in a given class by modeling all possible image content (learned through a training set) as a Gaussian mixture, with each Gaussian of the mixture portraying a single class. The regression portion of the algorithm has been chosen to be a modified Support Vector Regression, where the kernel has been learned by solving a semidefinite programming (SDP) and quadratically constrained quadratic programming (QCQP) problem. The SVR is further modified by scaling the training points in the SDP and QCQP problems by their relevance and importance to the examined regression. The result is a weighted average of different regressions depending on how much a single regression is likely to contribute, where advantages include reduced problem complexity, specificity with regard to image content, added degrees of freedom from a nonlinear approaches, and excellent generalization that a combined methodology has over its individual counterparts.

Index Terms— Interpolation, Nonlinear functions

1. INTRODUCTION

Single image superresolution is the process of using added, assumed, or made-up information in combination with a single low-resolution image to produce a high-resolution image. This information can come in many forms, including but not limited to a set of shifted low-resolution images of a single scene, assumed relationships between existing pixel values and edges [1], a training set, or any other data that would aid in enhancing visual acuity.

Our algorithm is concerned with maximizing information inherent within a training set of known input-output image pairs, while also considering imaging properties including the similarity of our domain to the range. This work is most similar to [2], though the concept of operating in the range space has been originally identified in bilateral filtering [3] and since improved in several other papers. In a series of previous works [4, 5, 6], techniques involving support vector regression (SVR) are examined and improved for prediction purposes of unknown high-resolution values based on low-resolution image patches. The novelty in [6] and this work is that, in addition to added flexibility by the nonlinear regression in SVR, classification is introduced, further improving generalization and training ability.

The stochastic framework combining classification and regression is taken from [2], where the conditional expectation of high-resolution values given low-resolution patches determines a regression drawing from a "semi-segmented"-domain. This type of segmented regression allows for a variety of content without losing specificity to unique inputs. For image processing, this property is especially advantageous because methods such as [3] are unable to match the variety that the proposed algorithm obtains through classification, while linear filtering in [2] provides less accuracy and specificity that is available in scalable nonlinear regression via SVR.

The remainder of this paper focuses on the capacity of the proposed algorithm to provide color image superresolution and is organized as follows. Section 2 briefly reviews and subsequently optimizes the support vector machine for regression by using well-known kernel learning techniques. Section 3 explains the incorporation of SVR into a classification framework and the necessary modifications needed to do this. Section 4 investigates the extension of the proposed algorithm from gray-scale to color images, including the explanation of which features to use and in what manner they are considered.

2. OPTIMAL KERNELS FOR SUPPORT VECTOR REGRESSION

The support vector machine (SVM), originally proposed in [7], is a supervised learning technique that determines a high-dimensional functional from a training set Ω,

$$\Omega = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), ..., (\mathbf{x}_N, y_N)\}. \quad (1)$$

The goal of SVR is to use relationships learned through Ω, and be able to generalize these relationships to unseen test points. In (1), $\mathbf{x}_i \in \mathbb{R}^n$ and $y_i \in \mathbb{R}, \forall i \in [1, N]$, and SVR estimates the function $f : \mathbf{x} \rightarrow y$ by $\hat{y} = \mathbf{w} \cdot \phi(\mathbf{x}) + b$ in the following optimization:

$$\min_{\mathbf{w},b,\xi} \left(\frac{1}{2}||\mathbf{w}||^2 + C \sum_{i=1}^{N} \left(\xi_i^+ + \xi_i^-\right) \right)$$

subject to

$$(\mathbf{w} \cdot \phi(\mathbf{x}_i) + b) - y_i \leq \varepsilon + \xi_i^+$$
$$y_i - (\mathbf{w} \cdot \phi(\mathbf{x}_i) + b) \leq \varepsilon + \xi_i^-$$

and

$$\xi_i^-, \xi_i^+ \geq 0, \ \forall\, i \in [1, N] \quad (2)$$

The high-dimensional mapping $\phi : \mathcal{X} \mapsto \mathcal{F}$ in (2) often better suits a representation of complicated relationships which could otherwise

This work is supported in part by a grant from Qualcomm, Inc., and matching funds from the U.C. Discovery Program

not be linearly realized. Within $\mathcal{F}$, a kernel function $K(\mathbf{s},\mathbf{t})$ written as a kernel matrix $K(\cdot,\cdot)$ is defined to be a collection of dot products for an arbitrary ϕ that may or may not be known. With this in mind, the dual to (2) can be found and is written in (3).

$$\max_{\alpha^+,\alpha^-} -\frac{1}{2}\sum_{i,j}\left\{(\alpha_i^+ - \alpha_i^-)(\alpha_j^+ - \alpha_j^-)K(\mathbf{x}_i,\mathbf{x}_j)\right\} \\ -\epsilon\sum_i(\alpha_i^+ + \alpha_i^-) + \sum_i y_i(\alpha_i^+ - \alpha_i^-)$$

subject to

$$\sum_i(\alpha_i^+ - \alpha_i^-) = 0 \text{ and } 0 \le \alpha_i^{+/-} \le C$$

with the solution hyperplane as

$$g(\mathbf{x}) = \sum_i(\alpha_i^+ - \alpha_i^-)K(\mathbf{x},\mathbf{x}_i) + b \tag{3}$$

where a dot product in $\mathcal{F}$ is defined by $K(\mathbf{s},\mathbf{t}) = \phi(\mathbf{s})\cdot\phi(\mathbf{t})$, the kernel function.

Using the kernel matrix K, computational complexity is reduced because determining $d = \phi(\mathbf{s})\cdot\phi(\mathbf{t})$, which is quite often intractable, is unnecessary when solving (3). This definition also allows ϕ to be unknown, in which case, K can be conceptually chosen to be a desired similarity metric depicting the "nearness" of two vectors. Thus, the selection of the kernel matrix K becomes important and should be sensitive to the training data. This section describes learning an optimal K by using semi-definite programming (SDP) and quadratically constrained quadratic programming (QCQP) problems, which were initially derived in [5].

2.1. The SDP Problem

To simplify, allow e to be a vector of all ones, and

$$\alpha^+ + \alpha^- = \beta^+ \\ \alpha^+ - \alpha^- = \beta^- \tag{4}$$

Also, let the final, optimized kernel be defined by $K_{opt} = \sum_i \mu_i^* k_i$, a linear combination of known, calculated kernels k_i weighted by elements in $\boldsymbol{\mu}^*$. While full derivations can be seen in [5], we are able to determine the optimal kernel K_{opt} by using the aforementioned substitutions in (4) for K_{opt}, writing the Lagrangian, taking the dual of (3), and using the Schur complement lemma. This yields the final optimization problem shown in (5).

$$\begin{aligned}\min_{\boldsymbol{\mu},t,\lambda,\nu_u^+,\nu_l^-,\nu_u^-} \quad & t \\ \text{s.t.} \quad & \begin{pmatrix} 2K & \gamma \\ \gamma^T & t - 2Ce^T(\nu_u^+ + \nu_u^-) \end{pmatrix} \succeq 0 \\ & \nu_u^+, \nu_u^-, \nu_l^- \succeq 0 \\ & \epsilon e + \nu_u^+ + \nu_u^- - \nu_l^- \succeq 0 \\ & \sum_i \mu_i k_i \succeq 0 \\ & trace(\sum_i \mu_i k_i) = c \end{aligned} \tag{5}$$

This result is general and can be applied to any regression problem.

2.2. The QCQP Problem

The QCQP arises from an added constraint, $\mu_i \ge 0$, which loses some generality, though it does ensure positive definiteness when inductively applying the learned kernel. On the other hand, the complexity of the kernel is never simplified because the positive eigenvalues of each $(\mu_i k_i)$ will never reduce kernel rank.

The formulation we obtained is derived in the same manner as [8], and the full solution is shown in [5], and is given in (6).

$$\begin{aligned}\max_{\beta^+,\beta^-,p} \quad & 2y^T\beta^- - 2\epsilon e^T\beta^+ - cp \\ \text{s.t.} \quad & p \ge \beta^- k_i \beta^- \\ & e^T\beta^- = 0 \\ & 0 \preceq \beta^+ + \beta^- \preceq 2C \\ & 0 \preceq \beta^+ - \beta^- \preceq 2C \end{aligned} \tag{6}$$

The μ_i values come out of the dual Lagrangian variables.

3. STOCHASTIC FRAMEWORK FOR A COMBINATIONAL TECHNIQUE

The proposed algorithm partitions the problem into smaller, more solvable problems according to data similarity. Therefore, for super-resolution, like image patches are treated similarly, and the notion of different *types* of image content such as edges, texture, etc., is introduced and utilized. These *types* of content can be realized as classes if the data were to be classified. With this terminology in mind, the overall process can be described by *1.)* using an initial preprocessing stage where the probabilities of an input being in particular classes are found, *2.)* obtaining high-resolution outputs by using a modified version of the regressions described in Sec. 2, and finally *3.)* summing the result of each regression weighted by the probability that it is in a given class. With certain assumptions on the representation of our SVR, this procedure conceptually describes the expected value of the output as will be demonstrated in the remainder of this section.

3.1. Clustering Image Content

Let I_{LR} and I_{HR} be the low and high-resolution image patches with sizes $D \times D$ and $U \times U$ respectively. To superresolve the center pixel of I_{LR} by a factor of U, we define vectors

$$\begin{aligned}\mathbf{x} &= \text{vectorize}(I_{LR}) - \text{center pixel}(I_{LR}) \in \mathbb{R}^{D^2\times 1} \\ \mathbf{y} &= \text{vectorize}(I_{HR}) - \text{center pixel}(I_{LR}) \in \mathbb{R}^{U^2\times 1}\end{aligned} \tag{7}$$

in a given training set Ω_c of $\mathbf{x}_i$ feature and $\mathbf{y}_i$ label pairs. The subscript c in Ω_c denotes the classification training set.

As in [2], to generalize over all image content, we assume that we can model image patches by a Gaussian mixture model (a sum of scaled Gaussians). By adding a level abstraction, we can think of each Gaussian in the Gaussian mixture as a particular class, believing that like images tend to cluster. Therefore, a numerical quantity can be extrapolated for the probability that an image patch $\mathbf{x}$ belongs in class j by evaluating $P(\mathbf{x}|J = j) = \mathcal{G}(\mathbf{x},\boldsymbol{\mu},\Sigma)$, where J is the random variable denoting the class of $\mathbf{x}$, and the parameters $\boldsymbol{\mu}$ and Σ are obtained through Expectation Maximization (EM).

The usage of this quantity becomes clear because the framework of our solution $g(\mathbf{x})$ is derived from the conditional expectation of

our output, the high-resolution pixels, given the input, low-resolution patches, and training data. This is expressed in (8).

$$\begin{aligned} g(\mathbf{x}) &= E[\mathbf{y}|\mathbf{x}, \Omega_c] \\ &= \sum_j E[\mathbf{y}|\mathbf{x}, J=j]P(J=j|\mathbf{x}) \end{aligned} \tag{8}$$

where $g(\mathbf{x})$ uses the training set Ω to estimate $f : \mathbf{x} \mapsto \mathbf{y}$.

Using Bayes' law, the value of $P(J=j|\mathbf{x})$ can be obtained:

$$P(J=j|\mathbf{x}) = \frac{P(\mathbf{x}|J=j)P(J=j)}{\sum_j P(\mathbf{x}|J=j)P(j=j)} \tag{9}$$

Using the probability of a given image patch $\mathbf{x}$ lying in a particular class, a smaller regression results, considerably simplifying the effort because instead of dealing with the entire space of image patches, we can concentrate on a specific class of image content. That is to say, the expected output, $E[\mathbf{y}|\mathbf{x}, J=j]$, is easier to calculate than $E[\mathbf{y}|\mathbf{x}]$, where the information $J=j$ is not provided.

3.2. Modified Support Vector Regression

The class conditional expectation, $E[\mathbf{y}|\mathbf{x}, J=j]$, in (8) is more difficult to estimate. Our proposed algorithm attempts a straightforward and scalable technique of approximating it by using a regression device, i.e. $E[\mathbf{y}|\mathbf{x}, J=j] = g_j(\mathbf{x})$.

There are situations where it is undesirable to use the same input for classification as the input for regression. For example, if we wish to cluster on 3×3 features (i.e. $\mathbf{x} \in \mathbb{R}^{9\times 1}$), but SVR can offer a good solution only if its input is of larger dimension (e.g. 5×5), then the g_j would benefit if its domain were to reflect this. That is to say, instead of g_j taking in $\mathbf{x}$, we can use $\mathbf{z}$ instead, where $\mathbf{z}$ promotes flexibility in the choice of domain. In the case of our proposed algorithm, for luminance prediction, $\mathbf{z} \in \mathbb{R}^{25\times 1}$. In terms of the new substitutions, the solution becomes

$$E[\mathbf{y}|\mathbf{x}, \mathbf{z}] = \sum_j g_j(\mathbf{z})P(J=j|\mathbf{x}), \tag{10}$$

and our approach is to apply SVR to estimate $g_j(\mathbf{x})$.

For good generalization, more discretion is needed when considering the relevancy of training points. In other words, the more likely a point is in a class, the more the SVR should consider the point in the regression. From the dual problem in (3), the weighting of training points by their importance is analogous to the effect of C on the solution hyperplane. The C variable is actually a cost parameter whose value comes out of cross validation. In the dual problem, the larger the cost parameter, the more the $\alpha^{+/-}_{(i,j)}$ values can deviate for an exact regression, in effect granting freedom to closely fit the training data in exchange for flatness in the objective function. Therefore, for pair $(\mathbf{z}_i, y_i) \in \Omega_r$, limiting $\alpha^{+/-}_{(i,j)}$, also limits the effect of the i^{th} point on the solution hyperplane. In terms of the primal problem in (2), C scales the slack variables $\xi^{+}_{(i,j)}$ and $\xi^{-}_{(i,j)}$, restricting the quantity of points deviating from the solution hyperplane and by how much these points deviate.

Our answer is to scale each ξ_i for all points in Ω_r (subscript r for regression) by how relevant the i^{th} point is to class j. This can be done with the product of all $\xi^{+/-}_{(i,j)}$ with their corresponding posterior probabilities P_{ij}. So, for the j^{th} regressor, the primal optimization problem is described by

$$\min_{\mathbf{w}_j, b} \frac{1}{2}\|\mathbf{w}_j\|^2 + C \cdot \overrightarrow{P_{j|i}}\,(J=j|\mathbf{x}_i)^T\,(\overrightarrow{\xi^{+}_j} + \overrightarrow{\xi^{-}_j})$$

subject to

$$\begin{aligned} y_i - (\mathbf{w}_j \cdot \phi(\mathbf{z}_i) + b_j) - \epsilon &\le \xi^{+}_{(i,j)} \\ (\mathbf{w}_j \cdot \phi(\mathbf{z}_i) + b_j) - y_i - \epsilon &\le \xi^{-}_{(i,j)} \\ \xi^{-}_{(i,j)}, \xi^{+}_{(i,j)} &\ge 0 \end{aligned} \tag{11}$$

where $\overrightarrow{P_{j|\mathbf{x}_i}}\,(J=j|\mathbf{x}_i)$ denotes a vector of length N containing the posterior probabilities of classes. This way, we can allow more slack for the variables that are less important (i.e. have smaller posterior probabilities.) The probability vector in (11) can be equivalently placed throughout the rest of the derivations in Sec. 2.

This solution has the potential to consume extensive computation both in CPU cycles and memory, and so a simplification of the problem would be to consider per class those points which meet a certain criterion with respect to their respective posterior probabilities. This is implemented by partitioning Ω_r into $\{\Omega_j\}$ and considering the i^{th} point for class j only if its posterior probability exceeds a certain threshold. Granted, this simplification rejects considerable amounts of data per class. Nevertheless, if SVR can indeed predict the relationship between low and high-resolution, then the regression may be sufficient for less relevant points in a given class. Furthermore, through experimentation, it turns out that only a few classes at any given test point are chosen and used for reconstruction the majority of the time. The implication from this is that for the test point $\mathbf{x}_{test}$, by multiplying $P_{j|\mathbf{x}}(j|\mathbf{x}_{test})$ with $g_j(\mathbf{x}_{test})$ in (10), we would maintain good accuracy by zeroing out test data that is irrelevant for a particular class anyway, leaving reconstruction for those classes which can accurately do so.

4. COMPONENT SUBSAMPLING AND COLOR SUPERRESOLUTION

The extension to color images is examined in this section. Color images are usually partitioned into three components. The simplest prediction technique is independent component interpolation. There are obvious disadvantages to this method, most notably the disregard for inherent correlation between components. One readily available remedy for this issue and those similar to it would be to use values from all three components to produce an evenly proportioned feature representative of all three components. However, using 5×5 windows in 3 different spaces means 75 dimensions, and estimation errors overcome whatever is gained by the added information.

Therefore, we need to maintain balance by trading off small feature vector size for a decent amount of quality information. As it turns out, in terms of the human visual system (HVS), changes in color Cr and Cb components are less detectable, and perceptual changes in luminance seem more important. In fact, all MPEG compression use a 4:2:0 resolution format, where luminance pixels outnumber either chrominance component by a factor of 4. Drawing from this, many techniques that use RGB interpolation (including [2]) weight the importance of each component by their average proportion of luminance. Our proposed algorithm is more direct in its approach and clusters luminance components only, disregarding color altogether. The rationale behind this thinking is that for purposes of image content recognition, particular objects may be tinted differently when the underlying texture as well as the transition of colors within the patch remain the same.

While clustering luminance components in $\mathbf{x}$, color regressions use a separate input $\mathbf{z}$ from a window of surrounding color components (either Cb or Cr). It is here that SVR has distinct advantages over the linear filtering used by most interpolation algorithms. By

filtering edges, halos or odd-colored auras often appear along texture transitions and borders. This is due in part to the averaging of pixel values, which smoothing linear filters have a tendency to do. Because there exists a multitude of shades of colors between any two chroma values, averaging often produces these unnatural and strange colors. The proposed algorithm avoids these undesirable byproducts that typically plague linear algorithms because the choice of training set often excludes these in-between values. The exclusion leaves out the odd-looking colors by effecting texture transitions that occur naturally in the image pairs of the training set.

5. RESULTS AND ANALYSIS

The SDP and QCQP problems in Sec. 2 have been verified in a previous work [5] using the cvx [10] Matlab toolbox. Image superresolution algorithms compute the QCQP and utilize the MOSEK toolbox [11].

The algorithm was set up with $D = 3$ and $U = 2$, meaning that I_{LR} was 3×3 and I_{HR} was 2×2. Luminance and chrominance regression features come from windows of sizes 5×5 and 3×3, respectively. Thus, $\mathbf{x} \in \mathbb{R}^{9\times 1}$, $\mathbf{y} \in \mathbb{R}^{4\times 1}$, $\mathbf{z}_{Lum} \in \mathbb{R}^{25\times 1}$, $\mathbf{z}_{Cr} \in \mathbb{R}^{9\times 1}$, and $\mathbf{z}_{Cb} \in \mathbb{R}^{9\times 1}$. We ensure good generalization by randomly selecting a test image that differs from a training set of 20 images. For fair comparison, the same training set is used for any relevant learning algorithms involving a training set (most of which can be seen at the website below) to which we compare our method.

Comparisons to new edge-directed interpolation (NEDI) [1] and bicubic interpolation are shown qualitatively in Fig. 1. The proposed method Fig. 1(d) offers more clarity than all the compared methods. Imperfections in Fig. 1(c) could be a by-product of a 2×2, two pass system in which [1] considers features and color independently. Experimental results assert that joint consideration is advantageous because optimization described by Sec. 2 results in a kernel with all 3×3 features for most classes. Additional images and comparisons can be found at research pages on UCSD's video processing website: `http://videoprocessing.ucsd.edu/~karl/color_krs_sr`

(a) Original

(b) Bicubic Interpolation

(c) Edge Directed Interpolation

(d) Combinational Classification/Regression

Fig. 1. Image Comparisons of Various Methods

6. CONCLUSIONS

This work has proposed an approach to single image superresolution that successfully offers a stochastic framework involving two learning techniques. In addition, the superresolution algorithm has extended regressions to the color domain based on luminance classification and prevented the ill-effects of bleeding color into surrounding edge areas. The exploitation of the proposed statistical method offers good numerical and visual results.

7. REFERENCES

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

[2] C. Brian Atkins and C. Bouman, *Classification based methods in optimal image interpolation*, Ph.D. thesis, Purdue University, 1998.

[3] Carlo Tomasi and Roberto Manduchi, "Bilateral filtering for gray and color images," in *ICCV*, 1998, pp. 839–846.

[4] Karl Ni, Sanjeev Kumar, T. Q. Nguyen, and N. Vasconcelos, "Single image superresolution based on support vector regression," in *International Conference on Acoustics, Speech, and Signal Processing*, 2006.

[5] Karl Ni, Sanjeev Kumar, and Truong Q. Nguyen, "Learning the kernel matrix for superresolution," in *IEEE Conference on Multimedia Signal Processing*, October 2006.

[6] Karl Ni and T. Q. Nguyen, "Kernel resolution synthesis for superresolution," in *International Conference on Acoustics, Speech, and Signal Processing*, to appear in May 2007.

[7] V. Vapnik, *The Nature of Statistical Learning Theory*, Springer-Verlag, New York, 1995.

[8] Gert R. G. Lanckriet, Nello Cristianini, Peter Bartlett, Laurent El Ghaoui, and Michael I. Jordan, "Learning the kernel matrix with semidefinite programming," *Journal of Machine Learning Research*, vol. 5, pp. 27–72, 2004.

[9] S. Qiu and T. Lane, "Multiple kernel learning for support vector regression," Tech. Rep., University of New Mexico, 2005.

[10] Michael Grant, Stephen Boyd, and Yinyu Ye, *CVX: Matlab Software for Disciplined Convex Programming*.

[11] *The MOSEK optimization toolbox for MATLAB manual. Version 4.0 (Revision 16)*, 2006.