A COUPLED FEATURE-FILTER CLUSTERING SCHEME FOR RESOLUTION SYNTHESIS

Toygar Akgun and Yucel Altunbasak

Center for Signal and Image Processing
Georgia Institute of Technology
Atlanta, Georgia, 30332-0250

ABSTRACT

For many digital image/video processing applications increasing the spatial resolution through modifications in the imaging system is infeasible. Hence post-processing algorithms designed to enhance resolution of the acquired image data prove beneficial. In this paper, we analyze recent work on classification based resolution enhancement and discuss its applicability to low-complexity display systems. In the light of our observations we point out certain short-comings of resolution synthesis and propose a modified training scheme to improve the performance under certain conditions.

Index Terms**—** Single-frame resolution enhancement, resolution synthesis, scaling

1. INTRODUCTION

For many digital image/video processing applications increasing the spatial resolution is not only desirable but also highly beneficial. At higher resolution, TV pictures look more natural and pleasing to the eye, computer vision tasks such as object detection and tracking can be performed with higher precision, medical diagnoses can be made with a higher confidence, security cameras can offer better identification, and satellite imagery can be interpreted with higher accuracy. As such, spatial resolution is an influential parameter in many mainstream imaging applications, and resolution enhancement task naturally arises as a means of increasing the effectiveness of any imaging system used in the mentioned applications. In this work, we analyze recent work on off-line training based resolution enhancement, namely resolution synthesis by Atkins *et. al.* [1]. We discuss its applicability to low-complexity display systems in terms of visual quality and computational complexity. In the light of our observations we point out certain short-comings of resolution synthesis and propose a modified training scheme to improve the performance under certain conditions.

Once a digital image is captured, the frequency content of the image is limited by the resolving power of the image acquisition system, which is a function of the density of the sensor array and the imaging optics. Scaling an image by linear shift invariant (LSI) filtering can not bring back the high frequency components degraded (reduced to noise level, completely filtered out or aliased) during sampling. This is where resolution enhancement differs from scaling. Resolution enhancement methods can be interpreted as advanced scaling techniques that can recover the missing or aliased high frequency components to a limited extend. Single-frame resolution enhancement techniques can estimate the missing high frequency components to a limited extend through spatially adaptive filtering and use of *prior information*. The main improvement offered by single frame resolution enhancement is observed around edges and textured areas. Compared to the results obtained by LSI scaling filters such as bicubic interpolation combined with unsharpen filtering, techniques such as the resolution synthesis algorithm can offer much smoother, continuous edges with sharp transitions, remove the blurry look from textured areas and rectify slight aliasing artifacts (where aliased signal components can not disturb the dominant spatial structure). By fusing information embedded in multiple aliased frames multi-frame resolution enhancement techniques can further improve the spatial resolution, bringing back missing details, rectifying heavier aliasing.

Resolution enhancement is an inherently ill-posed problem that requires extra information. In case of multi-frame resolution enhancement, typically referred to as superresolution, extra information is mainly extracted from multiple aliased observations. In case of single-frame resolution enhancement we do not have access to multiple frames, hence we are bound to use prior information. Prior information can be in the form of *a priori* distributions in the Bayesian framework or regularization terms in the deterministic approach. Another way of utilizing prior information is to learn a group of spatial structures (which we refer to as context classes) frequently observed in natural images and observe the way they are distorted during high resolution to low resolution conversion (sampling or down-sampling). There are at least two well-known single-frame resolution enhancement algorithms that utilize prior information in this format, namely resolution synthesis proposed by Atkins *et. al.* [1] and example-based super-resolution by Freeman *et. al.* [2]. Resolution synthesis (RS) is based on pixel classification and adaptive linear filter-

Support for this project has been provided by VESTEL Corporation

1-4244-1437-7/07/$20.00 ©2007 IEEE

ing, and allows for efficient hardware implementation. Since, our goal is to design a low-complexity resolution enhancement method that can be implemented in the next generation display systems, we focus on the RS algorithm.

The rest of the paper is organized as follows. Section 2 provides an overview of the RS algorithm, and discusses its applicability to customer grade flat panel displays. Section 3 details the proposed modifications to the training scheme, and finally Section 4 presents visual results.

2. RESOLUTION SYNTHESIS

RS is based on the assumption that pixels in natural images can be classified as belonging into a limited number of context classes. These context classes are defined by small pixel neighborhoods that exhibit visually identifiable spatial structures. To get a better grip on the idea, note that natural images are structured signals with much less variability than completely random images. These regularities typically observed in natural images [3] can be exploited in resolution enhancement problem. However, we recognize the fact that unless we are working on a highly restricted set of images with very specific training data, it is not possible to generate the *true* high-resolution signal components. Hence, we focus on generating *visually plausible* image details, such as sharp edges without disturbing jaggies, and plausible looking texture.

Next we briefly introduce the RS algorithm, which consists of two phases, namely, training and filtering, as shown in Figure 1. Training phase essentially computes the interpolation filters and the parameters of a Gaussian mixture model used to associate the low resolution pixels with all of the context classes to different degrees of membership. Filtering phase starts with computing the membership degrees of the input pixel. The output high resolution pixel values are then computed as a weighted linear combinations of the results computed by all filters, where the combination weights are the membership degrees. A detailed analysis of resolution synthesis can also be found in [4], [5].

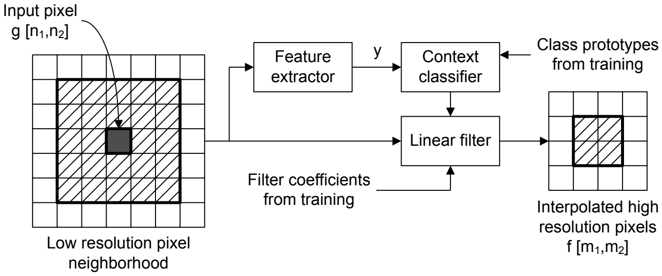
Input pixel
g [n1,n2]
Feature extractor
y
Context classifier
Class prototypes from training
Linear filter
Filter coefficients from training
Low resolution pixel neighborhood
Interpolated high resolution pixels
f [m1,m2]

Fig. 1. Resolution synthesis algorithm.

Every low resolution pixel $\boldsymbol{g}[n_1, n_2]$ is assumed to be from a context class that best explains the spatial structure within a small local neighborhood centered at $\boldsymbol{g}[n_1, n_2]$. Both the training and filtering phases are based on classifying the input pixel. In an effort to reduce computational complexity and enhance discrimination performance, every low resolution pixel is represented by a feature vector $\boldsymbol{y}$ extracted from a 3×3 neighborhood. Pixel classification is performed on the feature vectors, instead of using all pixels in the neighborhood. To extract the feature vector we first obtain a 8×1 vector $\tilde{\boldsymbol{y}}$ by subtracting $\boldsymbol{g}[n_1, n_2]$ from its neighbors. Then the feature vector $\boldsymbol{y}$ is computed as the normalized version of $\tilde{\boldsymbol{y}}$,

$$\boldsymbol{y} = \begin{cases} \tilde{\boldsymbol{y}} / \parallel \tilde{\boldsymbol{y}} \parallel^{-p}, & \tilde{\boldsymbol{y}} \neq \boldsymbol{0} \\ \boldsymbol{0}, & \text{otherwise} \end{cases} \tag{1}$$

where $0 \leq p \leq 1$ is a parameter that controls the amount of normalization. p is typically chosen as 0.75, [4]. The function that maps the input low resolution neighborhood to the feature vector is of great importance since the feature vectors has great influence on the visual performance of the algorithm. The feature vectors are modeled as random vectors drawn from a multivariate Gaussian mixture with M mixture classes, where every Gaussian mixture class corresponds to a context class. The expectation-maximization (EM) algorithm is applied to compute the maximum likelihood (ML) estimates of the Gaussian mixture parameters, namely the class means (μ_i), standard deviations (σ_i) and mixture probabilities (π_i). Once EM converges, we can compute the probability that any given feature vector belongs to a mixture (context) class. If these probabilities are interpreted as memberships than the resulting mixture model provides a fuzzy clustering of the feature vectors in the training set. New input pixels are classified by computing the probabilities that their feature vectors are drawn from a context class (Gaussian mixture classes). Derivation of the RS algorithm is based on the following assumptions:

Assumption 1: Feature vectors are modeled as a multivariate Gaussian mixture,

$$p_Y(\boldsymbol{y}) = \sum_{j=1}^{M} \pi_j p_{Y|J}, \qquad p_{Y|J} \sim \mathcal{N}(\mu_j, \sigma^2 \boldsymbol{I})$$

where j is the mixture class index.

Assumption 2: Given the input low resolution pixel neighborhood and the context class, the high resolution pixels are Gaussian

$$p_{F|G,J}(\boldsymbol{f}|\boldsymbol{g}, j) = \mathcal{N}(\boldsymbol{A}_j \boldsymbol{g} + \boldsymbol{\beta}_j, \sigma^2 \boldsymbol{A}_j^T \boldsymbol{A}_j).$$

Assumption 3: Given feature vector y, the class distribution is independent of the high resolution and low resolution pixels

$$p_{J|F,G}(j|\boldsymbol{f}, \boldsymbol{g}) = p_{J|Y}(j|\boldsymbol{y}).$$

Under these assumptions the MMSE estimator is [4]

$$\hat{\boldsymbol{f}} = \sum_{j=1}^{M} (\boldsymbol{A}_j \boldsymbol{g} + \beta_j) \underbrace{\frac{\pi_j \exp(-\frac{1}{2\sigma^2} \parallel \boldsymbol{y} - \boldsymbol{\mu}_j \parallel^2)}{\sum_{i=1}^{M} \pi_i \exp(-\frac{1}{2\sigma^2} \parallel \boldsymbol{y} - \boldsymbol{\mu}_j \parallel^2)}}_{w_j}. \tag{2}$$

From Eq. 2 we can see that final high resolution pixel estimates are computed as a weighted linear combination of the estimates for all context classes.

In its current form, resolution synthesis is computationally too demanding for systems with limited computational resources and memory. The high computational load is mainly due to the large number of classes required for satisfactory performance (typically anywhere between 30-100) and the requirement for weighted combination (soft filtering). Linear combination is especially demanding since it requires repeating application of a 5×5 filter, implying 25 additional multiplications and an additional accumulation for every class included in soft filtering. In addition, the combination weights (w_j's in Eq. 2) must be computed to obtain the final result. Although Atkins proposes several complexity reductions in [4], these modifications do not allow for efficient hardware implementations and require large amounts of on-board memory. We observed that directly reducing the number of classes (below ~ 30) severely degrades performance. Also using only one class (the class with maximum membership) to compute the high resolution pixels resulted in degraded performance. We found out that the discrimination power of the feature vectors defined by Eq. 1 was severely degraded as the number of context classes was reduced below ~ 25. Features extracted from 5×5 neighborhoods with a modified extraction rule proved to have much higher discrimination power and performed much better under slight aliasing. These shortcomings render resolution synthesis useless for customer grade flat panel displays, where the computational complexity must be kept below some threshold. *Our goal is to introduce some modifications so that RS can operate satisfactorily with as low as 11 context classes using hard decision (using a single class in filtering).*

3. PROPOSED ITERATIVE TRAINING SCHEME

Proposed training method is shown in Figure 2, it is based on the observation that interpolation filter design stage direct access to the high resolution pixels. We note that due to Assumption 3 given in Section 2 clustering with respect to feature vectors (distribution parameter estimation) is completely uncoupled with filter design and high resolution pixels are only utilized by interpolation filter design block, which is executed only once after the convergence of EM algorithm. Hence, if we can couple interpolation filters to the feature extraction and classification stages, the resulting clustering should improve. Given the low and high resolution training images, proposed method iteratively extracts the best interpolation filters and the context class prototypes that are used to determine input pixel's context. The iterative training works as follows.

0. Initialization

After extracting the feature vectors of all the low resolution pixels in the training set, class prototypes are initialized randomly. The prototype for class number 1 is manually set to a vector of all zeros. This guarantees that we have a class number 1 reserved for uniform areas. All covariance matrices are set to identity matrices.

1. Clustering with respect to features

After initialization, the low resolution pixels are classified with respect to their feature vectors, Block 1 in Figure 2. This is done by going through all low resolution pixels, computing the weighted Euclidian distance (the weighting matrix is the inverse of feature covariance matrix) between the pixel's feature vector, which is a representative of the local image characteristic of the low resolution pixel and the cluster prototypes, which are representatives of different context classes. Then the input low resolution pixel is labeled with the index of the cluster whose feature vector is the closest to the low resolution pixel's feature vector.

2. Filter update

Once the low resolution pixels are clustered with respect to their feature vectors (context) the interpolation filters for all clusters are updated with the filter that minimizes the mean-squared-error between the interpolated and the true high resolution pixels computed for all low resolution pixels in a specific cluster, Block 2 in Figure 2. Tikhonov regularization can be used to avoid filters that excessively amplify the high frequency components. While preparing the training samples, a small amount of blurring prior to downsampling is necessary to model the camera response and also to avoid aliasing. But completely filtering out the high frequency components effectively creates an inverse problem where the filters are asked to bring back completely removed signal components (this is only possible in multi-frame case), resulting in bad filters.

3. Clustering with respect to filters

After filter update, all the input pixels are clustered with respect to the minimum mean-squared-error interpolation filter, Block 3 in Figure 2. This is accomplished by going through all low-resolution training pixels, computing the interpolated high resolution pixels by all interpolation filters one by one, and comparing the interpolated pixels to the available high resolution pixels. The low resolution pixel is then labeled with the index of the interpolation filter that gives the minimum mean-squared-error between the interpolated and real high resolution pixels.

4. Class prototype update

Once all the input pixels are classified, the feature vectors of the obtained clusters are updated one by one, Block 4 in Figure 2. This update can be done in various different ways such as taking the average of the median of the feature vectors. Class covariance matrices are updated next. To reduce computational complexity, we assume diagonal covariance matrices. Then we go back to clustering with respect to features, and iterate in this fashion for predetermined times. In our experiments we worked with 2 iterations.

Once the filter coefficients and the feature vectors of all contexts are learned from training data, these parameters are

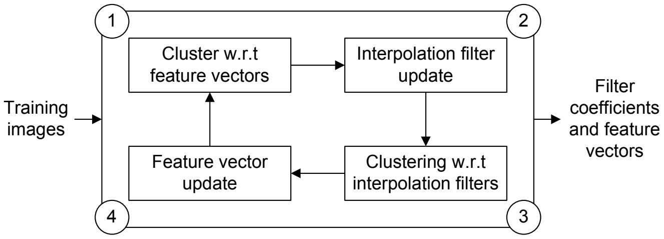
1
2
Cluster w.r.t feature vectors
Interpolation filter update
Training images
Filter coefficients and feature vectors
Feature vector update
Clustering w.r.t interpolation filters
4
3

Fig. 2. Proposed training scheme.

passed to the interpolation stage. For a given input pixel, first the feature vector is extracted and the pixel is hard classified. High resolution output pixels are obtained by a single filtering operation using the corresponding optimal filter.

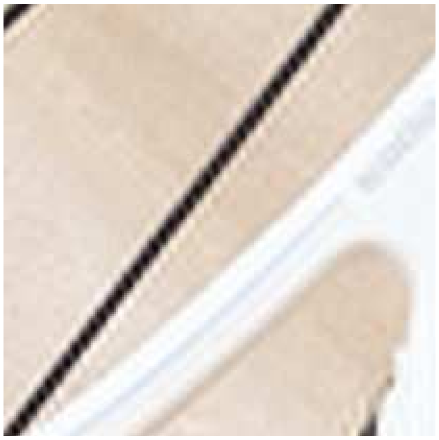

Fig. 3. Original resolution synthesis

4. COMMENTS AND RESULTS

In our experiments (for scaling ratios of 1.5 and 2) we observed that the proposed training scheme can provide satisfactory visual results with as low as 11 classes and hard decision. Regions cropped from the results obtained for a scaling ratio of 2 are presented in Figures 3 and 4. The original RS algorithm was trained with 11 classes and all 11 classes were used in the weighted combination. Proposed method was also trained with 11 classes, but filtering was done with hard decision (only one filter was used). Both algorithms were trained on the same training set which consisted of approximately 250000 low-high resolution pixel pairs. We implemented a fixed point version of the proposed algorithm on an entry level FPGA (Spartan 3 from Xilinx) and the details of this implementation will be reported in another publication.

Due to limited space we will not be able to present a detailed analysis of the proposed method here. We note that further improvements over the algorithm detailed in this paper are possible. It should be clear to the reader that clustering with respect to the filters and clustering with respect to the features are two different goals which may not agree for a specific choice of features and filter clustering method. Although it is possible to come up with a feature extraction method and a way of clustering pixels with respect to the best filter that agree for an arbitrarily large percentage of training pixels, finding such schemes is not straightforward. We have observed that for the current implementation increasing the number of iterations corrupted the interpolation filters and the class prototypes. Through exhaustive computer simulations we have concluded that clustering with respect to interpolation filters based on minimum MSE is the main reason that avoids convergence. Pixels in uniform areas are frequently assigned to wrong context classes due to their lack of structure (almost all filters perform good). Additional regularization terms are required to make clustering with respect to filters more robust.

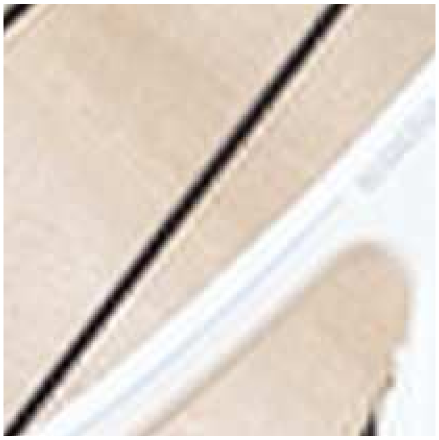

Fig. 4. Proposed

5. REFERENCES

[1] C.B. Atkins, C.A. Bouman, and J.P. Allebach, “Optimal image scaling using pixel classification,” in *IEEE Proceedings of the International Conference on Image Processing, 2001*, Oct 2001, vol. 3, pp. 864–867.

[2] W.T. Freeman, T.R. Jones, and E.C. Pasztor, “Example-based super-resolution,” *IEEE Computer Graphics and Applications*, vol. 22, no. 2, pp. 56–65, March 2002.

[3] E. P. Simoncelli, “Statistical models for images: Compression, restoration and synthesis,” in *31st Asilomar Conf. on Sig., Sys. and Computers*, Pacific Grove, CA, October 1997.

[4] B. Atkins, *Clasification based methods in optimal image interpolation*, Ph.D. thesis, Purdue University, 1998.

[5] R. Yoakeim and D. Taubman, “Quantitative analysis of resolution synthesis,” in *IEEE Proceedings of the International Conference on Image Processing, 2004*, Oct 2001, vol. 3, pp. 1645–1648.