Polyphase interpretation of empirical image interpolation

POLYPHASE INTERPRETATION OF EMPIRICAL IMAGE INTERPOLATION

Karl Ni[1,2] *and Truong Q. Nguyen*[1]

The University of California at San Diego[1], Massachusetts Institute of Technology, Lincoln Laboratories[2]

ABSTRACT

We observe several characteristics of empirical image interpolating algorithms and contribute four novel concepts and claims. First, we interpret well-known classification-based filtering algorithms in terms of their polyphase components. We examine the underlying principles behind the various fixed-scale linear interpolating kernels. Second, we conceptually extend the properties of the multiple filters to two dimensions to analyze frequency domain characteristics common to all empirically-designed interpolating filters. Third, we propose a general linear filter for image interpolation, which uses a universal magnitude response and zero-phase. Finally, the proposed filter is further generalized to support arbitrary scaling factors. We claim that at any scaling factor, the proposed algorithm yields low-complexity at a minimal loss of high image-quality with the ability to interpolate diverse image content.

Index Terms— polyphase, interpolation, classification, IFIR

1. INTRODUCTION

Image interpolation relates to methods of placing new pixel values into a regularly sampled grid given a discrete subset of points taken from a smaller grid. The smaller grid from which the inputs are drawn compose the low-resolution image, and the larger grid of points on which the processed points are placed compose the high-resolution image. The quality of the resultant high-resolution image is based on clarity of content, edge sharpness, and texture detail. The metric of a good interpolating design concentrates on the trade off between performance in terms of image quality and computational complexity.

Design of such algorithms can often be divided into two groups: analytical and empirical. We are concerned with empirical methods, specifically the approaches in which the input space has been partitioned into several non-overlapping operating areas. Additionally, high-resolution images are built locally, so the input spaces we examine consist of image patches. There are several techniques fitting this description, the most effective being C.B. Atkins' classification-based filtering technique [1], which we analyze in subsequent sections.

Empirically designed local interpolation algorithms often use linear equations to describe the interpolation process. This suggests MMSE formulation where some form of $\mathbf{y} = A\mathbf{x}$ is involved, $\mathbf{x}$ and $\mathbf{y}$ denoting low-resolution and high-resolution image content, respectively. Such formulations involving the interpolation operator A are not informative in an analytical sense, and without reorganizing A into two-dimensions, there is little to conceptualize.

The purpose of this paper is to provide such analysis and from it, draw conclusions and infer interesting developments. We devote the remainder of our work to three issues.

1. We analyze current algorithms involving linear interpolating filters from a polyphase perspective. The representation of popular classification-based filtering algorithms provides interesting insight.
2. We observe that all interpolating filters have a common magnitude response, and we propose a single interpolating kernel that is zero-phase to accommodate for directional filtering.
3. Using well-known techniques, we extend the interpolating process to arbitrary interpolating factors, an idea that is made possible through previously introduced concepts and without which requires retraining, a costly process.

2. POLYPHASE REPRESENTATION OF CLASSIFICATION-BASED FILTERING

Many classification-based algorithms including [1] make extensive use of a modified

$$\mathbf{y} = A\mathbf{x} \tag{1}$$

formulation. In this section, we show that the structure of A makes it easy to derive the underlying two-dimensional filter's polyphase components.

2.1. Polyphase Representation

Let d be the length and width of a low-resolution observation patch, and u be the corresponding high-resolution dimension. The vector $\mathbf{x}$ is vectorized from a $d \times d$ image patch, and likewise, $\mathbf{y} \in \mathbb{R}^{u^2 \times 1}$. Then, for class j, $A^{(j)} \in \mathbb{R}^{d^2 \times u^2}$, where $j \in 1, 2, \cdots, C$. Interpolation by two (as observed by [1] and several other works) sets $d = 5$ and $u = 2$ because the dimension of the center pixel of $\mathbf{x}$ doubles in each direction. We can then build a two dimensional filter by observing the organization of $A^{(j)}$. Each row of matrix $A^{(j)}$, as seen by Fig. 1 interpolates a single element in $\mathbf{y}$. Hence, we can

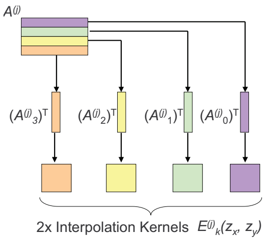


Fig. 1. The i^{th} row of matrix $A^{(j)}$ multiples $\mathbf{x}$ to create y_i. Reversing vectorization, we obtain the two-dimensional i^{th} polyphase component.

This work is supported by NVIDIA. It was performed while the first author was at UCSD.

find the high-resolution image by filtering the low-resolution image with information from each row in $A^{(j)}$, the polyphase components, and then interleaving the signal.

2.2. Filterbank Representation

Polyphase decomposition of signals [2] is the representation of signals as a regularly multiplexed, parallel set of subbands, where each subband is called a *polyphase component*. In terms of filterbanks, the aforementioned interpolation can be described as follows.

1. **Classify the local image content,**
2. **Filter the low-resolution image by all polyphase components E_k,**
3. **Upsample by u,**
4. **Shift each result by an appropriate offset,**
5. **Add the signals together**

Let $h[n]$ be a size N impulse response of a digital filter with transfer function $H(z) = \sum_{n=0}^{N-1} z^{-n} h[n]$, then the k^{th} polyphase component E_k, of $H(z)$ is defined as

$$E_k(z) = \sum_{n=0}^{\lfloor N/M \rfloor} z^{-n} h[nM+k] \qquad (2)$$

Expressing the collective group of E_k as a single filter requires a simple application of the second of two *noble identities* [3], where

$$\begin{aligned} \textit{First Noble Identity}: \quad & G(z)(\downarrow M) = (\downarrow M)G(z^M) \\ \textit{Second Noble Identity}: \quad & (\uparrow M)G(z) = G(z^M)(\uparrow M) \qquad (3) \end{aligned}$$

The noble identities allow us to push the down/upsampling block through the filter and reverse the order of blocks.

We can then express the transfer function as a sum of its polyphase components as

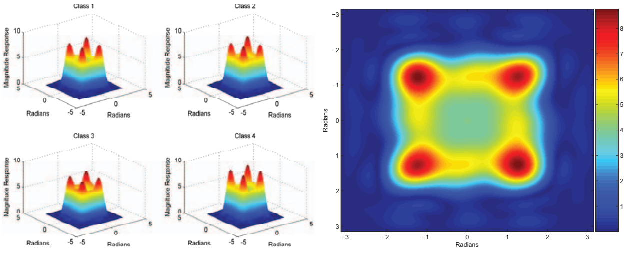
$$H(z) = \sum_{k=0}^{M} z^{-m} E_k(z^M), \qquad (4)$$

After equating the four independent filters with polyphase components, we now describe the interpolating process as the traditional and simple three step process that follows.

1. **Classify the image,**
2. **Upsample the image,**
3. **Filter using $H(z)$.**

3. ZERO-PHASE FILTER DESIGN

Our experimentation finds that the optimal number of classes[1] varies for each test image and is surprisingly low. Using filter responses from Sec. 2, the findings over a broad sweep of total classes and a fairly large training set are quite stunning. The magnitude responses for all filter responses take on exactly the same form. Fig. 2 is just a single example[2], and the trend is consistent for different C and across all j. We have some thoughts as to what is at work, but the general idea is that image interpolation is somehow governed by a *single* phenomena.



(a) Meshgrid of Padded Responses (b) Top-down View of Typical Frequency Response

Fig. 2. Magnitude Responses for Four Classes, $u = 2$.

The remainder of this section describes a single filter with the common magnitude response seen in Fig. 2. To accommodate different orientations, such an encompassing filter would almost certainly be zero-phase, and of course, given the symmetry of Fig. 2, real. We also impose certain wavelet conditions for equal energy among subbands. The assertion is that we can maintain near the performance of [1] at a fraction of the computational cost.

Our association of each row of A as the polyphase components, $E_k(z_x, z_y)$, of $H(z_x, z_y)$ defines four linear systems, which according to symmetry constraints are algebraically dependent. Let A_k, again, be the k^{th} row of A defined in (5).

$$A_k = \text{vectorize}\left\{E_k(z_x, z_y)\right\}, \qquad m = 0, 1, 2, 3. \qquad (5)$$

Letting $d = 5$ and $u = 2$, $H(z_x, z_y)$ is a Type II filter, so the impulse response of the k^{th} polyphase component is *not* symmetric. Instead, the k^{th} polyphase component must satisfy:

$$\begin{aligned} e_0[n_x, n_y] &= e_3[N - n_x - 1, N - n_y - 1] \\ e_1[n_x, n_y] &= e_2[N - n_x - 1, N - n_y - 1]. \qquad (6) \end{aligned}$$

If we define the "vectorize" operation as the concatenation of columns in an image patch into a one long column (MATLAB's `im2col` does the same thing), then

$$\begin{aligned} A_0[n] &= A_3[d^2 - n - 1] \\ A_1[n] &= A_2[d^2 - n - 1]. \qquad (7) \end{aligned}$$

Equation (7) gives us an advantage since we only need to find half as many parameters. Define $\hat{A}$ with only two rows instead of four such that

$$\hat{A} = \begin{bmatrix} -A_0- \\ -A_1- \end{bmatrix} \qquad (8)$$

To take advantage of symmetry, define a "flip" operator that reverses the order of elements inside a vector. Then, the training pairs associated with the new $\hat{A}$ are derived from the original training pair, $(\mathbf{x}_i, \mathbf{y}_i)$, to give

$$\begin{aligned} \hat{\mathbf{x}}_i &= \begin{bmatrix} \mathbf{x}_i & \mathbf{x}_i + \text{flip}\{\mathbf{x}_i\} \end{bmatrix} \\ \hat{\mathbf{y}}_i &= \begin{bmatrix} \hat{y}_i[0] \\ \hat{y}_i[1] \end{bmatrix} = \begin{bmatrix} y_i[0] & y_i[0] + y_i[3] \\ y_i[1] & y_i[1] + y_i[2] \end{bmatrix} . \qquad (9) \end{aligned}$$

Here, $\hat{\mathbf{x}}_i$ has an extra column supplementing $\mathbf{x}_i$ that enforces symmetry, and $\hat{\mathbf{y}}_i$ is half the height of $\mathbf{y}_i$, also with an additional column. We wish to solve the new system of equations:

$$\hat{\mathbf{y}} = \hat{A}\hat{\mathbf{x}} \qquad (10)$$

[1] We divide classes by the first principle component. PSNR is the metric of "optimal".

[2] We have zero-padded the impulse response, $h^{(j)}[n_x, n_y]$, out to 256^2 total samples (where before it was 10^2) to generate a smooth DFT curve. It is unnecessary to do so.

while simultaneously enforcing the wavelet constraint described by

$$\begin{aligned} H(\pi,\pi) &= \sum_{n_x,n_y} h[n_x,n_y]e^{-j\pi(n_x+n_y)} = 0 \\ &\leftrightarrow \sum_i A_{m,i} = 1. \end{aligned} \tag{11}$$

The solution considering (11) is similar to the original MMSE equations, with an additional term. Let us denote $\hat{Y}$ as a $u \times T$ matrix and $\hat{X}$ as a $d^2 \times T$ (where T is the number of training pairs) of all normalized training points. The optimization problem is written in (12)

$$\begin{aligned} \min_{\hat{A}} \quad & \left\| \mathbf{\hat{y}} - \hat{A}\mathbf{\hat{x}} \right\|^2 \\ \text{s.t.} \quad & \hat{A}\mathbf{1} = \mathbf{1}, \end{aligned} \tag{12}$$

where $\mathbf{1}$ denotes a vector of ones of appropriate length. We have written the constraint described by (11) as $\hat{A}\mathbf{1} = \mathbf{1}$ to regularize the energy in the polyphase subbands.

The problem is convex, and we can determine the closed form by writing the Lagrangian and taking its derivative. We have solved for the d^2-dimensional Lagrange optimization vector shown in (13).

$$\boldsymbol{\lambda} = \frac{R_{xy}R_{xx}^{-1}\mathbf{1} - \mathbf{1}}{\mathbf{1}^T R_{xx}^{-1}\mathbf{1}}, \tag{13}$$

where $R_{xx} = \hat{X}\hat{X}^T$ and $R_{xy} = \hat{X}\hat{Y}^T$. The final solution is given as:

$$\hat{A} = R_{xy}R_{xx}^{-1} - \mathbf{1}^T R_{xx}^{-1} diag(\boldsymbol{\lambda}), \tag{14}$$

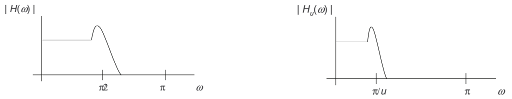
where $diag(\boldsymbol{\lambda})$ is a square matrix with the entries of λ on the diagonal and zeros elsewhere.

4. EXTENSIONS TO RATIONAL SCALING FACTORS

Standalone learning algorithms by design can be rather inflexible. Because the purpose is to learn the natural relationship between input/output training pairs of fixed pixel dimensions, inferences can be made only of that particular relationship. Unforeseen adjustments such as added noise models require alterations to the training set, without which place generalization beyond the intended scope of the original algorithm. For interpolation, we expect data from a $d \times d$ image patch to resolve a single pixel to $u \times u$ pixels. Any change to the problem statement, i.e. resolving a single pixel to $v \times v$ pixels, where $u \neq v$, may exceed the intended purpose of the algorithm. Consequently, to change u, the upsampling factor, algorithms such as [1] must obtain an entirely new training set. With the new framework, doing so is unnecessary, and this section addresses issues related to rational scaling factors.

The shape of Fig. 2 simplifies our task considerably. We are not concerned with the phase response, nor the bands in Fig. 2 that are close to zero, i.e. the higher frequencies. With a scaling factor of u, the interval of consequence ranges from $\omega = [-\omega_u, \omega_u]$, where we have defined $\omega_u = \frac{\pi}{u} + \delta_u$. Therefore, resolving an image to resolution v given only the filter $H(\omega)$ (that interpolates by u), which cuts off at ω_u, the task is to find an $H_v(\omega)$ from $H_u(\omega)^3$, where v is the desired interpolating factor.

Video postprocessing techniques almost universally advise against zero padding a filter response in the frequency domain.



(a) Interpolation by $2\times$ (Trained Response, where $u = 2$)

(b) Interpolation by $v\times$

Fig. 3. Magnitude Responses for Factors of $u = 2$ and v

On the whole, we concur that induced ringing artifacts by a synthetically zeroed response may be noticeable, albeit our experimentation suggests that the effect is not severe. A more reserved suggestion is to interpolate the interpolating filter itself and obtaining an IFIR filter [4]. Because we are most interested in the aliased copy on the interval $[-(\pi/u + \delta), \pi/u + \delta]$, we regularize our filter with another filter that preserves the non-zero portion in that interval while zeroing out higher, cutoff frequencies. Many references suggest equiripple filters from the Parks-McClellan algorithm because of its sharp transition at the cutoff frequency, a well-sustained passband, and good point-wise approximation of zero in the stopband. Such filters, however, require additional wavelet constraint enforcements. We have chosen the $G(z)$ to be the maximally flat (Daubechies) filter because of its many zeros at π.

Rational scaling implies that we wish to interpolate by $v = H/L$, where H and L are positive integers. Conventional approaches, most notably those concentrating on multiframe superresolution [5], use concepts of non-uniform sampling and exclusively place the responsibility of generating values onto the high-resolution grid in the co-domain. The analytical nature of such models requires assumptions on their correlation matrices as they must be assumed[4] Our correlation matrices are generated empirically, but we adopt the idea of filtering on the desired high-resolution grid. Interpolation by $v = \frac{H}{L}$ is as follows.

1. **Generate IFIR filter** $H_H(z) = G(z)H(z^{\frac{H}{u}})$
2. **Downsample for the interpolating filter** $H_v(z) = H_H(z^{\frac{1}{L}})$
3. **Upsample low-resolution image by** H
4. **Apply** $H_v(z)$ **to every** L^{th} **pixel and downsample by** L**.**

5. RESULTS

The training set, Ω, consists of preprocessed (mean-shifted, variance normalized, and vectorized) low-resolution/high-resolution pairs $(\mathbf{x}_i, \mathbf{y}_i)$. When measuring the Peak Signal to Noise Ratios (PSNR) values, we downsample by a fixed factor with MATLAB's `imresize` command (MATLAB applies an anti-aliasing filter that defaults to 11 × 11 taps), upsample by the same factor, apply the proposed filter $H(z_x, z_y)$, and measure the differences between the original image and the interpolated one. We numerically and visually compare to various methods that have been cited throughout the body of this work, and we benchmark using MATLAB's `profile` function.

We have compared our zero-phase filter interpolation against classification-based in the Mixture of Experts (MoE) and Tree-based (TB) framework, both of which come from [1] along with edge directed interpolation (NEDI) algorithms. Average function call times and PSNR results for various interpolation methods are shown in Table 1. Visual results against state of the art, shown in Fig. 4, yield results that are comparable if not equal in quality. Additionally, like

[3] Here, $v \geq u$. When $v < u$, we must deal with aliasing conditions.

[4] This is often done with common PSF's and spectral density estimates.

the classification-based algorithms, the zero-phase filter surpasses conventional approaches, including edge-directed interpolation.

Table 1. 2× Interpolation Performance and Benchmarks

Interpolation Time (seconds)					
Sequence	MoE	TB	NEDI [6]	Halfband	0-phase
Barbara	846.48	45.09	72.96	0.087	0.042
Bus	182.54	21.93	15.61	0.022	0.0101
Paris	182.61	21.62	15.41	0.022	0.0095
City	182.99	21.78	15.67	0.022	0.0096
Peak Signal to Noise Ratio (dB)					
Barbara	28.30	26.56	28.64	26.44	28.32
Bus	23.78	24.26	24.80	23.00	24.72
Paris	21.40	21.62	22.06	20.22	21.54
City	27.70	27.40	27.46	25.04	26.90

For purposes of brevity in this paper, images and examples have been shrunk. Non-integer scaling factors and additional images along with detailed analysis can be found at the author's website:

`http://videoprocessing.ucsd.edu/~karl/0phasefilt`

6. CONCLUSIONS AND FUTURE WORK

We have shown that interpolation algorithms with MMSE formulation can be expressed in terms of polyphase components. Additionally, interpolation filters for natural images of any variety all have an underlying magnitude response. Creating a filter with such a magnitude response and zero-phase yields excellent results.

In collaborating with other areas of research such as computer vision and biomedical imaging, the nature of local image patches appears to have some universal properties including observations examined in this paper. Depending on the application, further generalization of observed properties may lead to a better understanding of images as a whole, including immediate extensions image deblurring and other postprocessing techniques.

7. REFERENCES

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

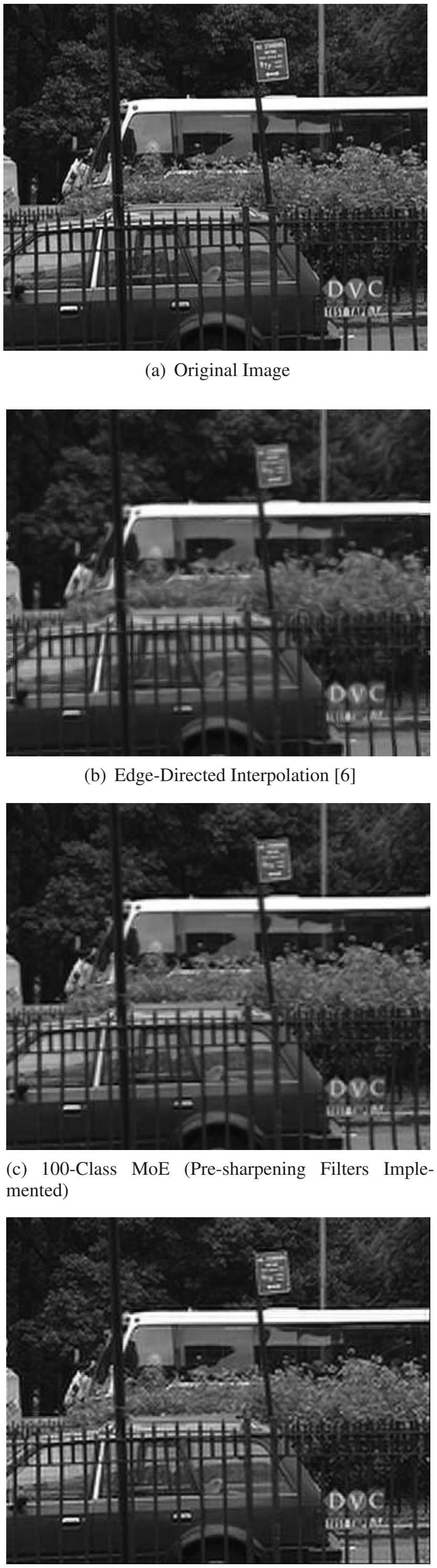
[2] P. P. Vaidyanathan, "Multirate digital filters, filter banks, polyphase networks, and applications: A tutorial," *Proc. of IEEE*, vol. 78, no. 1, pp. 56–93, Jan. 1990.

[3] Gilbert Strang and Truong Nguyen, *Wavelets and Filterbanks*, Wellesley-Cambridge Press, 1996.

[4] Y. Neuvo, D. Cheng-Yu, and S. Mitra, "Interpolated finite impulse response filters," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, no. 3, June 1984.

[5] Sean Borman and Robert L. Stevenson, "Simultaneous multi-frame MAP super-resolution video enhancement using spatio-temporal priors," in *Proceedings of the IEEE International Conference on Image Processing*, Kobe, Japan, Oct. 1999, vol. 3, pp. 469–473.

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



(a) Original Image

(b) Edge-Directed Interpolation [6]

(c) 100-Class MoE (Pre-sharpening Filters Implemented)

(d) Proposed Zero-phase Filtering Approach

Fig. 4. Comparisons to State of the Art