# A Conservative Sharpening Filter for Interpolated Images

Leonardo R. Emmendorfer
Center for Computational Science, Federal University of Rio Grande
96201-900 Rio Grande RS, Brazil
Email: leonardo.emmendorfer@gmail.com

***Abstract*—Interpolation methods are computationally efficient and rather effective for magnification from low-resolution images in order to generate higher resolution images. However, blur and other undesirable visual artifacts are often produced during the process. This work evaluates a sharpening filter that enhances the quality of interpolated images by preserving information from the original low-resolution images. The novel filter was shown to achieve statistically significant increments in both the PSNR and SSIM of magnified RGB images using two image sets from the literature.**

## I. Introduction

Image magnification (or superresolution) is a process where a higher resolution (HR) image is computed from one or more low resolution (LR) input images without introducing blur [1]. It can be considered as a difficult, ill-posed problem [2]. In general, no hints on the real high-resolution ground-truth signal are available.

Existing solutions to image magnification can roughly be classified into three classes: (1) reconstruction methods; (2) learning-based methods; and (3) interpolation methods [1]. The latter category is the most frequently adopted for single image superresolution [3]. The reconstruction task is based on prior knowledge about the model that maps the high-resolution image to the low-resolution one, while learning-based approached adopt machine learning methods over a set of examples of low resolution and their ideal high-resolution patches [1].

Several interpolation methods with improved features have been proposed in the literature. In [4] the interpolation first occurs respecting the direction where the second-order derivative of the image is lower. Then interpolated values are modified using an iterative refinement maximizing second-order derivative values and smoothing isolevel curves. In general, directional image interpolation takes advantage of the geometric regularity of image structures by performing the interpolation in a chosen direction along which the image is locally regular [1]. The adoption of edge detectors and edge directed models in interpolation algorithms is a relevant topic of research [5]. Several types of models for the gradient profile structure can be adopted to locate the edge points [6]. In [7] the adaptively sharpening of the gradient field is proposed to be based only on small neighborhoods, which leads to a computationally efficient approach.

Sparse representations [8], on the other hand, are based on modeling the low-resolution image as a down-sampled version of its high-resolution counterpart after blurring [2]. The methods are shown to effectively reconstruct edge structures and suppress artifacts from interpolated images. Improved image interpolation methods also include auto-regression models [9], multi-surface fitting models [10], among others [6], [11], [3].

The main disadvantage of more sophisticated interpolation approaches is the need for additional stages of information processing [12]. Although many alternatives have been proposed, simpler interpolation algorithms still play an important role in image magnification [3] at least in the initial stages of the process [11]. Using different neighborhoods and kernels, classic methods widely applied in image viewers and image processing tools have been obtained. [13].

However, simpler interpolation algorithms like Nearest Neighbor (NN) and Bicubic tend to produce visual artifacts such as ringing, aliasing, and blurring, among others, particularly in edge regions [14], [1], [12]. Among the visual artifacts resulting from this class of algorithms, blurring is probably the most prevailing. For instance, a given polynomial algorithm that interpolates a pixel next to the edge between two objects of different colors will use pixels from both sides of the edge in the original image, generating intermediate colors that are not found in the original image, and producing a less definite edge [15].

This motivates the adoption of sharpening filters after the interpolation, which would alleviate or eliminate blurring and other artifacts that are generated by the interpolation process and therefore enhance the quality of the images obtained. Many filters are available in the literature, which can be applied at this post-interpolation step. This work evaluates the effect of the application of sharpening filters on RGB images resulting from magnification by the adoption of simple interpolation methods. In order to preserve the advantage of computational efficiency, only simple and fast sharpening filters were adopted. A novel sharpening filter is evaluated which was designed specifically for enhancing the quality of interpolated images by replacing the computed intensity of interpolated pixels in the HR resulting image by the intensity of pixels from the LR original image. The idea was already explored in [15], where a much smaller set with only grayscale images was adopted for the empirical evaluation.

This paper is organized as follows: Section 2 presents sharpening filters. Section 3 describes and briefly analyses the proposed filter. Experimental setup and results are shown in Section 4 and conclusions are given in the last section.

## II. Sharpening filters

Sharpening filters are widely adopted in image processing. A sharpening filter aims to provide an image with better-defined borders when compared to its original. Methods such as the linear unsharp masking (USM) and Laplacian filter are among the most widely used for image sharpening. Both are defined as second-order derivatives and realized as local spatial filters. Although these image sharpening filters are simple and work well in many applications, they have drawbacks since both might be very sensitive to noise, resulting in unpleasant granularity and they often enhance too much high contrast areas, resulting in unpleasant overshoot artifacts [16].

Another simple but relatively effective approach for sharpening is the adoption of order statistics. The alpha-trimmed mean (ATM) filter [17], for instance, operates by removing the statistically most extreme points and averaging over the remaining points. Lower-upper-middle (LUM) sharpening filter [18] is also rank-order-based. It adopts the comparison of lower and upper-order statistics to the sample in the center of a filter window at coordinates $(i, j)$. The LUM sharpening filter is defined as

$$\hat{I}_{i,j} = \begin{cases} r_{(l)}, & \text{if } r_{(l)} < I_{i,j} \leq t_l \\ r_{(N-l+1)}, & \text{if } t_l < I_{i,j} < r_{(N-l+1)} \\ x_{i,j}, & \text{otherwise.} \end{cases}$$

where $r_{(1)} \leq r_{(2)} \leq \cdots \leq r_{(N)}$ is a rank-ordered set from the intensities of the $N = (2m+1)(2m+1)$ pixels in the $(2m+1) \times (2m+1)$ squared filter window centered at $(i, j)$. $t_l = (r_{(l)} + r_{(N-l+1)})/2$ is the midpoint between the statistics $r_{(l)}$ and $r_{(N-l+1)}$. $l$ is the parameter of the sharpening filter, with $1 \leq l \leq (N+1)/2$. One can notice that $l$ controls the amount of sharpening desired, where $l$ at its highest value $(N+1)/2$ leads to a simple identity filter and $l = 1$ corresponds to the maximum sharpening since $I_{(i,j)}$ is always shifted to one of the extreme statistics $r_{(1)}$ or $r_{(N)}$. Here in this paper the parameter l is normalized to $\bar{l} = \frac{l}{(N+1)/2}$, therefore $\bar{l} = 1$ corresponds to a identity filter independently of the window size.

In [19] an adaptive method for sharpening is proposed, where edge areas are classified into different types with varying slopes and multiple edge enhancement filters applied to the respective edge types. Among the drawbacks of the method are the unclear classification criterion, the blurring influence around edges, and the computational performance [16]. Several other approaches should be mentioned. Contrast limited adaptive histogram equalization (CLAHE) [20] is a variant of adaptive histogram equalization in which the contrast amplification is limited to reduce the problem of noise amplification. Weighted medians were already shown as relevant approaches [21], [22], [23]. The adoption of wavelets in the context of image sharpening was proposed by [24], among others. More sophisticated algorithms for image sharpening are also available. The edge-directed unsharp masking sharpening method (EDUMS) [25] is guided by edge-directed information and was able to reach positive empirical results.

Image sharpening is an active area of research since many issues remain to be approached. Concerns arise from the practitioners since this type of filter potentially increases overshoot artifacts that adversely affect the results, which is specially critical for some applications [26].

## III. Proposed Sharpening Filter

Each channel of an image is represented by a 2D matrix $I$, where $I_{i,j}$ is the intensity of a pixel in coordinates $(i, j)$. Let $R$ be the resulting higher-resolution image obtained by interpolation from a lower-resolution image $S$ by a factor $f \times f$.

A magnification from $S$ is the computation of each pixel $R_{i,j} \in R$ as a function of some pixels in $S$, which correspond to the neighborhood adopted by the specific interpolation algorithm. Let $E$ be the Nearest Neighbor interpolation from $S$ by the same factor $f \times f$. For each $R_{i,j}$, the value of the closest original pixel from $S$, in terms of the Euclidean distance computed from coordinates, can be found in $E$ at $E_{i,j}$. Consider an interpolation algorithm where, at regularly distributed coordinates $(p, q)$ the pixels in $R$ are just copies of the original pixels from $S$ at corresponding positions ($R_{p,q} = E_{p,q}$). The remaining $R_{i,j}$s are computed as a function of some neighborhood of original pixels from $S$. Depending on how this interpolation is performed, blur will be obtained from this computation since the interpolation function is often smooth.

The proposed filter stems from the idea that computed intensities for interpolated pixels should be replaced by intensity values from the original image $S$, in order to prevent blur. For a given interpolated pixel $R_{i,j}$ the closest original pixels, in terms of the Euclidean distance computed from coordinates, can be found in image $E$ in a filter window centered at $(i, j)$. Let $G(i, j)$ be the $(d+1) \times (d+1)$ squared filter window for image $E$ centered at $(i, j)$. A pixel is chosen from the options available in $G(i, j)$ which value better represents the computed pixel $R_{i,j}$; this will be entitled as the "best neighbor" $B_{i,j}$ of $R_{i,j}$. Let us define $B_{i,j}$ as the pixel in the filter window $G(i, j)$ which is the most similar to the value of $R_{i,j}$, in terms of color similarity. Therefore, the expression for $B_{i,j}$ is:

$$B_{i,j} = g_{best(i,j)} \in G(i, j) \tag{1}$$

where $best(i, j) = argmin_h(|R_{i,j} - g_h|)$, $g_h \in G(i, j)$ indicates the value in $G(i, j)$ which is the most similar to the intensity of $R_{i,j}$ for single-channel images. For RGB images, a distance metric beteen two colors must be assumed. In our experiments the Euclidean distance, in the RGB color space, between any two colors $(red_{c_1}, green_{c_1}, blue_{c_1})$ and $(red_{c_2}, green_{c_2}, blue_{c_2})$ was adopted for all RGB images considered.

Since every $B_{i,j}$ is selected from pixels in $E$, therefore the resulting image $B$ is composed solely by pixels with intensity values that are already present in the input image $S$. This same

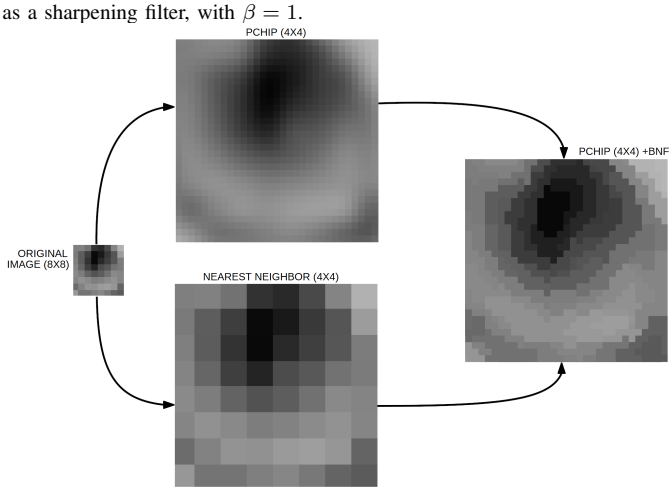
property is achieved by the nearest neighbor interpolation. Additionally, if we set the size of the filter window to 1, the best neighbor $B_{i,j}$ of every pixel $R_{i,j}$ is $E_{i,j}$, then $B$ is equivalent to the image $E$ resulting from nearest-neighbor interpolation. By increasing the window size, diverse behavior might arise.

A second step can be adopted, optionally. The final resulting image $\hat{I}$ is composed by the linear combination between filtered image $B$ and the interpolated image $R$ as given by:

$$\hat{I} = \beta B + (1 - \beta)R \tag{2}$$

where $\beta$ is a parameter of the filter. Since the method is based on the selection of the best neighbor for each pixel, it will be hereafter called Best Neighbor Filter (BNF). Fig. 1 illustrates the computation of $B$ from an $8 \times 8$ image with a filter window of size $5 \times 5$. The computational complexity of sorting

Fig. 1. 4×4 magnification from an illustrative 8×8 image obtained using Piecewise Cubic Hermite Interpolating Polynomial (PCHIP) method and BNF as a sharpening filter, with $\beta = 1$.




the available options in $G(i, j)$ is $O(d \log(d))$ therefore the computational complexity of BNF is $O(mnd \log(d))$, where $m$ and $n$ are the dimensions of the image. In order to guarantee the computational efficiency of the method, the size of the filter window should be kept small. Too small window sizes, however, might not guarantee a diversity of intensity values in $G(i, j)$. For instance, with a window size smaller than $f$ there might be a single intensity value available in the respective neighborhood $G(i, j)$, which is not useful for the purpose of this filter.

### *A. Analysis of the Proposed Filter*

Despite a broader theoretical analysis of the proposed filter is out of the scope of this paper, this subsection provides an illustration of the behavior of the filter under some specific situations. Consider a one-dimensional image $U = \{(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)\}$ where $x_1 < x_2 < \cdots < x_n$ and each $y_i \in [0, 1]$ is the pixel intensity corresponding to a coordinate $x_i \in \mathbb{R}$. Consider also that $U$ was sampled from a ground-truth unit step signal $s : \mathbb{R} \to \{0, 1\}$ given by:

$$s(x, t) = \begin{cases} 1 \text{ if } x \geq t \\ 0 \text{ otherwise} \end{cases} \tag{3}$$

where $t \in [0, 1]$ is a real-valued thereshold.

Interpolation between any two coordinates $x_k$ and $x_{k+1}$ is trivial under those assumptions except when $x_k \leq t \leq x_{k+1}$. In this theoretical analisys a simple pairwise linear interpolation between all subsequent pairs of pixels $x_k$,$x_{k+1}$ is applied to $U$ prior to the adoption of BNF, resulting an interpolated image $P$.

When applied to a pairwise linear interpolation $P$ computed from a one-dimensional input image $U$ which was sampled from a ground-truth unit step signal $s$, the BNF filter is expected to better resemble the original signal $s$ from which $U$ was sampled when compared to the resemblance obtained from the unfiltered interpolation $P$. Next, we sketch a theoretical verification for the statement above.

Without loss of generality, consider the relation between the parameter $t$ in (3) and the coordinates of two subsequent pixels $x_0$ and $x_1$ in $U$ is given by $x_0 \leq t \leq x_1$ where $x_0 = 0$ and $x_1 = 1$. Therefore, assuming an efficient sampling, the intensity of the input image $U$ at $x_0$ is $s(x_0, t) = 0$ and the intensity at $x_1$ is $s(x_1, t) = 1$ for any $0 \leq t \leq 1$. Under those circumstances, intensities of the interpolated pixels in the image $P$ at any coordinate $x$ are generated simply as $p(x) = x, x \in [0, 1]$. The expected squared error $E([p(x) - s(x, t)]^2)$ of any interpolated pixel in $[0, 1]$ is computed by integrating the squared error over all equally likely possible values for $t$ and also simultaneously over all equally likely possible values for $x$, which leads to:

$$E([p(x) - s(x, t)]^2) = \frac{\int_0^1 \int_0^1 [p(x) - s(x, t)]^2 dt dx}{\int_0^1 \int_0^1 dt dx} \tag{4}$$

$$= \frac{\int_0^1 \int_0^1 [x - s(x, t)]^2 dt dx}{\int_0^1 \int_0^1 dt dx} = \frac{1}{6} \tag{5}$$

The application of BNF with $\beta = 1$ to the linearly interpolated image $P$ results in an image where the intensity of each pixel follows a unit step signal as:

$$b(x) = \begin{cases} 1 \text{ if } x \geq \frac{1}{2} \\ 0 \text{ otherwise.} \end{cases}$$

Therefore the computation of the expected squared error of the filtered image $E([b(x) - s(x, t)]^2)$ can be more easily obtained as the average $\{E^{t<\frac{1}{2}}([b(x) - s(x, t)]^2) + E^{t\geq\frac{1}{2}}([b(x) - s(x, t)]^2)\}/2$ between two equally likely situations which correspond to $t < \frac{1}{2}$ and $t \geq \frac{1}{2}$ respectively. Since both:

$$E^{t<\frac{1}{2}}([b(x) - s(x, t)]^2) = \frac{\int_0^1 \int_0^1 [b(x) - s(x, t)]^2 dt dx}{\int_0^1 \int_0^1 dt dx} \tag{6}$$

$$= \frac{\int_0^1 \int_t^{\frac{1}{2}} [b(x) - s(x, t)]^2 dx dt}{\int_0^1 \int_0^1 dx dt} \tag{7}$$

$$= 0 \tag{8}$$

and

$$E^{t\geq\frac{1}{2}}([b(x)-s(x,t)]^2) = \frac{\int_0^1\int_0^1[b(x)-s(x,t)]^2dtdx}{\int_0^1\int_0^1 dtdx} \quad (9)$$

$$= \frac{\int_0^1\int_{\frac{1}{2}}^t[b(x)-s(x,t)]^2dxdt}{\int_0^1\int_0^1 dxdt} \quad (10)$$

$$= 0 \quad (11)$$

then $E([b(x)-s(x,t)]^2) = 0 < E([p(x)-s(x,t)]^2)$, which means that BNF is expected to better resemble the original signal $s$ under the depicted conditions.

The adoption of other simple interpolation algorithms instead of the linear alternative adopted might lead to similar conclusions, but this aspect should be explored further. The analysis perfored here covers a very specific situation related to a simple type of image. Despite that, the general formulation roughly corresponds to the generation of blur from simple interpolation methods in places of an image where a sharper border was expected. This suggests a potencial improvement of image quality after the adoption of BNF, which is to be empirically evaluated in the following Section.

## IV. Experimental Results

The experimental design stems from the hypothesis that BNF could improve the quality of images resulting from interpolation. In order to determine whether that hypothesis can be confirmed we designed experiments where evaluation is performed over $f \times f$ magnification of LR images which were generated previously from downscaling the original HR source images by the same $f \times f$ factor. The quality of both the filtered and the unfiltered images are measured as the similarity between each reconstructed image and the original HR source, so the effect of applying the sharpening filter can be tested. The computation of the similarity between a reconstructed image and the corresponding HR source image is performed using the peak signal-to-noise ratio (PSNR) and the structural similarity (SSIM) [27] independently. Four interpolation methods were selected for this evaluation. Piecewise Cubic Hermite Interpolating Polynomial (PCHIP) is a widely adopted computationally effective algorithm which represents a simple polynomial interpolation approach. New edge-directed interpolation (NEDI) [28] represents the class of edge-based interpolation algorithms. Interpolated pixels are obtained as weighted averages of the neighbors, where weights depend on the edge direction. The iNEDI is a modification from NEDI which attempts to provide sharper images although with a high computational cost [29]. Finally, the Iterative Curvature Based Interpolation algorithm (ICBI) [13] is characterized by a two-stage process where interpolation first occurs respecting the direction where the second-order derivative of the image is lower. Then interpolated values are modified using an iterative refinement maximizing second-order derivative values and smoothing isolevel curves [4].

Two sets of evaluations are performed. In the first set PCHIP algorithm is adopted as the interpolation method and two image sets are adopted for the evaluation of BNF using four magnification factors, while the second set of evaluations focuses on one image set with varying filters and interpolation algorithms applied. The *BSD500* image set[1] is composed by a total of 500 mostly outdoor RGB images with varying sizes. The image set was previously adopted for the evaluation of segmentation algorithms in [30]. A smaller image set denoted here as *Morguefile25* was also adopted which includes 25 RGB $512 \times 512$ versions of images from [31][2]. The images in *Morguefile25* were selected in [4] where the same 25 images were adopted for the evaluation of magnification algorithms. Not only the source images but also the results from diverse interpolation algorithms adopted in [4] were made availabe[3].

Fig. 2. Examples of images from image sets *BSD500* (a) and *Morguefile25* (b).

(a) (b)

Figs.3a and 3b illustrate the sharpening effect resulting from the BNF filter using the default parameter $\beta = 1$ when applied to small regions of images interpolated using PCHIP. One image is selected from each set. Corresponding original images are shown in Fig. 2. The locations corresponding to the small regions illustrated in Fig. 3 are shown as delimited with red squares in Fig. 2. For all cases, one can notice that BNF is able to provide sharper edges for objects which suffered from blur after interpolation. Staircase-like block artifacts generated by PCHIP are alleviated after filtered with BNF. As the magnification factor increases, however, those types of artifacts from PCHIP become even more evident as a result of the sharpening effect of the filter.

### A. Empirical evaluation using PCHIP as the interpolation method

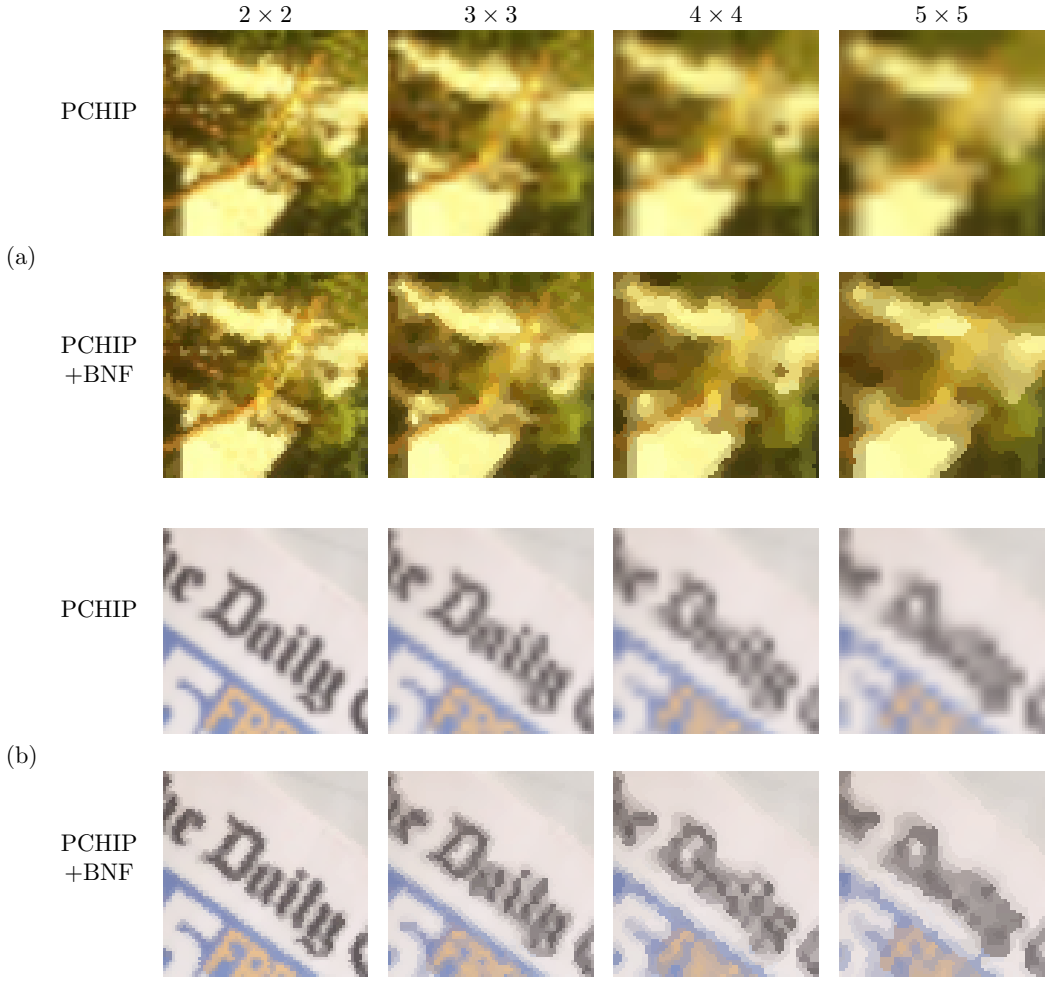
The evaluation is performed as follows. A parameter $f$ is set, which determines the magnification and corresponding downscaling factor, both denoted as $f \times f$. Source images from each set are downscaled by the given factor, resulting in a set with $k$ LR images $\mathbf{S} = \{S_1, S_2, \cdots, S_k\}$ which are reconstructed by PCHIP. This results in a set of interpolated images $\mathbf{R} = \{R_1, R_1, \cdots, R_k\}$. BNF filter is then applied

[1]https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/ resources.html

[2]All images are subject to the license agreement available at the web page http://morguefile.com/archive/terms.php.

[3]http://www.andreagiachetti.it/icbi/

Fig. 3. The sharpening effect resulting from the application of the BNF filter with $\beta = 1$ as illustrated by small regions of images interpolated using PCHIP, which was applied to downscaled versions of images from the *BSD500* (a) and *Morguefile25* (b) sets. Four magnification factors are considered: $2 \times 2$, $3 \times 3$, $4 \times 4$ and $5 \times 5$. Original images are shown in Fig. 2.




to all images in $R$, by searching for the optimal parameter value for each image which maximizes the similarity measure between the source image and the filtered version of the interpolated image from $\mathbf{R}$. The search is performed exhaustively over a discretized parameter space $\beta \in [0, 1]$ with 2 digits of precision, independently for PSNR and SSIM, for each interpolated image in $\mathbf{R}$. For each similarity measure, the set of quality evaluations of the $k$ filtered images is compared to the respective set of quality evaluations for the unfiltered images reconstructed using PCHIP only. Wilcoxon signed-rank tests are performed for evaluating the statistical significance of an eventual difference, at the $\alpha = 0.05$ significance level.

The window size for BNF under this evaluation was set as a function of the magnification factor $f$ as $(f + 1) \times (f + 1)$ which leads to a $3 \times 3$ window for $2 \times 2$ magnification and $5 \times 5$ window for $4 \times 4$ magnification. The criterion adopted here is the minimization of the window size since even smaller windows would eventually prevent BNF from having a viable supply of diverse values from the neighbors of an interpolated pixel. For odd values of $f$, however, the window size was set to $(f+2) \times (f+2)$ which preserves the center of the window exactly in the coordinates of the corresponding pixel in $(i, j)$ since $f + 2$ is odd. This leads to window sizes of $5 \times 5$ and $7 \times 7$ respectively for $3 \times 3$ and $5 \times 5$ magnifications. A $f \times f$ window would not allow for a proper variety for the values in the neighborhood of a pixel and BNF would behave trivially like a Nearest Neighbor approach.

Table I shows the results for average optimal PSNR (dB) and average optimal SSIM $\pm$ the respective standard deviations $\sigma$ obtained from the $2 \times 2$, $3 \times 3$, $4 \times 4$ and $5 \times 5$ reconstruction of the images from each image set. The BNF filter was able to achieve statistically significant results for the increment in both PSNR and SSIM for all image sets and also all magnification factors considered at the significance level adopted. P-values from all the comparisons resulted in values below $0.001$ (not shown). The average optimal values obtained for the BNF parameter $\beta^\star$ for each image set, considering each similarity measure, are also shown in Table I. The

TABLE I

AVERAGE OPTIMAL PSNR AND SSIM $\pm$ RESPECTIVE STANDARD DEVIATION $\sigma$ OBTAINED FROM THE MAGNIFICATION OF THE DOWNSCALED VERSIONS OF THE IMAGES FROM TWO IMAGE SETS CONSIDERED USING PCHIP AS THE INTERPOLATION METHOD AND BNF AS A FILTER. RESULTS FROM BNF (BOLD) ARE SIGNIFICANTLY SUPERIOR TO THE RESULTS CORRESPONDING TO THE RESPECTIVE UNFILTERED IMAGES AT THE SIGNIFICANCE LEVEL $\alpha = 0.05$, INDEPENDENTLY FROM THE SIMILARITY MEASURE ADOPTED.

| Image Set | Factor | Method | | | | Optimal Parameter Value | |
|---|---|---|---|---|---|---|---|
| | | PCHIP (no filter) | | PCHIP+BNF | | | |
| | | Avg.PSNR$\pm\sigma$ | Avg.SSIM$\pm\sigma$ | Avg.PSNR$\pm\sigma$ | Avg.SSIM$\pm\sigma$ | $\beta^\star$ (PSNR) Avg.$\pm\sigma$ | $\beta^\star$ (SSIM) Avg.$\pm\sigma$ |
| *BSD500* | $2\times2$ | 33.02$\pm$3.5393 | 0.7928$\pm$0.0955 | **33.10$\pm$3.5534** | **0.7942$\pm$0.0952** | 0.27$\pm$0.06 | 0.21$\pm$0.07 |
| | $3\times3$ | 30.93$\pm$3.4109 | 0.7077$\pm$0.1255 | **30.94$\pm$3.4122** | **0.7081$\pm$0.1253** | 0.15$\pm$0.07 | 0.12$\pm$0.08 |
| | $4\times4$ | 29.55$\pm$3.2594 | 0.6553$\pm$0.1388 | **29.57$\pm$3.2625** | **0.6556$\pm$0.1387** | 0.17$\pm$0.06 | 0.10$\pm$0.05 |
| | $5\times5$ | 29.00$\pm$3.2359 | 0.6326$\pm$0.1444 | **29.01$\pm$3.2372** | **0.6328$\pm$0.1443** | 0.15$\pm$0.07 | 0.08$\pm$0.05 |
| *Morguefile25* | $2\times2$ | 36.21$\pm$3.3361 | 0.8879$\pm$0.0694 | **36.33$\pm$3.3340** | **0.8888$\pm$0.0691** | 0.22$\pm$0.05 | 0.16$\pm$0.07 |
| | $3\times3$ | 32.44$\pm$2.9745 | 0.8132$\pm$0.1002 | **32.46$\pm$2.9726** | **0.8136$\pm$0.1000** | 0.13$\pm$0.06 | 0.09$\pm$0.07 |
| | $4\times4$ | 31.22$\pm$2.9380 | 0.7763$\pm$0.1128 | **31.26$\pm$2.9346** | **0.7768$\pm$0.1126** | 0.19$\pm$0.05 | 0.09$\pm$0.06 |
| | $5\times5$ | 29.77$\pm$2.8498 | 0.7446$\pm$0.1210 | **29.79$\pm$2.8516** | **0.7449$\pm$0.1209** | 0.15$\pm$0.05 | 0.06$\pm$0.04 |

magnification factor and also the similarity measure adopted both seem to affect $\beta^\star$. In general, one might observe that $\beta^\star$ is higher for smaller magnification factors, notably for the $2\times2$ factor. Also, the variation on the window size resulting from the oddity of the factor $f$ might be effecting $\beta^\star$. A deeper study that covers those effects, however, is out of the scope of this paper.

### *B. Empirical evaluation with varying interpolation methods and filters*

The second set of evaluations covers a wider range of sharpening filters besides BNF, including also the Laplacian filter[4] and LUM which are also relatively simple filters. Similar to the experiments previously described, the filters are applied to images that were reconstructed by the adoption of interpolation algorithms.Three interpolation algorithms are considered in the evaluation: NEDI, iNEDI, and ICBI. A single image set *Morguefile25* was used in this setting. We adopted the reconstructions obtained from the interpolation algorithms mentioned which were made available for download by the authors of [4] along with the respective image set. The magnification factor for this set of evaluations is set to $4\times4$, with a $5\times5$ window size adopted for both BNF and LUM.

For each image and also considering each interpolation algorithm and filter a search for the optimal parameter value which maximizes the similarity measure between the source image and the corresponding interpolated and subsequently filtered image is performed, independently for PSNR and SSIM, as adopted in the previous set of evaluations. The same discretized parameter space $[0,1]$ with 2-digit precision is adopted for the single parameter of each filter considered.

Table II shows the results for average optimal PSNR (dB) $\pm$ standard deviation $\sigma$ and optimal SSIM $\pm$ standard deviation $\sigma$ obtained from the magnification of the downscaled versions of the original images using three interpolation methods and, for each interpolation method, three sharpening filters. LUM filter was able to achieve a higher average PSNR for the magnification of the images when compared to BNF for the images resulting from NEDI interpolation. For the other two interpolation algorithms, results from BNF are superior. BNF was able to achieve statistically significant improvements for all interpolation algorithms considered while LUM could not improve the PSNR for iNEDI algorithm significantly. The highest average PSNR considering all cases for this image set was obtained from BNF, which is $30.9960$ and corresponds to the filtered images which were interpolated using iNEDI. The highest increment in PSNR, when compared to the unfiltered version of the image set, is also obtained from BNF, but corresponds to the images reconstructed using ICBI. The filtered set of images corresponds to an average PSNR $1.30\%$ higher when compared to the unfiltered corresponding set. BNF was also the only filter that achieved statistically significant improvements in SSIM for the magnification of the RGB images from the image set considered. The weaker statistical results obtained from LUM results from the fact that the increment in SSIM obtained from LUM comes from a very small number of images since, for most of them, the optimal value of the parameter $\bar{l}$ is $1$ which makes LUM an identity filter. This makes the SSIM of most of the LUM filtered images equal to the corresponding values from the unfiltered counterparts. That differs from BNF which provides a more consistent increase in the evaluation metrics. BNF was able to improve both PSNR and SSIM for all images considered in this evaluation, independently from the interpolation algorithm adopted. However, the highest average SSIM considering all cases for this image set was obtained from LUM, which is $0.7654$ and corresponds to the filtered images which were interpolated using ICBI. This also corresponds to the highest increment in SSIM when compared to the unfiltered version of the image set.

To provide better guidance for understanding the blur reduction effect obtained by BNF in the experiment, an illustrative case is shown in Fig. 4, using the optimal parameter values obtained. Initially, ICBI is applied to the image shown in

[4]A numerical approximation for the Laplacian $\bigtriangledown^2$ of an image, was obtained as adopted by [32].

TABLE II

AVERAGE OPTIMAL PSNR AND SSIM $\pm$ RESPECTIVE STANDARD DEVIATION $\sigma$ OBTAINED FROM THE $4 \times 4$ MAGNIFICATION OF THE DOWNSCALED VERSIONS OF THE RGB IMAGES FROM *Morguefile25* SET USING THREE INTERPOLATION METHODS AND, FOR EACH INTERPOLATION METHOD, THREE SHARPENING FILTERS. FILTERS THAT ACHIEVED STATISTICALLY SIGNIFICANT INCREMENT IN AVERAGE PSNR OR SSIM AT A SIGNIFICANCE LEVEL $\alpha = 0.05$ ARE SHOWN IN BOLD.

| Similarity Measure | | | | | | | |
|---|---|---|---|---|---|---|---|
| PSNR | | | | SSIM | | | |
| **Method** | **Avg. PSNR$\pm\sigma$** | **Variation** | **P-value** | **Method** | **Avg. SSIM$\pm\sigma$** | **Variation** | **P-value** |
| NEDI (no filter) | 30.1379±2.866551 | | | NEDI (no filter) | 0.7487±0.119265 | | |
| NEDI+Laplace | 29.4592±2.297150 | -2.25% | <0.001 | NEDI+Laplace | 0.7205±0.117919 | -3.77% | <0.001 |
| **NEDI+LUM** | **30.3351±2.795938** | **+0.654%** | **<0.001** | NEDI+LUM | 0.7505±0.116610 | +0.236% | >0.999 |
| **NEDI+BNF** | **30.1648±2.864864** | **+0.0890%** | **<0.001** | **NEDI+BNF** | **0.7491±0.119154** | **+0.0518%** | **<0.001** |
| iNEDI (no filter) | 30.9594±2.901049 | | | iNEDI (no filter) | 0.7635±0.115679 | | |
| iNEDI+Laplace | 29.8303±2.268673 | -3.65% | <0.001 | iNEDI+Laplace | 0.7199±0.115720 | -5.70% | <0.001 |
| iNEDI+LUM | 30.9773±2.891314 | +0.0578% | 0.1003 | iNEDI+LUM | 0.7644±0.114383 | +0.125% | >0.999 |
| **iNEDI+BNF** | **30.9960±2.899985** | **+0.118%** | **<0.001** | **iNEDI+BNF** | **0.7644±0.115474** | **+0.117%** | **<0.001** |
| ICBI (no filter) | 30.3821±2.666460 | | | ICBI (no filter) | 0.7626±0.115997 | | |
| ICBI+Laplace | 29.9133±2.153304 | -1.54% | 0.03668 | ICBI+Laplace | 0.7582±0.110864 | -0.575% | 0.006129 |
| **ICBI+LUM** | **30.7089±2.653119** | **+1.08%** | **<0.001** | **ICBI+LUM** | **0.7654±0.113839** | **+0.369%** | **0.009151** |
| **ICBI+BNF** | **30.7757±2.823299** | **+1.30%** | **<0.001** | **ICBI+BNF** | **0.7635±0.115618** | **+0.121%** | **<0.001** |

Fig. 2b. The result for a small region of the image is shown in Fig. 4a. The results from the application of sharpening filters are shown in Figs. 4b, 4c, and 4d. The optimal parameters values, with respect to PSNR, adopted for this image are $\alpha^\star = 1$, $\bar{l}^\star = 0.62$ and $\beta^\star = 0.68$. The PSNR resulting from ICBI is 28.56. After each filter is applied, PSNR results are 28.26, 28.81 and 29.20, respectively from Laplace, LUM and BNF. Fig. 4d, which corresponds to the result from the BNF filter, shows an image with relatively sharp edges and any noticeable artifact. The other two filters resulted in images with visible artifacts. The edges visible in Fig. 4c, which corresponds to the application of the LUM filter, are very sharp but evident ringing artifacts exist. The BNF-filtered image in Fig. 4d is more pleasant to look when compared to the corresponding result from the $4\times4$ magnification using PCHIP and also filtered with BNF shown in Fig. 3b, since ICBI is a better algorithm and less artifacts were produced.

A note on the computational time of the proposed method should be mentioned, although a full experiment is out of the scope of this paper. The computational times of BNF were similar to the respective times obtained from LUM and much superior to the computational times obtained from the Laplacian filter for both magnification factors considered. LUM took 95.1s and 97.8s for filtering $512\times512$ RGB images at $2\times2$ and $4\times4$ magnification factors respectively, while BNF took 116.5s and 222.1s respectively for the same magnification factors. Laplace filter consumed only 0.016s on both cases[5]. The computational times for both LUM and BNF can be enhanced since our implementation of both methods in Octave can be much improved, while other programming languages should also be considered.

## V. CONCLUSION

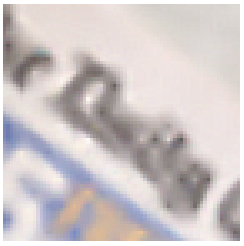
This work evaluates a novel sharpening filter that enhances the quality of zoomed images by replacing the intensity of interpolated pixels by the closest value of the local neighborhood

[5]Experiments were performed using GNU Octave version 4.4.0 over an Intel® Core™ i5-7200U CPU at 2.5GHz.

Fig. 4. The sharpening effect resulting from the application of three filters with optimal parameter values with respect to PSNR as illustrated by a small region of an image which was interpolated by a $4 \times 4$ factor using ICBI. The original image from *Morguefile25* set is shown in Fig. 2b.

(a) ICBI

(b) ICBI +Laplace



(c) ICBI +LUM

(d) ICBI +BNF

of low-resolution pixels. The filter is designed specifically for the post-interpolation stage, aiming to enhance the quality of images that were magnified. The quantitative evaluation revealed that BNF filter consistently increased both PSNR and SSIM for RGB images interpolated using four algorithms from the literature. The results are statistically significant.

Qualitative evaluations show that images obtained from BNF presented fewer visual artifacts when compared to the reference upscaled images. Although no explicit edge detection mechanism is applied, the filter was able to improve the visual aspect of the edges between objects, in different situations considered. This feature is worthwhile in many applications, such as segmentation and computer vision.

The evaluation of the proposed method should be much extended, including comparison to other sharpening filters and the adoption of other interpolation algorithms as the reference, since the filter proposed presents no restrictions for the adoption of interpolation methods other than those used here. For instance, the works recently proposed in [7] and [33] should be considered.

The proposed method is not restricted to the magnification factors applied, therefore other alternatives could be adopted in future evaluations. The investigation on how the parameter $d$ affects the behavior and the results of the filter are also of interest and should be the object of future research. This should consider the fact that the parameter $d$ affects computational time, therefore greater values for $d$ are not recommended.

## REFERENCES


[1] A. Jurio, M. Pagola, R. Mesiar, G. Beliakov, and H. Bustince, "Image magnification using interval information," *IEEE Transactions on Image Processing*, vol. 20, no. 11, pp. 3112–3123, 2011.

[2] W. Dong, L. Zhang, R. Lukac, and G. Shi, "Sparse representation based image interpolation with nonlocal autoregressive modeling," *IEEE Transactions on Image Processing*, vol. 22, no. 4, pp. 1382–1394, 2013.

[3] A. Amanatiadis and I. Andreadis, "A survey on evaluation methods for image interpolation," *Measurement Science and Technology*, vol. 20, no. 10, p. 104015, 2009.

[4] A. Giachetti and N. Asuni, "Real-time artifact-free image upscaling," *IEEE Transactions on Image Processing*, vol. 20, no. 10, pp. 2760–2768, 2011.

[5] Y. J. Lee and J. Yoon, "Nonlinear image upsampling method based on radial basis function interpolation," *IEEE Transactions on Image Processing*, vol. 19, no. 10, pp. 2682–2692, 2010.

[6] K. Nasrollahi and T. B. Moeslund, "Super-resolution: a comprehensive survey," *Machine vision and applications*, vol. 25, no. 6, pp. 1423–1468, 2014.

[7] Q. Song, R. Xiong, D. Liu, Z. Xiong, F. Wu, and W. Gao, "Fast image super-resolution via local adaptive gradient field sharpening transform," *IEEE Transactions on Image Processing*, vol. 27, no. 4, pp. 1966–1980, 2018.

[8] S. Mallat and G. Yu, "Super-resolution with sparse mixing estimators," *IEEE transactions on image processing*, vol. 19, no. 11, pp. 2889–2900, 2010.

[9] X. Zhang and X. Wu, "Image interpolation by adaptive 2-d autoregressive modeling and soft-decision estimation," *IEEE transactions on image processing*, vol. 17, no. 6, pp. 887–896, 2008.

[10] F. Zhou, W. Yang, and Q. Liao, "Interpolation-based image super-resolution using multisurface fitting," *IEEE Transactions on Image Processing*, vol. 21, no. 7, pp. 3312–3318, 2012.

[11] K. Nguyen, C. Fookes, S. Sridharan, M. Tistarelli, and M. Nixon, "Super-resolution for biometrics: A comprehensive survey," *Pattern Recognition*, vol. 78, pp. 23–42, 2018.

[12] A. Amanatiadis and I. Andreadis, "A survey on evaluation methods for image interpolation," *Measurement Science and Technology*, vol. 20, no. 10, p. 104015, 2009.

[13] A. Giachetti and N. Asuni, "Fast artifacts-free image interpolation," in *Proceedings of the British Machine Vision Conference*, 2008, pp. 13–1.

[14] L. Pan, W. Yan, and H. Zheng, "Super-resolution from a single image based on local self-similarity," *Multimedia Tools and Application*, vol. 75(18), pp. 11 037–1105, 2016.

[15] L. R. Emmendorfer, "An empirical evaluation of sharpening filters applied to magnified images," in *Proceedings of the 2020 IWSSIP'20: 27th International Conference on Systems, Signals and Image Processing*. Niterói, RJ, Brazil: Institute of Computing at Fluminense Federal University (IC-UFF), 2020, pp. 187–192.

[16] T. Horiuchi, K. Watanabe, and S. Tominaga, "Adaptive filtering for color image sharpening and denoising," in *14th International Conference of Image Analysis and Processing-Workshops (ICIAPW 2007)*. IEEE, 2007, pp. 196–201.

[17] R. C. Gonzalez and R. E. Woods, *Digital image processing*, 3rd ed. London: Pearson Education, 2008.

[18] R. C. Hardie and C. Boncelet, "Lum filters: a class of rank-order-based filters for smoothing and sharpening," *IEEE transactions on signal processing*, vol. 41, no. 3, pp. 1061–1076, 1993.

[19] H. Kotera and W. Hui, "Multiscale image sharpening adaptive to edge profile," *Journal of Electronic Imaging*, vol. 14, no. 1, p. 013002, 2005.

[20] K. Zuiderveld, "Contrast limited adaptive histogram equalization," in *Graphics gems IV*. Academic Press Professional, Inc., 1994, pp. 474–485.

[21] M. Fischer, J. L. Paredes, and G. R. Arce, "Weighted median image sharpeners for the world wide web," *IEEE Transactions on Image Processing*, vol. 11, no. 7, pp. 717–727, 2002.

[22] T. C. Aysal and K. E. Barner, "Quadratic weighted median filters for edge enhancement of noisy images," *IEEE Transactions on Image Processing*, vol. 15, no. 11, pp. 3294–3310, 2006.

[23] G. R. Arce and J. L. Paredes, "Image enhancement and analysis with weighted medians," in *Nonlinear Image Processing*. Orlando: Academic Press, Inc, 2001, pp. 27–67.

[24] P. Zafeiridis, N. Papamarkos, S. Goumas, and I. Seimenis, "A new sharpening technique for medical images using wavelets and image fusion." *Journal of Engineering Science & Technology Review*, vol. 9, no. 3, 2016.

[25] K.-S. Peng, F.-C. Lin, and K.-T. Teng, "Efficient image resolution enhancement using edge-directed unsharp masking sharpening for real-time ASIC applications," *Journal of Computer Science & Systems Biology*, vol. 8, no. 3, p. 174, 2015.

[26] J. L. Clark, C. P. Wadhwani, K. Abramovitch, D. D. Rice, and M. T. Kattadiyil, "Effect of image sharpening on radiographic image quality," *The Journal of prosthetic dentistry*, vol. 120, no. 6, pp. 927–933, 2018.

[27] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli *et al.*, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[28] X. Li and M. T. Orchard, "New edge-directed interpolation," *IEEE Transactions on Image Processing*, vol. 10, no. 10, pp. 1521–1527, 2001.

[29] N. Asuni and A. Giachetti, "Accuracy improvements and artifacts removal in edge based image interpolation." *VISAPP (1)*, vol. 8, pp. 58–65, 2008.

[30] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 898–916, May 2011.

[31] [Online], Available: http://morguefile.com, 2020.

[32] J. W. Eaton, D. Bateman, S. Hauberg, and R. Wehbring, *GNU Octave version 4.2.2 manual: a high-level interactive language for numerical computations*, 2018.

[33] Y. Romano, J. Isidoro, and P. Milanfar, "RAISR: rapid and accurate image super resolution," *IEEE Transactions on Computational Imaging*, vol. 3, no. 1, pp. 110–125, 2016.