# Image and Video Upscaling from Local Self-Examples

GILAD FREEDMAN and RAANAN FATTAL
Hebrew University of Jerusalem

We propose a new high-quality and efficient single-image upscaling technique that extends existing example-based super-resolution frameworks. In our approach we do not rely on an external example database or use the whole input image as a source for example patches. Instead, we follow a *local self-similarity* assumption on natural images and extract patches from extremely localized regions in the input image. This allows us to reduce considerably the nearest-patch search time without compromising quality in most images. Tests, that we perform and report, show that the local-self similarity assumption holds better for small scaling factors where there are more example patches of greater relevance. We implement these small scalings using dedicated novel non-dyadic filter banks, that we derive based on principles that model the upscaling process. Moreover, the new filters are nearly-biorthogonal and hence produce high-resolution images that are highly consistent with the input image without solving implicit back-projection equations. The local and explicit nature of our algorithm makes it simple, efficient and allows a trivial parallel implementation on a GPU. We demonstrate the new method ability to produce high-quality resolution enhancement, its application to video sequences with no algorithmic modification, and its efficiency to perform real-time enhancement of low-resolution video standard into recent high-definition formats.

## 1. INTRODUCTION

Increasing image resolution, or *image upscaling*, is a challenging and fundamental image-editing operation of high practical and theoretical importance. While nowadays digital cameras produce high-resolution images, there are tremendously many existing low-resolution images as well as low-grade sensors, found in mobile devices and surveillance systems, that will benefit resolution enhancement. At its essence, image upscaling requires the prediction of millions of unknown pixel values based on the input pixels, which constitute a small fraction of that number. This difficult task challenges our understanding of natural images and the regularities they exhibit. Upscaling is also intimately related to a variety of other problems such as image inpainting, deblurring, denoising, and compression.

Perhaps the simplest form of single-image upscaling predicts the new pixels using analytical interpolation formulae, e.g., the bilinear and bicubic schemes. However, natural images contain strong discontinuities, such as object edges, and therefore do not obey the analytical smoothness these methods assume. This results in several noticeable artifacts along the edges, such as ringing, staircasing (also known as 'jaggies'), and blurring effects. An alternative approach, suggested by by Freeman et al. [2000; 2002], uses an example-based Markov random model to relate image pixels at two different scales. This model uses a universal set of example patches to predict the missing upper frequency band of the upsampled image. While this approach is capable of adding detail and sharpening edges in the output image, it also produces considerable amount of noise and irregularities along the edges due to shortage in relevant examples and errors in the approximate nearest-patch search. Based on prior research on image compression, Ebrahimi and Vrscay [2007] suggest to use the input image itself as the source for examples. While this typically provides a limited number of examples, compared to a universal database, it contains much more *relevant* patches.

In this paper we propose a new high-quality and efficient single-image upscaling technique that extends existing example-based super-resolution frameworks in several aspects. We point out and exploit a *local* scale invariance in natural images where small patches are very similar to *themselves* upon small scaling factors. This property holds for various image singularities such as straight and corner edges, as shown in Figure 1. We use this observation to take the approach of Ebrahimi and Vrscay one step farther and search for example patches at extremely *localized regions* in the input image. We compare this localized search with other alternatives for obtaining example patches and show that it performs significantly better in terms of both computation time and matching error.
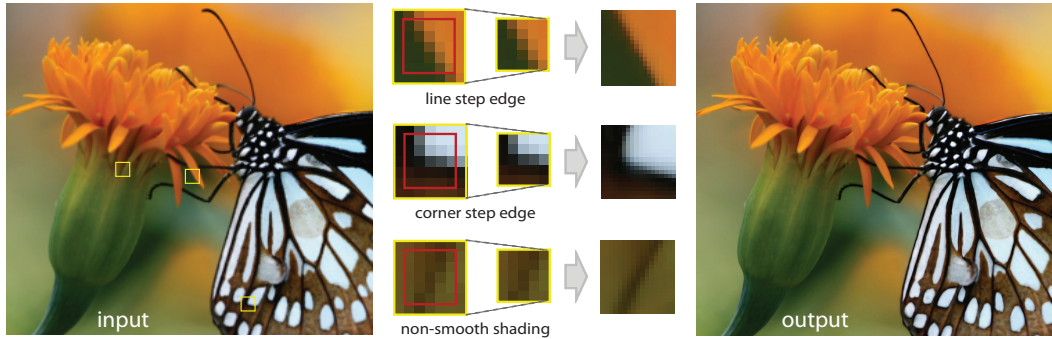
**Fig. 1:** *Image upscaling by exploiting local scale-similarity in natural images. The patches (yellow) when downscaled are very similar to their cropped version (red). This relation holds for various types of singularities.*

Further tests, we report here, show that the scale invariance assumption holds better for small scaling factors, where more example patches of a greater relevance are found. Therefore, we perform multiple upscaling steps of small scaling factors to achieve the desired magnification size. We implement these non-dyadic scalings using dedicated novel filter banks which we derive for general $N+1:N$ upsampling and downsampling ratios. The new filters are designed based on several principals that we use to model the upscaling process. Among these principals is the requirement that the filters have to be nearly-biorthogonal such that the upsampled image is consistent with the input image at an early stage, before the missing high-frequency band is predicted. This exploits better the input data by reducing the amount of prediction needed at the example-based learning stage and, as we show, leads to an increased visual realism. While previous methods impose this requirement by solving large systems of back-projection equations, the new filter banks achieve this consistency via explicit and thus efficient computation.

We demonstrate the new method ability to produce high-quality resolution enhancement, compare it to existing state-of-the-art methods, and show that its applies to video sequences with no modifications in the algorithm. We also show how the localized searches and explicit filter computations permit very efficient implementation and report real-time GPU performance when enhancing low-resolution video to high-definition format.

## 2.  PREVIOUS WORK

Image upscaling was studied extensively by the computer graphics, machine vision, and image processing communities. The methods developed over the years differ in their formulation and underline prior image model and the input data they use. Here we briefly describe the main approaches to the problem and the principles behind them. We focus on single-image upscaling methods which is the assumed settings of our new method.

The classic and simplest approach uses linear interpolation in order to predict intermediate pixels values. This method is usually implemented using linear filtering, such as the bilinear and bicubic filters, and it is commonly found in commercial software. These interpolation kernels are designed for spatially smooth or band-limited signals which is often not the case in natural image. Real-world images often contain singularities such as edges and high-frequency textured regions. As a result, these methods suffer from various

edge-related visual artifacts such as ringing, aliasing, jaggies, and blurring. Thevenaz et al. [2000] provide a more elaborate survey of these methods and their evaluation.

More sophisticated methods adapt the interpolation weights based on the image content. For example, Li et al. [2001] adapt the interpolation weights according to the local edge orientations and Su et al. [2004] choose three out of the four nearest pixels for linear interpolation. This allows to reduce the ringing effects and obtain somewhat sharper edges. Non-quadratic smoothness functionals yield a different type of non-linear image regularization which can be used for upscaling. For example, Aly and Dubois [2005] enlarge images by minimizing the total variation functional. Shan et al. [2008] minimize a similar metric using a sophisticated feedback-control framework that keeps the output image consistent with the input image when downscaling it to the input resolution.

Inspired by recent studies of natural image statistics, several methods use random Markov field models to define a probability density over the space of upscaled images. The output image, in many cases, is computed by maximizing these models. These approaches can be divided to two main classes; ones that define non-parametric example-based models and ones that are based on analytical image modeling.

Example-based image enlargement is explored by Freeman et al. [2000] and further developed in [Freeman et al. 2002]. This image prediction model relies on a database of example patches that are decomposed into their a low-frequency band, i.e., a smoothed version, and the residual higher frequency band. The input image is interpolated to a higher resolution using analytic interpolation and the missing high-frequency band is then predicted from the example patches. The matching is performed according to the low-frequency component of the example patches. This approach is capable of producing plausible fine details across the image, both at object edge and in fine-textured regions. However, lack of relevant examples in the database results in fairly noisy images, that show irregularities along curved edges. The use of larger databases is more time consuming due to the added comparisons in the nearest-neighbor searches. The use of approximate nearest neighbor searches offers a limited solution, as it introduces its own errors. Tappen et al. [2004] also use a patch-based model and require the output to be consistent with the input.

Motivated by earlier works that study the fractal nature of images and its application to image compression [Barnsley 1988], Robert et al. [1997] and Vrscay et al. [2002] interpolate images

using fractal compression schemes which contain extra decoding steps. This approach suffers from strong block artifacts which can be reduced using overlapping range blocks [Reusens 1994; Polidori and Dugelay 1995]. Based on these works, Ebrahimi and Vrscay [2007] use the input image at multiple smaller scales as the source for example patches, relying on self-similarity in small patches. While this offers an example database of a limited size compared to universal databases, as we show later, this example data is considerably more *relevant* to the input image being enlarged. Suetake et al. [2008] also use the input image to compute an example codebook which is later used to estimate the missing high-frequency band, in a framework similar to [Freeman et al. 2002].

Recently, several parametric image models were proposed for upscaling. These methods fit analytical models to describe various image features that show statistical dependency at different scales. Fattal [2007] models the relation between edge descriptors, extracted from the input, and gradients at a higher resolution. A fully analytical prior for the reconstructed edge profile is used by Sun et al. [2008]. These approaches are considerably faster than their example-based counterparts and are capable of reproducing sharp edges with no apparent noise. Nevertheless, the resulting images tend to appear somewhat unrealistic as they are made of generic edges that often separate color plateaus. Sun et al. [2003] describe a Markov random field that combines example-based and parametric modeling together.

Besides single image upscaling, many works deal with multi-frame super-resolution where multiple shots of the same scene, taken at translational offsets, are used to generate a single high-resolution image of the scene. Farsiu et al. [2004] review this literature and propose to use robust regularization to deal with the noise that limits this operation [Lin and Shum 2004]. Given high-resolution photographs of a static scene Bhat et al. [2007] enhance videos of that scene by rendering pixels from the photographs. Recently, Glasner et al. [2009] unify the multi-frame and example-based super-resolution techniques and derive a single-image method. This method uses the formalism of multi-frame super-resolution yet relies on self-similarities in the image to obtain samples differing by sub-pixel offsets.

## 3. UPSCALING SCHEME

The basic upscaling scheme we use is closely related to the framework previously used by Freeman et al. [2002] and others [Tappen et al. 2004; Suetake et al. 2008]. However, we replace most of the components it uses with novel, application-specific, components that we describe here. The first contribution is an alternative source of example patches that we propose to use. In Section 3.1 we discuss and measure a refined scale similarity property in natural images. The similarity assumptions used by existing methods exploit similarities across the image and at multiple scales of it. We refine this assumption and observe that various singular features in natural images are similar to *themselves* under small scaling factors. This property, which we call *local self-similarity*, allows us to search and find relevant example patches in very restricted neighborhoods around the same relative coordinates in the input image. This approach reduces substantially the nearest-patch retrieval times compared to global image searches or searching in external databases. This is achieved at no apparent visual compromise in the majority of images that do not contain exact repetitions at multiple scales.
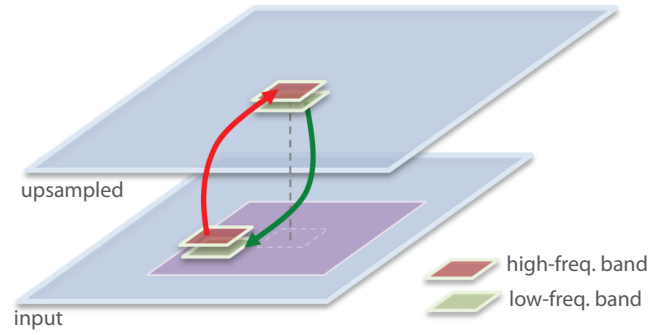


**Fig. 2:** *Upscaling scheme. A patch of lower frequency band from the upsampled image is matched (green arrow) with its nearest patch within a small window in the low-passed input image (purple). The upper frequency band of the matched patch in the input is used (red arrow) to fill in the missing upper band in the output upsampled image.*

In our experiments we validate that very relevant patches can be found in the restricted relative neighborhoods in the input image when applying scaling at *small* factors. Therefore, in order to make this approach effective, we maximize the amount of data present in the example patches and its relevance by scaling the image in multiple steps of small magnification factors. This is done using by our second new component; new dedicated *non-dyadic* filter banks. These filter banks, which we describe in Section 3.2, define the interpolation and smoothing operators $\mathcal{U}$ and $\mathcal{D}$ that perform scalings at factors smaller than two. Another desirable property these new filter banks achieve is consistency with the input image through explicit computations, without solving the back-projection equations as previous methods do.

We begin by describing the new upscaling scheme, depicted in Figure 2. Given an input image $I_0$, defined on a coarse grid of pixels $\mathcal{G}_0$, we start off by interpolating it to a finer grid $\mathcal{G}_1$ using our linear interpolation operator $\mathcal{U}$ that maps images from $\mathcal{G}_l$ to $\mathcal{G}_{l+1}$, where $\mathcal{G}_l$ are rectangular pixel grids with resolutions that increase by the scaling factor (that grow with $l$). This *initial* upsampled image $L_1 = \mathcal{U}(I_0)$ lacks a fraction of its upper frequency band, proportional to the scaling factor. This missing band is then predicted using a non-parametric patch-based model that does not rely on external example databases but rather exploits the *local self-similarity* assumption as follows. Example patches are extracted from a smoothed version of the input image $L_0 = \mathcal{U}(\mathcal{D}(I_0))$, where $\mathcal{D}$ is a downsampling operator that maps images from $\mathcal{G}_l$ to $\mathcal{G}_{l-1}$ and is also defined by our dedicated filter bank. The high-frequency prediction is done by first matching every patch $\mathbf{p} \subset \mathcal{G}_1$ in the upsampled image $L_1$ with its most-similar patch $\mathbf{q}(\mathbf{p}) \subset \mathcal{G}_0$ in the smoothed input $L_0$. This search is *not* performed against every patch in $L_0$, but rather against restricted small windows (purple window in the Figure 2) centered around the same relative coordinates in $\mathcal{G}_0$ as the center coordinates of the query patch $\mathbf{p}$ in $\mathcal{G}_1$. As we explain later, this requires the two images to be spectrally compatible which we ensure when designing $\mathcal{U}$ and $\mathcal{D}$. The complement high-frequency content in the input image at the matched patch, $H_0(\mathbf{q}) = I_0(\mathbf{q}) - L_0(\mathbf{q})$, is used to fill-in the missing higher frequency band in the upsampled image by simply pasting it, i.e., $I_1(\mathbf{p}) = L_1(\mathbf{p}) + H_0(\mathbf{q}(\mathbf{p}))$. Different accounts for the same pixel, due to overlaps between nearby patches are averaged together. The particularities of this scheme are detailed in the next sections.
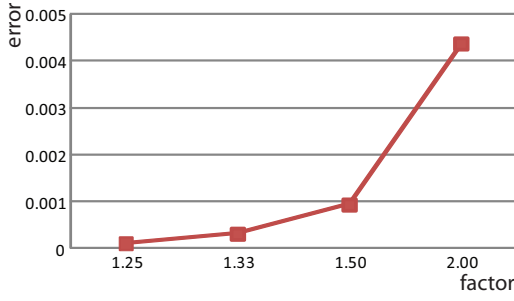
**Fig. 3:** *Graph showing the error in localized searches when upsampling an image at different scaling factors.*

## 3.1   Local Self-Similarity

Freeman et al. [2002] use a universal example database of small patches taken from arbitrary natural images. Barnsley [1988] and others [Vrscay 2002; Freeman et al. 2000; Ebrahimi and Vrscay 2007; Suetake et al. 2008; Glasner et al. 2009] rely on self similarities within the image; small patches in natural images tend to repeat themselves within the image and across its scales. This allows to replace the external database with the input image itself, taken at smaller scales, as the source for example patches. While this provides a limited number of example patches, compared to external databases of an arbitrary size, the patches found within the input image are more *relevant* for upscaling it. Thus, in many cases one can use less examples and obtain the same or even better results while reducing the time cost involved in the nearest-neighbor searches which is known to be the major bottleneck of nonparametric example-based image models.

We refine the self-similarity observation in natural images and show that various singularities such as edges, which commonly appear in natural images and require resolution enhancement, are invariant to scaling transformations and are hence similar to *themselves* at an instance basis. We call this property *local self-similarity* since it implies that that relevant example patches can be found at a very restricted set of patches; for every patch in the image, very similar patches can be found in its downscaled (or smoothed) version at localized regions around the same relative coordinates. This is the case for patches containing discontinuities in intensity, i.e., edges in the image, discontinuous first derivative, i.e., shading of faceted surfaces. These isolated singularities can appear at different geometrical shapes, e.g., lines, corners, T-junction, arcs, etc., as shown in Figure 1. We exploit this local similarity to reduce the work involved in the nearest-patch search from being dependent on the number of image pixels, down to being extremely small and fixed. We describe here several experiments that quantify the extent to which the local self-similarity holds and compare its use with other patch search methods.

In the first experiment, we attempt to quantify how well the local self-similarity holds at various scaling factors. In Figure 3 we show mean $L_1$ error between query and retrieved patches, computed in RGB color space (pixel values between zero and one) after normalizing each color channel by its average intensity. The query patches are windows of 5-by-5 pixels and the search was performed in windows of 10-by-10 pixels. The graph in the figure shows clearly that error increases as the scaling ratio grows larger. This can be explained as follows. Large scaling factors involves stronger smoothing when computing the example image. This is
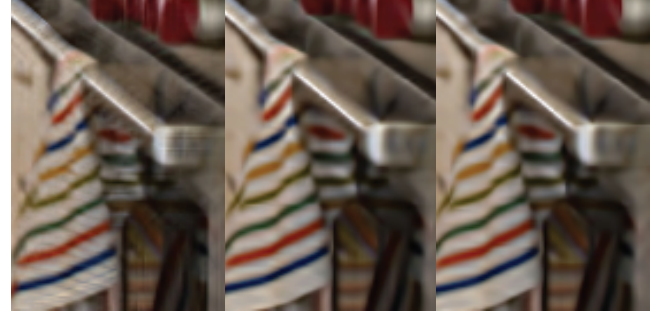


**Fig. 4:** *Comparison between different scaling steps, used to achieve a final magnification factor of 3. The scaling sequences used are: (left) 2:1, 3:2, (middle) 5:4, 5:4, 4:3, 3:2, and (right) 5:4 repeated five times. The 2:1 scaling were performed using* Bior *6.8 wavelets.*
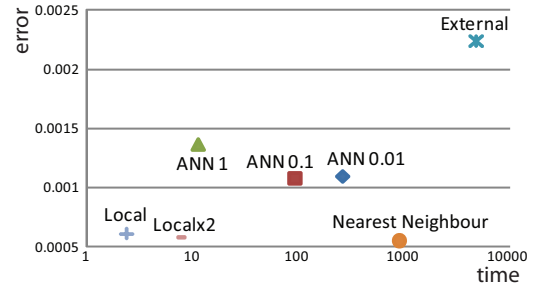


**Fig. 5:** *Graph showing the error versus the time cost of various search strategies: local searches within windows of* 10*-by-*10 *pixels (*Local*), windows of* 20*-by-*20 *pixels (*Localx2*), exact nearest search in the input image (*Nearest Neighbor*) and an external database (*External*), and approximate nearest-neighbor search in the entire image (*ANN*) at different error bounds. These tests were performed with scaling factor of 5:4. Note that the time axis of the graph is shown in logarithmic scale.*

equivalent to (and implemented by) downscaling the input image by large factors. In this process, image features will become closer to one another and the example patches will no longer contain pure isolated singularities that obey the scale invariance assumption. In Figure 4 we show how different scaling steps effect the final upscaled image. The artifacts produced by large factors, dyadic in this case, are clear. Given this, we upscale images by performing multiple scaling steps of small factors to achieve the desired final magnification. In the next section we describe how we implement these small non-dyadic scalings using dedicated filter banks that we design.

In the second experiment we compare the quality of the patches retrieved by various approaches as well as the running times. Here also, we use patches of 5-by-5 pixels in normalized RGB color space and report the mean values estimated from six test images upscaled by a factor of 5:4. Figure 5 shows the results obtained using: local searches within windows of 10-by-10 pixels, windows of 20-by-20 pixels, and exact nearest search and approximate nearest-neighbor[1] searches in the entire input image. The graph shows

---

[1]We used the ANN package by Mount & Ayra, found at: `http://www.cs.umd.edu/~ mount/ANN` to perform the approximate nearest-neighbor searches.
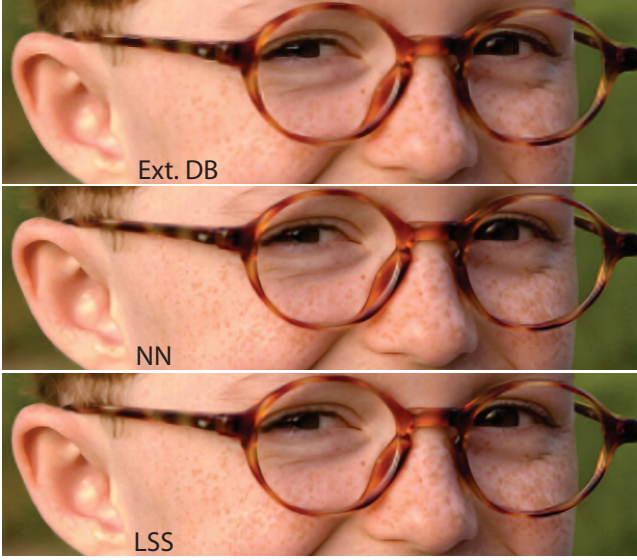
**Fig. 6:** *Images upscaled by a factor of 3 using nearest search in an external database (Ext. DB), nearest patch within the entire image (NN), and using our localized search (LSS).*



**Fig. 7:** *Image with repetitions. Top images shows the input bottom images show: (left) input pixels, (middle) output produced by our method, and (right) the results of Glasner et al. [2009].*

local self-similarities found along most of the edges in general natural images. In addition, in the approach of Glasner et al. the output image is computed by solving the super-resolution as well as back-projection systems of large linear equations. As we describe in the next sections, we synthesize the upscaled image using custom filter banks that consist of local and explicit calculations. Altogether, our approach acts locally, both when searching for example patches and when constructing the output image, and allow a fully parallel implementation, as we describe in Section 4.

### 3.2   Non-Dyadic Filter Banks

The main conclusion from the tests described above is that small localized patch searches are effective as long as the scaling factors are small. Therefore, we perform multiple upscaling steps of small factors to achieve the desired magnification. The upscaling scheme, described in Section 3, uses analytical linear interpolation $\mathcal{U}$ and smoothing $\mathcal{D}$ operators to compute the initial upsampled image and the smoothed input image, used for generating the example patches. We show here that the choice of these operators is consequential for our application and point out several conditions these filter banks should obey in order to model the image upscaling process.

Dyadic image upscaling, where the image dimensions are doubled, typically consists of interpolating the image by adding zeros between every two pixels followed by a filtering step. Dyadic downscaling consists of filtering the image first and then subsampling every other pixel. The downscaling operation is identical to the computation of the coarser level approximation coefficients in the forward wavelet transform, and the upscaling corresponds to the inverse wavelet transform, applied without adding any detail (wavelet) component. In fact, wavelet theory [Mallat 1999] is primarily concerned with the design and analysis of such scaling schemes and filter banks. This vast literature offers a countless number of filter banks, obeying all or most of the requirements we are about to discuss here, for computing dyadic transformations. However, very little is available for non-dyadic case, where the scaling factor is other than two. Xiong et al. [2006] use the lifting scheme to compute a 3:2 scaling for image coding. Their construction maps a linear function to the same values it contained, while skipping every third value. Hence it does not reconstruct first-order polynomials which, as we shall discuss below, is very important for our application. Pollock et al. [2007] describe orthogonal sym-

that the local search we propose is more accurate and considerably faster than the use of kd-based approximate nearest-neighbor search algorithm set to various error bounds. Searching for the nearest patch exhaustively, in the entire input image, does not achieve a significant gain in accuracy compared to the localized searches—in fact there is an 80% agreement between the two. Figure 6 confirms that indeed no visual difference is found between these two search strategies. Searching within windows larger than 10-by-10 pixels does not achieve a significant gain in quality yet takes more than four times longer to compute.

Furthermore, we tested the use of an external database, consisting on the six images used by Freeman et al. [2002] enriched by another highly-detailed image containing the same number of total pixels. Despite having more example patches and the use of an accurate nearest-neighbor search, this option resulted in noticeable higher errors, as shown in Figure 6. Besides the lower relevance of this database, compared to the image itself, nearest patches computed based on low-frequency band may be ambiguous. Indeed, Freeman et al. match patches also based on the high-frequency component which is being built. This solution introduces spatial dependencies when searching for patches, which undermines the locality and explicitness of computation.

**Relation to existing approaches.** The local self-similarity we present validate and exploit in this work is one of the main novelties we propose for example-based image upscaling. Our tests show that the local searches, this similarity allows, achieve a significant speedup at almost no quality cost compared to searching the entire image. However, Glasner et al. [2009] show that by searching within the entire input image and *across* scales, different instances of the same feature may be found. Figure 7 shows a case where such repetitions are found and exploited. In order to find these repetitions in the data Glasner et al. perform a thorough search at multiple scales that differ by small non-dyadic factors and small translational offsets. While this approach increases the nearest-patch search efforts, this type of repetitions is scarce compared to the
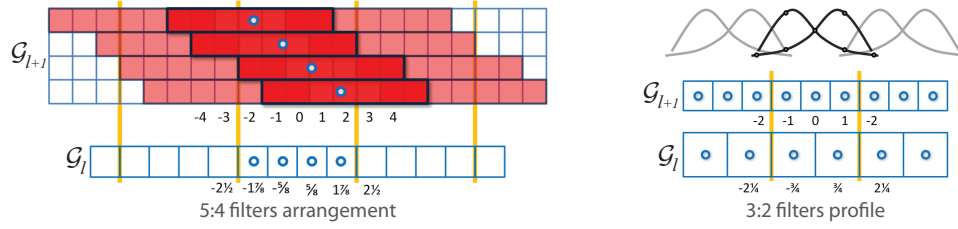
**Fig. 8:** *Grid relation and filter placements. Left illustration shows a coarse grid (bottom) placed against the finer grid (top) in a 5:4 ratio. The placement of the non-zero coefficients on the fine grid is shown in red. At the right we illustrate the profiles of two filters in a 3:2 scaling factor.*

metric Shannon-like wavelets that are infinite and hence do not fit our needs. While wavelet packets analysis [Mallat 1999] generalizes the spectral tiling tree, it is still largely implemented in the common dyadic spatial setting.

Here we derive the formalism that extends the dyadic scaling of wavelet transforms to $N\!+\!1\!:\!N$ scaling factors. Figure 8 illustrates this for the cases of $N\!=\!2$ and $N\!=\!4$. As shown in the figure, the relative placement of the coarse $\mathcal{G}_l$ and fine $\mathcal{G}_{l+1}$ grid points has a periodicity of $N$. Besides the dyadic case, $N\!=\!1$, there is no reason to expect the filters to be strictly translation invariant; they can differ within each period. For example, if we require the filters to map a linear ramp function sampled at one grid to the same function sampled at a coarser (or finer) grid, the filters weights will have to adapt to the different relative offsets between the grids (seen in the figure) and therefore be different within each period of $N$ filters. Therefore, with no further spatial dependency, the $N\!+\!1\!:\!N$ transformations consist of filtering which is translation invariant to periods of $N + 1$ grid points and hence require $N$ distinct filters in order to handle the irregular grids relation. In practice, this means that the downscaling operator can be computed using $N$ standard translation-invariant filtering operations, performed in $\mathcal{G}_{l+1}$, followed by subsampling each filtered image every $N + 1$ pixels to produce the total of $N$ values in $\mathcal{G}_l$. Formally, this is given by

$$\mathcal{D}(I)(n) = (I * \bar{d}_p)\big((N{+}1)q + p\big), \qquad (1)$$

where $p = n \bmod N$, $q = (n - p)/N$, the filters $d_1, .., d_N$ are the $N$ distinct smoothing filters, the filter mirroring is given by $\bar{d}[n] = d[-n]$, and $*$ denotes the discrete convolution. The analog extension of the dyadic case applies to the upsampling step; every sample within the period of $N$ is filtered with a different upsampling filter, $u_1, .., u_N$. These filtered images are then summed as following,

$$\mathcal{U}(I)(n) = \sum_{k=1}^{N}(\uparrow I * \bar{u}_k)(n), \qquad (2)$$

where the zero upsampling operator is defined by $(\uparrow I)[(N{+}1)n] = I[n]$ and zero otherwise.

Filters that constitute a mapping and its inverse are known as *biorthogonal* filters [Mallat 1999]. In our derivation this relation can be required if we first upsample and then downsample, i.e.,

$$\mathcal{D}\big(\mathcal{U}(I)\big) = I. \qquad (3)$$

Applying the operators in the reverse order cannot be expected to result in an identity mapping since $\mathcal{G}_l$ and $\mathcal{G}_{l+1}$ are of different space dimension. Formally, relation (3), between the operators, can

be expressed in terms of the filters by

$$\langle u_i[n], d_j[n - (N+1)k]\rangle = \delta_k \cdot \delta_{i-j}, \qquad (4)$$

for every integer $k$ and $1 \le i, j \le N$, where $\langle \cdot, \cdot \rangle$ is the usual dot-product and $\delta_0 = 1$ and zero otherwise. In Section 3.3 we discuss the importance of producing an upsampled image that is consistent with the input upon downsampling.

The lack of translation invariance, or the translation invariance in periods of $N+1$ grid points, introduces a minor complication when generating low- and high-frequency example patches for the high-frequency prediction step of our algorithm. Being different from one another, the $N$ distinct filters respond differently to the same input signal. Therefore, their output must not be mixed in the high-pass prediction step when comparing and pasting patches. To avoid this, we search for examples in offsets of $N + 1$ pixels, such that the same filter responses are always aligned together. We then compensate for this reduction in the number of example patches by creating multiple example images $L_0 = \mathcal{U}\big(\mathcal{D}(I_0)\big)$ that are produced by offsetting the input image $I_0$ by $1, .., N+1$ pixels along each axis. Thus, altogether the number of example patches remains the same as if we searched in offsets of a single pixel and the filtered values are not intermixed.

**Connection to wavelet bases.** This derivation does not include the high-pass wavelet filters, which normally span the high frequency subspace of the finer space $\mathcal{G}_{l+1}$ that is the complement of the space spanned by the upsampling filters $u_i$. The reason is that we do not use them in our upscaling scheme; the high-pass prediction step fills-in the high-pass layer directly, in pixel values and not detail coefficients (that normally multiply the wavelet filters to get pixel values). Working in pixel resolution avoids the loss of spatial resolution that the wavelet filters introduce and leads to better quality results in the patch-based synthesis.

However, by not restricting the synthesis of the high-frequency layer to the complement high-frequency wavelet space, the lower frequency approximation layer will also be affected by this step, i.e., the upsampled image spanned by $u_i$. Thus, the consistency with the input will be undermined. However, our tests show that the predicted high-frequency layer has an unnoticeable effect upon downsampling and deviates the intensities of the synthesized image from the input by about 1%.

## 3.3 Filter Design

Here we describe the guiding principles which we follow to design the upsampling and downsampling filters such that they model the

**Fig. 9:** *A comparison between nearly biorthogonal filters and filters designed with no such consideration (omitting the $O_1^{\mathcal{U}}$ term at the (9) in the Appendix). The left pair of images show the initial upsampled image, without running the high-frequency prediction step, after it is downsampled to the input resolution, i.e., $\mathcal{D}(\mathcal{U}(I_0))$. The right pair shows the output of our upscaling scheme. The salient edges are well reproduced in both cases however the lack of biorthogonality leads to a loss of fine details in the image which makes it appear less realistic.*

image upscaling process. We do this by introducing several conditions we believe image upscaling frameworks, similar to ours, should obey in order to model properly the problem and allow efficient computing. This conditions are used, at the Appendix, for designing the filtering operators we use in our upscaling scheme. In this paper we demonstrate the results obtained by taking these considerations into account. While it is hard to meet even all of the presented conditions using compact filters, more considerations can be incorporated to this framework.

**(C1) Uniform scaling.** When upscaling and downscaling an image we want the resulting image to differ by a similarity transformation; a spatially-uniform scaling transformation. Such a transformation changes the distance between every two points by a fixed factor, the scaling factor between the image grids. This property can be imposed on our upsampling and downsampling operators by the aid of linear functions. A linear function shows a fixed difference, $I(x + dx) - I(x)$, between points of a fixed distance, $dx$. Therefore, we require our operators to preserve the shape of linear functions, i.e., map a linear function defined at $\mathcal{G}_l$ to a linear function defined on $\mathcal{G}_{l+1}$ and vice versa. This condition of exactly reproducing linear functions is the counterpart of the vanishing moments condition, commonly used in the wavelet design literature [Mallat 1999].

**(C2) Low frequency span.** Cameras have a finite point spread function as well as contain a blurring anti-aliasing filter. This is used to restrict the bandwidth of a signal (scene) to approximately satisfy the sampling theorem according to the sensor sampling rate. This has implications on both $\mathcal{D}$ and $\mathcal{U}$. The downscaling operator $\mathcal{D}$ should therefore model the difference in the amount of blurring, needed before sampling signals at $\mathcal{G}_{l+1}$, and the stronger blurring needed for a lower sampling rate of $\mathcal{G}_l$. This lies behind the common practice of designing $\mathcal{D}$ to be a low-pass filter that transfers the lower frequency band [Pratt 2001]. The length of this frequency band is roughly proportional to the scaling ratio, $N/(N+1)$ in our case.

A similar condition applies to the interpolation operator $\mathcal{U}$. As discussed above, the camera filtering limits the signal's spectrum according to the low sampling rate of the input image. The initial upsampled image $L_1$ should contain this data, rendered at a higher sampling rate. Therefore, the reproducing kernels of $\mathcal{U}$ must span this low-frequency sup-space which spans the lower $N/(N+1)$ band of the spectrum.

**(C3) Singularities preservation.** The prediction of the missing higher frequency band, and the proper reconstruction of the singularities, we discussed in Section 3.1, relies on properly matching patches from the initial upsampled image with ones in a smoothed version of the input. In order to obtain accurate matches, the singularities in the smoothed image $L_0 = \mathcal{U}(\mathcal{D}(I_0))$ and the initial upsampled image $L_1 = \mathcal{U}(I_0)$ must have a similar shape. This, in fact, poses a condition over the downsampling operator $\mathcal{D}$ rather than the upsampling operator $\mathcal{U}$, since both $L_0$ and $L_1$ are constructed by $\mathcal{U}$, meaning that any difference between the two cannot be attributed to $\mathcal{U}$. In practice, this condition is met if the downsampling operator $\mathcal{D}$ preserves the shape of edge-like singularities, appearing in $I_0$, when producing $I_{-1}$.

**(C4) Consistent and optimal reproduction.** Some of the existing approaches [Tappen et al. 2004; Fattal 2007; Shan et al. 2008; Glasner et al. 2009] require that the final upsampled image should be consistent with the input in the sense that if it is reduced back to the input resolution, it must be identical to the input image. We argue that this should also be the case at an earlier stage; the initial upsampled image $L_1$ must already be consistent with the input, i.e., $\mathcal{D}(L_1) = I_0$ as it contains the same information, held in a denser grid. However, since $L_1 = \mathcal{U}(I_0)$, this condition boils down to obeying the biorthogonality condition (4).

Achieving this property implies that the prediction step is not required to undo the loss (or weakening) of data that would otherwise occur. The importance of this optimal use of the original pixels is demonstrated in Figure 9. The existing methods, mentioned above, enforce this relation over the output image $I_1$ implicitly, by solving $\mathcal{D}(I_1) = I_0$ either using a linear solver or through a nonlinear iterative scheme. By designing our filter banks such that they are nearly biorthogonal, we approximate this condition over $L_1$ through an *explicit* and hence efficient computation. As explained in the previous section, the effect of the high-frequency prediction step inserts an error of about 1% in the consistency between $I_1$ and $I_0$. In addition, the inexact biorthogonality mentioned here adds an additional error of about 1%. Altogether, our testings show that the total deviation of $\mathcal{D}(I_1)$ from $I_0$ is of about 2% in pixel intensity value which is visually unnoticeable.

In the Appendix we explain how we use these conditions to define new non-dyadic filter banks. This requires addressing two main difficulties. The first arises from the fact that we need to define compact filters, consisting of a small number of non-zero coefficients,

in order to achieve 'simple' low- and high-frequency example images. Filters with large support suffer from interferences in their response to different nearby edges. This undermines the separation between the singularities which is essential for their scale invariance. On the other hand, a small number of degrees of freedom does not allow us to fully obey all the design conditions we listed above. We treat this over-determinedness by relaxing some of the requirements to objectives in an optimization problem rather than treating them as hard constraints. The second difficulty stems from the non-linearity of the biorthogonality condition (4). These equations make the filter design hard to solve or optimize for all the unknown filter coefficients at the same time. We overcome this by first computing the downsampling filters $d_j$ with no regard to the biorthogonality condition (4), and then compute $u_i$ given $d_j$ while taking into account the biorthogonality.

In Table 1, at the Appendix, we provide the optimal filters we obtained using this framework and used to produce *all* our results.

## 4.    RESULTS

We implemented our method in C++ and run it on an Intel Core 2 Quad CPU 2.8GHz machine. We perform the upscaling using progressively larger scaling ratios for the following reason. As the image size increases nearby singularities become more isolated, allowing us to use stronger smoothing without undermining the scale invariance property of image singularities, discussed in Section 3.1. With this consideration in mind, we approximate the desired magnification factor using the small factors that our filters support, namely, $5{:}4, 4{:}3$, and $3{:}2$. For example, to achieve a magnification of 3 we use $5{:}4, 5{:}4, 4{:}3$, and $3{:}2$, and for a factor of 4 we use $5{:}4, 5{:}4, 4{:}3, 4{:}3$, and $3{:}2$. We then use a simple bicubic downscaling to match the desired magnification factor accurately. We implemented the scheme in the YCbCr color space, where we add high frequencies only to the Y channel. This enables a speedup of about a factor of three compared to using RGB color space. We did not observe a decrease in quality for the scaling factors we tested.

It takes us 4 seconds to upscale an image of 200-by-200 pixels by a factor of 4 along each axis when we run on a single core. Our method is trivially parallel because the different regions in the image are uncoupled in our explicit computation, implying that the running time can be further divided by the number of cores. We exploited this parallel nature of our algorithm and implemented it also on an Nvidia™ Geforce™ 480GTX GPU, using Cuda 3.0, with no changes in the algorithm. This implementation allows us to upscale videos form 640x360 to 1920x1080 (a factor of 3) at 23.9 FPS (including display time). We used scaling steps of 4:3, 3:2, and 3:2.

In Figure 11 we compare the quality of our upscaled images with the results produced by current state-of-the-art methods as well as a leading commercial product Genuine Fractals™. The method of Glasner et al. [2009] reconstructs edges that are somewhat sharper than what our method does, but also produces a small amount of ringing and jaggies. Our method performs considerably less computations when searching patches and avoids solving large linear systems and hence, even in a single core implementation, it runs more than an order of magnitude faster. Despite being examples-based method, our running times are also lower than the times reported for methods that use analytical models [Fattal 2007; Shan et al. 2008; Sun et al. 2008] on the same or stronger machines.



**Fig. 10:** *Failure at cluttered fine-detailed areas, as appears in the the koala's fur. This region does not obey the self-similarity assumption and our method produces false line-like edges. Image courtesy of [Glasner et al. 2009]*

**Video upscaling.** We used our upscaling method with no further modification to enlarge the resolution of video sequences. Similarly to the method by Shan et al. [2008], no flickering artifacts are observed between successive frames. In agreement with Shan et al., we found that once the consistency condition **C4** is obeyed and the output video is stable at its low-frequency component, an overall stability is achieved. In the supplemental material we compare our method to their and obtain high quality results. Our running times, in both the CPU and GPU implementations, are lower.

**Limitations.** Our method is capable of efficiently reconstructing realistic looking edges and shows a decreased amount of jaggies and ringings. However, fine-detailed cluttered regions, such as the Koala fur in Figure 10, are not reproduced realistically and appear somewhat faceted. One possible alternative is to carefully 'turn-off' our high-pass prediction and settle for smooth output in such areas. We leave this option as a future work.

## 5.    CONCLUSIONS

We presented a new example-based image upscaling method that performs less nearest-patch computations and uses a custom designed filter banks to synthesize the image explicitly. The faster search is based on a local self-similarity observation that we point out in natural images, where edges and other singularities are locally scale invariant. The tests we performed to measure this invariance show that this assumption holds best for small scaling factors. Comparisons show that the localized search, permitted by the local scale invariance, outperform approximate global searches both in quality and running time. We formulated and designed novel filter banks that allow us to perform such small, non-dyadic, image scalings. These filter banks extend the common dyadic transformations and are designed to model the image upscaling process. With these filters we achieve, using explicit computations, upscaled image which is highly consistent with the input image. Altogether, we propose a fully local high-quality upscaling algorithms that operates in real-time when implemented in a GPU.

Similarly to image deblurring and denoising, we believe that more thorough efforts should be invested in developing the theory and implementation of image upscaling. The non-dyadic filter banks introduced here motivates the construction of such filters for other problems that will benefit finer scaling. In addition, we intend to explore the use of the various elements presented here for other application, e.g., exploiting self-similarities in geometric entities.

## ACKNOWLEDGMENTS

## REFERENCES

ALY, H. AND DUBOIS, E. 2005. Image up-sampling using total-variation regularization with a new observation model. *IEEE Trans. Image Processing 14,* 10, 1647–1659.

BARNSLEY, M. 1988. Fractal modelling of real world images. In *The Science of Fractal Images*, H.-O. Peitgen and D. Saupe, Eds. Springer-Verlag, New York.

BHAT, P., ZITNICK, C. L., SNAVELY, N., AGARWALA, A., AGRAWALA, M., CURLESS, B., COHEN, M., AND KANG, S. B. 2007. Using photographs to enhance videos of a static scene. In *Rendering Techniques 2007*, J. Kautz and S. Pattanaik, Eds. Eurographics, 327–338.

EBRAHIMI, M. AND VRSCAY, E. R. 2007. Solving the inverse problem of image zooming using self-examples. In *Proc. ICIAR 2007*. Lecture Notes in Computer Science, vol. 4633. Springer, 117–130.

FARSIU, S., ROBINSON, M., ELAD, M., AND MILANFAR, P. 2004. Fast and robust multiframe super resolution. *Image Processing, IEEE Transactions on 13,* 10 (Oct.), 1327–1344.

FATTAL, R. 2007. Image upsampling via imposed edge statistics. *ACM Trans. Graph. 26,* 3, 95.

FREEMAN, W. T., JONES, T. R., AND PASZTOR, E. C. 2002. Example-based super-resolution. *IEEE Comput. Graph. Appl. 22,* 2 (March), 56–65.

FREEMAN, W. T., PASZTOR, E. C., AND CARMICHAEL, O. T. 2000. Learning low-level vision. *Int. J. Comput. Vision 40,* 1 (October), 25–47.

GLASNER, D., BAGON, S., AND IRANI, M. 2009. Super-resolution from a single image. In *ICCV09*. 349–356.

LI, X. AND ORCHARD, M. T. 2001. New edge-directed interpolation. *IEEE Trans. Image Processing 10,* 10, 1521–1527.

LIN, Z. AND SHUM, H.-Y. 2004. Fundamental limits of reconstruction-based superresolution algorithms under local translation. *IEEE Trans. Pattern Anal. Mach. Intell. 26,* 1, 83–97.

MALLAT, S. 1999. *A Wavelet Tour of Signal Processing, Second Edition (Wavelet Analysis & Its Applications)*. Academic Press.

POLIDORI, E. AND DUGELAY, J.-L. 1995. Zooming using iterated function systems. In *NATO ASI on image coding and analysis, July 8-17, 1995, Trondheim, Norway*.

POLLOCK, S. AND CASCIO, I. 2007. Non-dyadic wavelet analysis. *Optimisation, Econometric and Financial Analysis*, 167–203.

PRATT, W. K. 2001. *Digital Image Processing: PIKS Inside*. John Wiley & Sons, Inc., New York, NY, USA.

REUSENS, E. 1994. Overlapped adaptive partitioning for image coding based on the theory of iterated functions systems. In *Acoustics, Speech, and Signal Processing, 1994. ICASSP-94., 1994 IEEE International Conference on*. Vol. v. V/569–V/572 vol.5.

ROBERT, M. G.-A., DENARDO, R., TENDA, Y., AND HUANG, T. S. 1997. Resolution enhancement of images using fractal coding. In *in Visual Communications and Image Processing '97*. 1089–1100.

SHAN, Q., LI, Z., JIA, J., AND TANG, C.-K. 2008. Fast image/video upsampling. *ACM Trans. Graph. 27,* 5, 1–7.

SU, D. AND WILLIS, P. 2004. Image interpolation by pixel-level data-dependent triangulation. *Computer Graphics Forum 23,* 2, 189–202.

SUETAKE, N., SAKANO, M., AND UCHINO, E. 2008. Image super-resolution based on local self-similarity. *Journal Optical Review 15,* 1 (January), 26–30.

SUN, J., NING ZHENG, N., TAO, H., AND YEUNG SHUM, H. 2003. Image hallucination with primal sketch priors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 729–736.

SUN, J., XU, Z., AND SHUM, H.-Y. 2008. Image super-resolution using gradient profile prior. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. 1–8.

TAPPEN, M. F., RUSSELL, B. C., AND FREEMAN, W. T. 2004. Efficient graphical models for processing images. In *Proc. CVPR 2004*. 673–680.

THVENAZ, P., BLU, T., AND UNSER, M. 2000. Image interpolation and resampling. In *Handbook of Medical Imaging, Processing and Analysis*, I. Bankman, Ed. Academic Press, San Diego CA, USA, 393–420.

VRSCAY, E. R. 2002. *From Fractal Image Compression to Fractal-Based Methods in Mathematics*. The IMA Volumes in Mathematics and Its Applications. Springer-Verlag, New York.

XIONG, R., XU, J., AND WU, F. 2006. A lifting-based wavelet transform supporting non-dyadic spatial scalability. In *Image Processing, 2006 IEEE International Conference on*. 1861–1864.

## Appendix

As described in Section 3 we use two operators: the upscaling operator $\mathcal{U}$, which is used to create the initial upscaled image, and the downscaling operator $\mathcal{D}$, which together with the upscaling operator is used to smooth the input image. Each of these operators is defined by its own set of filters. Here we describe how we design these filters such that the principles **C1-4**, discussed in Section 3.3, are approximately obeyed. This design process consists of deriving different terms that promote each of the condition in a way which is easy to solve for, i.e., we restrict to quadratic terms which can be solved linearly once Lagrange rule is applied.

As we explained in Section 3.3, the support of these filters should be kept as compact as possible, in order to avoid mixing the response of the filters to different image features. The design principles pose more conditions than the number of non-zero coefficients we are willing to allocate to the filters. Therefore, we cannot fully obey these conditions and relax some of the requirements to objectives in an optimization problem rather than treating them as hard constraints. In addition, the biorthogonality condition **C4** creates a non-linear relation between the filters of $\mathcal{U}$ and $\mathcal{D}$ and makes their computation a difficult problem. We overcome this non-linearity by splitting the filter design into two linear sub-problems as follows. The downscaling operator $\mathcal{D}$ models the physical image acquisition process and can therefore be computed first, independently of $\mathcal{U}$. The upsampling filters are computed in a second step and ensure the biorthogonality as well as other properties.

In order to model the image acquisition process correctly the downscaling operator $\mathcal{D}$ should perform spatially-uniform scaling. As

we explained in **C1** this can be required over linear functions; the downscaling operator should map linear functions, defined over $\mathcal{G}_{l+1}$, to linear functions, defined over $\mathcal{G}_l$. Therefore, we define $L(x') = x'/a$, where $x' \in \mathcal{G}_{l+1}$ and $a$ is the scaling factor $(N+1)/N$, and require

$$\mathcal{D}\left(L(x')\right)(x) = \mathcal{D}\left(x'/a\right)(x) = x. \tag{5}$$

In addition to this requirement, we want $\mathcal{D}$ to obey the singularities preservation principle **C3**, by making sure that a function $f$, that models the singularities, is mapped correctly between the two scales. We implement this by minimizing the distance between the downscaled $f$ and an analytically scaled $f$ as follows

$$O_1^D = \frac{1}{M} \sum_{\mu, \sigma, x} \left( \mathcal{D}\left(f\left(\frac{x'-\mu}{\sigma}\right)\right)(x) - f\left(\frac{ax-\mu}{\sigma}\right) \right)^2. \tag{6}$$

We use a Gaussian, $f(x) = e^{-x^2}$, to model singularities, shift it by sub-pixel offsets $\mu$ and stretch it $\sigma$ to account for all the offsets and scales the data may come in. $M$ is a normalization factor equal to the number of summed terms. Furthermore, we want $\mathcal{D}$ to span low frequencies, according to **C2**, and define the following objective

$$O_2^D = \frac{1}{N} \sum_{j=1}^{N} \|Dd_j\|^2, \tag{7}$$

where D is the discrete differentiation operator. This is equivalent to minimizing the power spectrum of $\|\hat{\mathcal{D}}(\omega)\|^2/\omega^2$ in Fourier domain, which are the eigenvalues of the Laplacian matrix. Altogether, we get the following constrained optimization problem,

$$\min. O_1^D + \alpha^D O_2^D \text{ s.t., } \mathcal{D}\left(x'/a\right)(x) = x, \tag{8}$$

where $\alpha^D$ is used to prioritize the two objectives. By applying Lagrange multipliers rule, we obtain a small linear system of equations which we solve to get $d_j$.

Given the computed the downscaling filters $d_j$, we can now compute the upsampling filters $u_i$ in a similar way. The main difference here, is that we optimize for biorthogonality condition **C4** and omit the singularities preservation objective term. The biorthogonality condition is achieved by relaxing (3) to the following objective

$$O_1^U = \frac{1}{M'} \sum_{k} \sum_{i,j=1}^{N} \left( \langle u_i[n], d_j[n-(N+1)k] \rangle - \delta_k \cdot \delta_{i-j} \right)^2, \tag{9}$$

where $k$ spans the indices where adjacent $d_j$ overlap $u_i$, and $M'$ is a normalization constant equal to the number of terms summed. This term promotes a dot product of one between corresponding downscaling and upscaling filters and zero otherwise.

As in the computation of $d_j$, we want $u_i$ to span low frequencies, according to **C2**, and achieve this by a similar term,

$$O_2^U = \frac{1}{N} \sum_{i=1}^{N} \|Du_i\|^2. \tag{10}$$

We optimize both objectives subject to the uniform scaling constraint **C1** which is, in this case,

$$\mathcal{U}\left(L'(x')\right)(x) = \mathcal{U}\left(ax'\right)(x) = x, \tag{11}$$

where $L'(x') = ax'$. The resulting optimization problem for $u_j$ is

$$\min. O_1^U + \alpha^U O_2^U \text{ s.t., } \mathcal{U}\left(ax'\right)(x) = x, \tag{12}$$

**Table I. :** *Filter bank coefficients*

| ratio | filter | -3 | -2 | -1 | 0 | 1 | 2 | 3 |
|---|---|---|---|---|---|---|---|---|
| 5:4 | $d_0$ | -0.013 | -0.017 | 0.074 | 0.804 | 0.185 | -0.045 | 0.011 |
| | $d_1$ | -0.005 | 0.032 | -0.129 | 0.753 | 0.421 | -0.09 | 0.017 |
| | $d_2$ | 0.017 | -0.09 | 0.421 | 0.753 | -0.129 | 0.032 | -0.005 |
| | $d_3$ | 0.011 | -0.045 | 0.185 | 0.804 | 0.074 | -0.017 | -0.013 |
| | $u_0$ | -0.028 | -0.053 | 0.061 | 0.925 | 0.304 | 0.007 | 0.014 |
| | $u_1$ | 0 | 0.038 | -0.086 | 0.862 | 0.52 | -0.128 | 0.062 |
| | $u_2$ | 0.062 | -0.128 | 0.52 | 0.862 | -0.086 | 0.038 | 0 |
| | $u_3$ | 0.014 | 0.007 | 0.304 | 0.925 | 0.061 | -0.053 | -0.028 |
| 4:3 | $d_0$ | -0.015 | -0.016 | 0.073 | 0.772 | 0.219 | -0.044 | 0.01 |
| | $d_1$ | 0 | 0.014 | -0.093 | 0.578 | 0.578 | -0.093 | 0.015 |
| | $d_2$ | 0.01 | -0.044 | 0.219 | 0.772 | 0.073 | -0.016 | -0.015 |
| | $u_0$ | -0.042 | -0.041 | 0.069 | 0.928 | 0.365 | -0.008 | 0.03 |
| | $u_1$ | 0.029 | -0.056 | 0.726 | 0.726 | -0.056 | 0.0293 | 0 |
| | $u_2$ | 0.03 | -0.008 | 0.365 | 0.928 | 0.069 | -0.041 | -0.042 |
| 3:2 | $d_0$ | 0 | 0 | -0.022 | 0.974 | 0.227 | 0 | 0 |
| | $d_1$ | 0 | 0 | 0.227 | 0.974 | -0.022 | 0 | 0 |
| | $u_0$ | 0 | -0.1 | 0.119 | 0.927 | 0.6 | -0.047 | 0 |
| | $u_1$ | 0 | -0.047 | 0.6 | 0.927 | 0.119 | -0.1 | 0 |

where $\alpha^U$ is used to prioritize the two objectives.

In Table I we provide the filters we constructed using these calculations. Note, that the filters used in the convolutions are the inverse versions of these filters (i.e. $u_i[-n]$, and $d_i[-n]$)

**Fig. 11:** *A comparison between state-of-the-art methods and our. Images courtesy of [Glasner et al. 2009] and [Sun et al. 2008]*