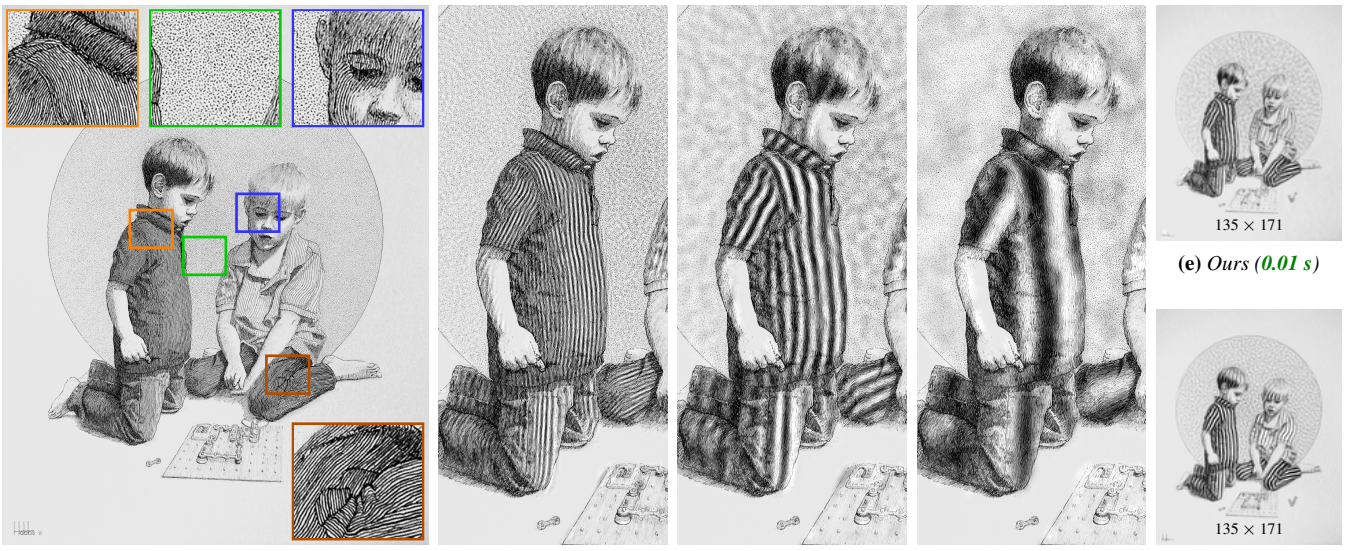
DOI: 10.1111/cgf.142612
EUROGRAPHICS 2021 / N. Mitra and I. Viola
(Guest Editors)

Volume 40 (2021), Number 2

Real-Time Frequency Adjustment of Images and Videos

Rafael L. Germano[†] Manuel M. Oliveira[†] Eduardo S. L. Gastal[†]

Instituto de Informática – UFRGS



(a) *Reference* (1,618 × 2,048) **(b)** *Ours* ($\alpha = 0.3$) **(c)** *Ours* ($\alpha = 0.1$) **(d)** *Ours* ($\alpha = 0.03$) **(e)** *Ours* (***0.01 s***) **(f)** *SR* (***1.5 s***)

Figure 1: *Hand drawing "Brothers", by artist Tyler Hobbs (used with permission). (a) Reference image with* 1,618 × 2,048 *pixels. (b–d) Examples of frequency adjustments applied to (a) using our technique, with the images kept at the original resolution. (e) Downscaled version of (a) generated with our technique in real time (0.01 seconds for frequency adjustment and image reconstruction). (f) Downscaled version of (a) obtained with spectral remapping (SR) [GO17] (1.5 seconds). In both (e) and (f), resampling to* 135 × 171 *pixels was done using the method of [NH14]. Please refer to the supplemental video to see our technique working in real time in our interactive demo.*

Abstract

We present a technique for real-time adjustment of spatial frequencies in images and videos. Our method allows for both decreasing and increasing of frequencies, and is orthogonal to image resizing. Thus, it can be used to automatically adjust spatial frequencies to preserve the appearance of structured patterns during image downscaling and upscaling. By pre-computing the image's space-frequency decomposition and its unwrapped phases, these operations can be performed in real time, thanks to our novel mathematical perspective on frequency manipulation of digital images: interpreting the problem through the theory of instantaneous frequencies and phase unwrapping. To make this possible, we introduce an algorithm for the simultaneous phase unwrapping of several unordered frequency components, which also deals with the frequency-sign ambiguity of real signals. As such, our method provides theoretical and practical improvements to the concept of spectral remapping, enabling real-time performance and improved color handling. We demonstrate its effectiveness on a large number of images subject to frequency adjustment. By providing real-time control over the spatial frequencies associated with structured patterns, our technique expands the range of creative and technical possibilities for image and video processing.

CCS Concepts

• ***Computing methodologies*** → ***Image processing;***

1. Introduction

The frequency domain is a common way of analyzing digital signals. It is used in digital signal processing as a tool for filtering, compression, and several other applications [OS10]. For some applications, however, *time*-frequency analysis is an alternative and preferred tool, since it allows for the study of transient phenomena. In image and video processing, the analogous *space*-frequency domain has also been extensively used for example with wavelets for image compression, data analysis and denoising [PI97, BH13]. Space-frequency analysis may be preferred to traditional frequency analysis because the signal becomes represented by waves concentrated both in space and frequency [Mal09]. This is useful because the frequency content becomes more meaningful in each associated region.

We present a new mathematical perspective on frequency manipulation of digital images and videos. Using the theory of instantaneous frequencies and phase unwrapping operators [Mal09], our **real-time spatial frequency adjustment technique** is the first method capable of performing image edits such as the ones shown in Figure 1 in real time. The performance improvement resulting from our formulation is also critical for making frequency adjustments in videos practical. The idea of our method is to decompose an image into local phase and amplitude components such that the phase encodes the high-frequency structured content. This decomposition provides simple and interesting ways of manipulating frequency through phase manipulation.

Our technique has two steps: first, it estimates, for each pixel p, the phase values $\theta^1(p)$, $\theta^2(p)$, ... associated with several cosine functions $\cos(\theta^n(p))$, that when combined, locally define the image's structured high-frequency patterns (Section 3.2). In the second step, the obtained phase values are used to perform per-pixel adjustment of spatial frequencies and image reconstruction in real time. This is achieved by accumulating $\cos(\alpha\theta^n(p))$ for some frequency scaling factor $\alpha > 0$ over all $\theta^n(p)$ (Section 5). Note that this preserves the monotonic relationship among the adjusted frequencies. Despite working with cosine functions for the decomposition, our technique is able to handle complex structured patterns as demonstrated throughout the paper (e.g., see Figure 8).

Our novel mathematical framework is fundamental in allowing our approach to work in real time (Figure 2). To make this possible, we introduce technical contributions to ridge analysis and 2-D/3-D phase unwrapping of digital images and videos. As far as we know, we are the first to describe an algorithm for simultaneous unwrapping of several unordered components (Section 4.2). This is essential for handling complex patterns such as the sunflower's disk in Figure 8, or the lizard's leg in Figure 11. We also show how to handle the problematic sign-ambiguity that arises from the conjugate-symmetry of real waves (Section 4.1.1). Our approach also improves color handling when compared to recent frequency remapping methods [GO17], by jointly unwrapping the color channels' phase functions (Section 4.3). This improvement is visible in Figure 5 and on the shirt example between 4min20s and 5min of our supplemental video.

† rlgermano@inf.ufrgs.br, oliveira@inf.ufrgs.br, eslgastal@inf.ufrgs.br

By defining the frequency scaling factor α as a function of an image resizing factor r, spatial frequencies are automatically adjusted to preserve the appearance of structured patterns during image downscaling (Section 5.2.3, Figure 1(e)). Combined with our method's real-time performance, this opens up a variety of interesting applications, such as dynamically computing detail-preserving thumbnails for image galleries, or adapting an image's content in real time based on viewing distance. We also demonstrate the use of our technique for frequency upscaling (Figure 12), and how it can be combined with existing super-resolution techniques to obtain novel super-resolution results (Figure 10).

In summary, the **contributions** of our work include:

- *The first real-time technique for spatial frequency adjustment of images and videos* (Section 3). It is based on a novel mathematical framework that enables its improved performance over the state of the art.
- *A phase-unwrapping technique to recover the phases of real waves* that locally reconstruct an image (Section 4). Our solution naturally handles a sign ambiguity involving the phase unwrapping of real waves, in addition to simultaneously treating many unordered phase components without explicit sorting. As such, it extends the state-of-the-art on phase unwrapping that only covers a single phase component wrapped to $[-\pi, \pi)$.
- *A technique for processing color images for use with spatial frequency adjustment* (Section 4.3). It is based on a multi-channel optimization performed in RGB space and avoids some color artifacts associated with the use of PCA-based color reconstruction.

2. Related Work

Our work was inspired by the *spectral remapping* (SR) technique for image downscaling [GO17]. SR remaps frequencies that are not representable in the target resolution to new representable frequencies, preserving the orientations of the original high-frequency patterns. This solution effectively encodes important structured details at lower frequencies during downscaling, instead of discarding them with an anti-aliasing filter. This results in more faithful representations of the original content. The technique, however, has only been demonstrated for image downscaling in combination with a remapping function that takes similarly-oriented waves to a common frequency. More importantly, *(i)* it is *non-interactive*, requiring the solution of sparse linear system for each combination of input image I and downscaling factor r; and *(ii)* the PCA-based reconstruction strategy used for processing color images may introduce some noticeable artifacts depending on the target resolution.

Our frequency-adjustment technique can be employed to image downscaling and addresses the aforementioned issues: it supports both decrease and increase of spatial frequencies, can be used with any image resizing factor $r > 0$, preserves the monotonic relation among the remapped frequencies, and can be used in interactive applications, being up to $100\times$ faster than SR. In addition, we also demonstrate its extension to videos.

Although our method can generate its results directly at the target resolution, it complements and can be used in combination with any resampling strategy, including downscaling and upscaling [KSP13, NH14, OG15, SN15, WWA*16]. Our

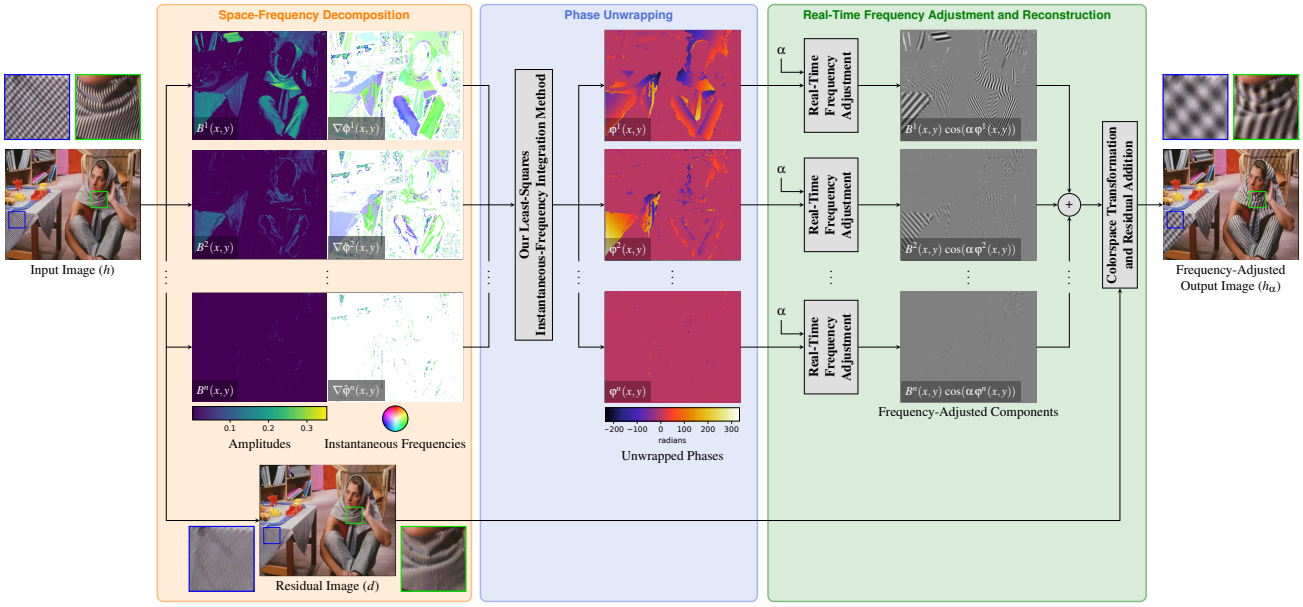


Figure 2: *Overview of our frequency-adjustment technique.* (Orange block) *The input image is decomposed into several cosine waves (Eq. (4)), represented by per-pixel scalar amplitudes and instantaneous-frequency vectors. The frequency vector direction and magnitude are given by the vector from the center of the color wheel to the pixel color in the wheel. This decomposition may be performed in the RGB space (applied to each color channel independently) or in some other color space where channels better encode the high-frequency content, such as the PCA-defined color space used for this illustration.* (Blue block) *The detected frequencies are then integrated by our method to recover a continuous scalar map of unwrapped phases. Note that the unwrapped phase values for each pixel are not sorted between the images, and as such the phase data may appear more discontinuous than they actually are. Note also the absence of phase information for low-detail regions, which are treated as residue only.* (Green block) *The frequency-adjusted content is obtained by reconstructing the waves from their original amplitudes and corresponding scaled unwrapped phases (Eq. (19)), using a user-provided scaling factor* α *(for this example,* $\alpha = 0.25$*). All waves are summed to obtain the frequency-adjusted content. Finally, to obtain the reconstructed image, the frequency-adjusted content is transformed back to RGB space (if initially in RGB space no transformation is needed) and summed back to the residual.*

technique can also be used with any single-image super-resolution method (e.g., [LSK*17, FF10, YWHM10]), and we demonstrate a hybrid solution combining super-resolution and spatial frequency adjustment in Section 6.

To achieve our results, we introduce a *phase unwrapping* algorithm that simultaneously handles several real-wave phase components, in addition to inter-component dependencies (Section 4). Phase unwrapping is the process of recovering a continuous function $\theta(x)$ from samples $\hat{\theta}(x)$ whose values are only known up to modulo 2π [OS10]. This procedure is a fundamental building block in many applications, ranging from satellite topography and marine/earth seismology to the analysis of brain waves [CSK77, GZW88]. Phase unwrapping is tightly linked to the theory of instantaneous-frequency decompositions and space/time-frequency analysis [Mal09]. In this context, it is often called "ridge" analysis, and it may be derived from Gabor [DEG*92] and wavelet transforms [LO10]. There is a significant amount of theory and algorithms developed in this area for 1-D signals, such as acoustic and speech data [GKM96]. The corresponding 2-D/3-D problems (required for our frequency-adjustment technique) is significantly more challenging to solve [GP98]. Appendices B and C include a more detailed discussion of this topic.

To the best of our knowledge we are the first to describe a practical algorithm for dealing with the specific issues that arise during ridge analysis and 2-D/3-D phase unwrapping of digital images and videos. We review the phase-unwrapping problem in Section 4.

3. Frequency Adjustment

Consider a continuous wave signal $s(x,y) = \cos(\theta(x,y))$ described by a continuously-differentiable phase function θ. The horizontal $\nabla_x\theta(x,y)$ and vertical $\nabla_y\theta(x,y)$ *instantaneous frequencies* of this signal are given by the partial derivatives of its phase [Mal09]:

$$\nabla_x\theta(x,y) = \frac{\partial}{\partial x}\theta(x,y) \quad \text{and} \quad \nabla_y\theta(x,y) = \frac{\partial}{\partial y}\theta(x,y). \qquad (1)$$

Similarly, the local wavefront orientation (normal direction) at (x,y) is given by the gradient vector $\nabla\theta(x,y) = (\nabla_x\theta(x,y), \nabla_y\theta(x,y))$.

A *frequency-adjusted* version s_α of s, where frequencies are scaled by a factor $\alpha > 0$, is obtained as:

$$s_\alpha(x,y) = \cos(\alpha\theta(x,y)). \qquad (2)$$

It is easy to see that the horizontal and vertical frequencies of s_α are $\alpha\nabla_x\theta$ and $\alpha\nabla_y\theta$, i.e., scaled versions of the corresponding frequencies of s. The local direction of wave propagation is preserved in s_α since $\nabla\alpha\theta = \alpha\nabla\theta$. Finally, s_α is still a continuous function, as

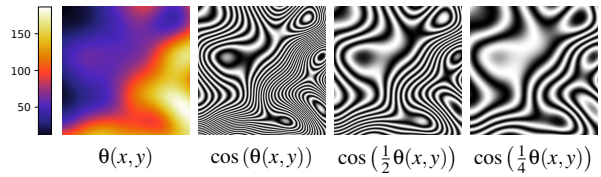


Figure 3: *Example of frequency adjustment. The continuous phase function $\theta(x,y)$ is defined by a Perlin noise scalar field (left). Scaling the phase by factors $\alpha > 0$ results in frequency-adjusted signals $\cos(\alpha\theta(x,y))$, which preserve the local wavefront orientations.*

was s. Figure 3 illustrates an example of frequency adjustment for a signal where θ is known analytically.

3.1. Frequency Adjustment of Natural Images

Given a natural image $h(x,y)$, we seek a decomposition which represents it as a sum of N components with phase functions $\theta^1 \dots \theta^N$, amplitude functions $A^1 \dots A^N$, and a residual $d(x,y)$:

$$h = s + d \quad \text{where} \quad s(x,y) = \sum_{n=1}^{N} A^n(x,y)\cos(\theta^n(x,y)). \tag{3}$$

Obtaining such a decomposition for a given image h is a severely under-constrained problem. Thus, we require the summation in $s(x,y)$ to only encode high-frequency content that should be affected by the frequency adjustment (remaining content goes into the residual d). Furthermore, the phase functions are required to be piecewise smooth for the instantaneous frequencies to be well defined almost everywhere, making $s(x,y)$ (and thus h) amenable to frequency adjustment. Finally, the amplitudes $A^n(x,y)$ should vary slowly when compared to local frequencies, so that high-frequency variations become encoded in the phases only [Mal09].

The decomposition shown in Eq. (3) is related to the Hilbert-Huang transform (HHT), that separates a signal into a collection of so-called Intrinsic Mode Functions (IMFs) and a residue [HSL*98, Hua14]. The HHT differs from our decomposition by the way the phases are computed: while we use windowed Fourier transforms (Gabor analysis) to obtain instantenous frequencies [GO17] (which are integrated to generated the unwrapped phase), HHT uses the Hilbert transform. Both decompositions however require unwrapping to obtain a continuous phase, which we discuss in the following sections.

For a frequency adjustment factor $\alpha > 0$, the frequency-adjusted image h_α is obtained by using the scaled phase functions $\alpha\theta^n$ during reconstruction, as in Eq. (2).

3.2. Finding a Decomposition

Time/Space-frequency analysis may be employed to measure the instantanous frequencies of several non-harmonic spectral components [Mal09]. Gastal and Oliveira [GO17] describe a practical algorithm for this purpose, approximating the neighborhood around each pixel as a sum of Gaussian-windowed plane waves. From the constant frequencies and phase-shifts of such waves it is possible to extract (Appendix A), for each pixel, a series of phase values $\hat{\varphi}^n(x,y)$ and associated amplitudes $B^n(x,y)$, $n = 1 \dots N$, such that

$$h = s + d \quad \text{where} \quad s(x,y) = \sum_{n=1}^{N} B^n(x,y)\cos(\hat{\varphi}^n(x,y)). \tag{4}$$

Despite the similarity of this expression to that of Eq. (3), the values $\hat{\varphi}^n$ cannot be directly employed for frequency adjustment since they are *wrapped* to $[0,\pi]$. The problem that occurs when trying to do so is illustrated in Figure 4.

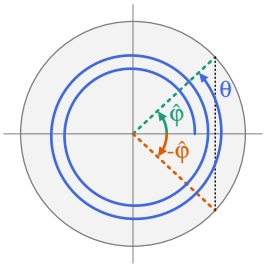
The occurrence of wrapped values is an inherent mathematical limitation of phase measurements. Since $\cos(\theta) = \cos(\theta + 2\pi m)$ for $m \in \mathbb{Z}$, the phase θ is not uniquely defined. Moreover, when trying to recover θ as $\arccos(\cos(\theta))$, one obtains the principal value $\hat{\varphi} \in [0,\pi]$. Recovering a continuous and differentiable phase function $\theta(x,y)$ from wrapped values $\hat{\varphi}(x,y)$ is called *phase unwrapping* [OS10]. The measured phase values $\hat{\varphi}^n$ recovered from space-frequency analysis techniques (including Gastal and Oliveira's) are wrapped, and therefore require unwrapping in order to define phase functions θ^n amenable to frequency adjustment.

4. Phase Unwrapping in Two Dimensions

While 1-D phase unwrapping has a simple solution [Mer79], 2-D unwrapping is more involving and requires specialized algorithms [GP98]. It has been extensively studied due to its relevance to many fields, including interferometry and magnetic resonance imaging, and several algorithms exist aimed at these and other applications (we include a detailed review in Appendix B). *These algorithms work with a wrapped phase* $\hat{\theta} \in [-\pi,\pi)$. As such, $\hat{\theta}$ is related to the true phase θ by the expression

$$\theta(x,y) = \hat{\theta}(x,y) + 2\pi k(x,y), \tag{5}$$

for unknown $k(x,y) \in \mathbb{Z}$. Thus, solving the phase unwrapping problem in this case consists of finding the shift factors $k(x,y)$.



Digital images are real functions and modeled using real waves: the cosines in Eq. (3). Therefore, in this situation the measured phase *has a reduced angular range*, $\hat{\varphi} = \arccos(\cos(\theta)) \in [0,\pi]$, and an *ambiguous sign*, since $\cos(\hat{\varphi}) = \cos(-\hat{\varphi})$. As a result, $\hat{\varphi}$ is related to the true phase θ by an unknown sign, σ, in addition to an unknown shift, k (the diagram to the right illustrates the case $k = 2$, $\sigma = +1$):

$$\theta(x,y) = \sigma(x,y)\,\hat{\varphi}(x,y) + 2\pi k(x,y) \;\text{ for }\; \sigma(x,y) \in \{-1,+1\}. \tag{6}$$

Such *sign ambiguity* and *reduced angular range* make phase unwrapping for digital images and videos a challenging task.

There are other applications that work with digital images (real signals) that may also suffer from sign ambiguity. However, in most situations one can work around this problem since the common underlying frequency transform has predefined frequency bands, such as when using oriented steerable pyramids [WRDF13]. This cannot be done in our situation since frequency adjustment requires the use

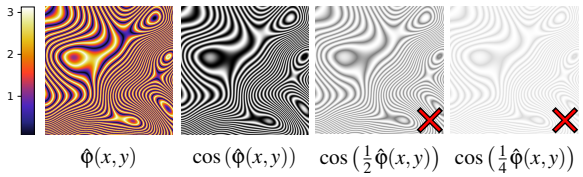


Figure 4: *Example of unsuccessful frequency adjustment. When the phase function $\hat{\varphi}(x,y)$ is wrapped (values recovered as $\hat{\varphi} = \arccos(\cos(\varphi))$ are in $[0,\pi]$ – see color scale on the left), scaling the phase by factors $\alpha > 0$ does **not** result in frequency-adjusted signals. One must therefore **unwrap** the phase function before being able to perform frequency adjustment. Compare to Figure 3.*

of instantaneous frequencies extracted from non-harmonic frequencies. That is, in our case the local frequencies are not predetermined, but are instead detected during the decomposition, and thus vary between locations of the image. In other situations, the sign ambiguity may be resolved through procedures that are equivalent to a simple 1-D unwrapping of a single phase component along a particular dimension [WRDF14]. This also does not work in our case since frequency adjustment must work with 2-D and 3-D unwrapping of several phase components.

In the following sections we present our novel 2-D phase unwrapping algorithm designed for spatial frequency adjustment of digital images (its extension to videos, which requires 3-D processing, is described in Section 5.3). Our solution handles the following issues that arise when dealing with real waves: *(i)* phase measurements with ambiguous sign, due to the conjugate-symmetry of the spectrum of real signals (Section 4.1); *(ii)* presence of several unordered phase components, required for representing complex patterns (Section 4.2); and *(iii)* correlated multichannel components in color images (Section 4.3). Furthermore, implementation details and optimizations are described in Section 5 and Section 5.1.

4.1. Unwrapping the phase of one real wave

For frequency adjustment of digital images, one is interested in unwrapping the phase $\theta(x,y)$ from a real wave component $\cos(\theta(x,y))$. The wrapped phase measurement $\hat{\varphi}(x,y) \in [0,\pi]$ is thus related to the true phase according to Eq. (6). Since $k(x,y)$ defines a piecewise-constant function, the unwrapped and wrapped gradients are related only by the unknown sign $\sigma(x,y)$ at each position (x,y):

$$\nabla\theta(x,y) = \sigma(x,y)\nabla\hat{\varphi}(x,y) \quad \text{almost everywhere.} \tag{7}$$

In Appendix A we show how to obtain $\nabla\hat{\varphi}$ directly from a space-frequency decomposition. We now focus on how to integrate the gradient field $\nabla\theta = \sigma\nabla\hat{\varphi}$ to recover the phase function, *without explictly computing the signs* σ.

(Throughout the text we denote the pointwise product of two functions f and g as fg; that is, $(fg)(x,y) = f(x,y)\,g(x,y)$. Furthermore, $f = g$ denotes pointwise equality: $f(x,y) = g(x,y), \forall (x,y)$.)

4.1.1. Integrating the phase gradient field for one real wave.

To recover the unwrapped phase θ, the gradient field $\nabla\theta = \sigma\nabla\hat{\varphi}$ is integrated in the least-squares sense. Although the signs $\sigma(x,y)$ are unknown, our insight is that it is possible to check whether neighboring pixels experience a change in sign (Section 4.1.2). This information is sufficient for integrating $\nabla\theta$ in a sign-consistent way through some clever algebraic manipulations, shown below.

Let T_x and T_y be two unit-shift operators that act on a function θ as $T_x(\theta)(x,y) = \theta(x-1,y)$, and $T_y(\theta)(x,y) = \theta(x,y-1)$. Furthermore, let $\circ$ in $T_\circ$ stand for either x or y. Then, a trapezoidal integration rule gives the following linear relations between the unknown phase values $\theta(x,y)$ of all pixels [Sol15]:

$$\theta - T_\circ(\theta) = \tfrac{1}{2}\left[\nabla_\circ\theta + T_\circ(\nabla_\circ\theta)\right]. \tag{8}$$

Given that $\nabla\theta = \sigma\nabla\hat{\varphi}$, Eq. (8) may be expanded as (we use underlines to highlight the parts of the equations that have changed):

$$\theta - T_\circ(\theta) = \tfrac{1}{2}\left[\underline{\sigma\nabla_\circ\hat{\varphi}} + T_\circ(\underline{\sigma\nabla_\circ\hat{\varphi}})\right]. \tag{9}$$

By introducing a function φ satisfying $\sigma\varphi \stackrel{\text{def}}{=} \theta$, Eq. (9) becomes

$$\underline{\sigma\varphi - T_\circ(\sigma\varphi)} = \tfrac{1}{2}\left[\sigma\nabla_\circ\hat{\varphi} + T_\circ(\sigma\nabla_\circ\hat{\varphi})\right]. \tag{10}$$

Note that to solve the above equation one must know σ. To remove this dependency, we start by multiplying both sides by σ:

$$\underline{\sigma^2}\varphi - \underline{\sigma}T_\circ(\sigma\varphi) = \tfrac{1}{2}\left[\underline{\sigma^2}\nabla_\circ\hat{\varphi} + \underline{\sigma}T_\circ(\sigma\nabla_\circ\hat{\varphi})\right]. \tag{11}$$

Note that $\sigma^2 = 1$ for any (x,y), since $\sigma(x,y) \in \{-1,+1\}$. Furthermore, since $T(fg) = T(f)T(g)$,[‡] Eq. (11) may be simplified to

$$\varphi - \underline{\Delta_\circ\sigma}\,T_\circ(\varphi) = \tfrac{1}{2}\left[\nabla_\circ\hat{\varphi} + \underline{\Delta_\circ\sigma}\,T_\circ(\nabla_\circ\hat{\varphi})\right], \tag{12}$$

where $\Delta_\circ\sigma \stackrel{\text{def}}{=} \sigma T_\circ(\sigma)$. Note that $\Delta_x\sigma$ *and* $\Delta_y\sigma$ *encode the locations where* σ *changes sign*. More precisely, $\Delta_x\sigma(x,y) = -1$ whenever the sign of $\sigma(x,y)$ differs from the sign of the neighboring pixel $\sigma(x-1,y)$, and $\Delta_x\sigma(x,y) = +1$ otherwise.[§] An analogous relation is true for $\Delta_y\sigma(x,y)$ in the y-dimension.

As shown below in Section 4.1.2, it is possible to compute $\Delta_\circ\sigma$ despite σ being unknown. As such, the only unknowns in Eq. (12) are the values of the function φ, which may be found by solving the associated linear system through least squares:

$$\min_{\varphi} \sum_{\circ\in\{x,y\}} \left\| \varphi - \Delta_\circ\sigma T_\circ(\varphi) - f_\circ \right\|^2 + \lambda\|\varphi\|^2, \tag{13}$$

where $f_\circ(x,y)$ encodes the right-hand-side of Eq. (12), for $\circ \in \{x,y\}$. The recovered phase φ is enough to perform frequency adjustment, without knowing the per-pixel sign σ, since the cosine is an even function. Thus $\cos(\alpha\varphi) = \cos(\alpha\sigma\varphi) = \cos(\alpha\theta)$ and so it is not necessary to recover θ.

The minimum-norm regularization term $\|\varphi\|^2$ in Eq. (13) is required for two reasons. First, without regularization, the existence of one solution φ would imply an infinite number of solutions $\varphi + C$, for any real integration constant C. Second, due to the inherent trade-off between space and frequency locality [Mal09], the numerical space-frequency analysis only detects phase information for

[‡] In 1-D: $T(fg)(x) = (fg)(x-1) = f(x-1)\,g(x-1) = T(f)(x)\,T(g)(x)$.

[§] From the definition of the unit shifts T_x and T_y:
$\Delta_x\sigma(x,y) = \sigma(x,y)T_x(\sigma)(x,y) = \sigma(x,y)\,\sigma(x-1,y)$, and
$\Delta_y\sigma(x,y) = \sigma(x,y)T_y(\sigma)(x,y) = \sigma(x,y)\,\sigma(x,y-1)$.

high-frequency portions of the image [GO17]. As a result, the measurements $\nabla\hat{\varphi}$ and consequently $f_\circ$ have undefined values for some pixels, which may result in unconstrained unknowns. The same occurs along the boundary of the image domain. The regularizer forces any unconstrained variables to zero. Finally, an additional benefit of the regularizer is that it penalizes the occurrence of large absolute values in φ, making the frequency-adjusted reconstruction $\cos(\alpha\varphi)$ less sensitive to small changes in α (the change in $\cos(\alpha\varphi)$ when varying α is given by $\frac{\partial}{\partial\alpha}\cos(\alpha\varphi) = -\varphi\sin(\alpha\varphi) \propto \varphi$). This is important in interactive applications where α is user-controlled.

Eq. (13) has a unique solution obtainable from the linear system

$$(D_x^T D_x + D_y^T D_y + \lambda I)\,\varphi = D_x^T f_x + D_y^T f_y, \tag{14}$$

where I is the identity matrix, and D_x and D_y are similar to backward-difference matrices, but *(i)* have some entries with flipped signs (according to $\Delta_x\sigma$ and $\Delta_y\sigma$); and *(ii)* only include contraints for non-missing values (the reconstruction of pixels with missing values is handled by windowing, discussed in Section 5). We use $\lambda = 10^{-6}$ in the regularizer since it produces good results.

4.1.2. Determining sign changes and computing $\Delta\sigma$. Let p and q be two neighboring pixels, for which we compute local gradients $\nabla\hat{\varphi}(p)$ and $\nabla\hat{\varphi}(q)$ according to Appendix A. Also, recall that $\nabla\theta = \sigma\nabla\hat{\varphi}$. Two situations are possible:

1. **There is** a sign change between the phase gradients $\nabla\theta(p)$ and $\nabla\theta(q)$ of p and q, meaning that $\sigma(p) \neq \sigma(q)$;
2. **There is no** sign change between p and q, and $\sigma(p) = \sigma(q)$.

Possibility (1) implies $\Delta\sigma(p) = \sigma(p)\,\sigma(q) = -1$, while possibility (2) implies $\Delta\sigma(p) = \sigma(p)\,\sigma(q) = +1$. To determine which one is true, we recall that the phase θ should be continuously differentiable, meaning that the phase gradient $\nabla\theta$ should be continuous. As such, we choose the option that better preserves the continuity of $\nabla\theta$, i.e., that results in the smallest change between $\nabla\theta(p)$ and $\nabla\theta(q)$. Thus,

$$\Delta\sigma(p) = \operatorname*{arg\,min}_{v\in\{-1,+1\}} \left\| \nabla\hat{\varphi}(p) - v\,\nabla\hat{\varphi}(q) \right\|^2. \tag{15}$$

4.2. Unwrapping the phases of several real waves

Natural images contain complex structured patterns, such as in the honeycomb and sunflower photographs of Figure 8. These patterns require the summation of several cosine components in order to be correctly represented by the decomposition of Eq. (3). The several phase functions $\{\theta^1, \theta^2, \ldots\}$ that build-up a pattern are in principle independent and could be separately unwrapped. In practice, however, this independent unwrapping is severely error prone. To understand why, note that, before unwrapping, one would be required to assign each n-th gradient measurement $\nabla\hat{\varphi}^n(p)$, of a pixel p, to a particular phase component $\theta^{\kappa_p(n)}$, where $\kappa_p(n)$ is an indexing function (permutation) specific for p. Furthermore, the permutations κ_p and κ_q for all neighboring pixels p and q should be consistent to each other, assigning the gradients that belong together to the same phase component. Otherwise, severe unwrapping errors will occur.

Due to ambiguous information returned by the space-frequency decomposition, sometimes it is not possible to precisely define the permutations κ_p for all pixels p. Furthermore, for particular pixels, the permutations are often incorrectly defined due to noise or frequency measurement errors. As a result, the whole phase unwrapping solution becomes compromised.

We avoid the aforementioned difficulties by, instead, defining weighted associations between *all* measurements $\nabla\hat{\varphi}^n$ and *all* phase functions θ^n (in place of discrete permutations). This significantly reduces the possibility of unwrapping errors, especially in the presence of ambiguous information returned by the space-frequency decomposition. It also makes the algorithm more robust to noise and measurement errors, and handles missing phase information.

Let p and q denote two neighboring pixels (in any direction). Furthermore, let $N(p)$ and $N(q)$ denote the number of measured phase components for p and q, respectively. For valid indices $n = 1\ldots N(p)$ and $m = 1\ldots N(q)$, the inter-component version of the integration rule (Eq. (12)), which couples the n-th component of pixel p to the m-th component of pixel q, is given by

$$\varphi^n - \Delta_\circ^{nm}\sigma\, T_\circ(\varphi^m) = \tfrac{1}{2}\left[\nabla_\circ\hat{\varphi}^n + \Delta_\circ^{nm}\sigma\, T_\circ(\nabla_\circ\hat{\varphi}^m)\right]. \tag{16}$$

As before, $\Delta^{nm}\sigma(p)$ determines if there is an expected change in sign between the n-th gradient measurement $\nabla\hat{\varphi}^n(p)$ of p, and the m-th measurement $\nabla\hat{\varphi}^m(q)$ of q.

Some components are more likely to belong together than others. Thus, let the weight μ_{pq}^{nm} (defined below) be proportional to the probability of $\nabla\hat{\varphi}^n(p)$ and $\nabla\hat{\varphi}^m(q)$ belonging to the same phase component θ^k. We transform Eq. (16) into a *weighted* least-squares system, that simultaneously searches for the unwrapped phase components φ^n that best adapt to *all* couplings between measurements, but giving greater weight to the most probable ones (through μ^{nm}):

$$\min_{\varphi^n} \sum_{\circ\in\{x,y\}} \sum_{\forall n}\sum_{\forall m} \left\| \mu_\circ^{nm}\left\{L_\circ^{nm} - f_\circ^{nm}\right\}\right\|^2 + \lambda\sum_{\forall n}\|\varphi^n\|^2. \tag{17}$$

Here, $L_\circ^{nm}$ is the left-hand-side of Eq. (16), and $f_\circ^{nm}$ is its right-hand side. Furthermore, it is understood that n and m only vary over the valid indices for each pixel. For μ_{pq}^{nm}, we use the wave alignment measure proposed by Gastal and Oliveira [GO17] (which determines how well the local Gabor approximations, associated with $\nabla\hat{\varphi}^n(p)$ and $\nabla\hat{\varphi}^m(q)$, match around the midpoint between p and q), scaled by a frequency-vector orientation similarity measure (Eq. (31)).

4.3. Processing Color Images

Frequency adjustment should be performed in all channels of a color image, since they have correlated information [RAGS01]. Thus, while each color channel has its own phase decomposition (Eq. (3)), their phase functions are unwrapped simultaneously, in a single optimization step. Let $\mathcal{E}_c$ be the energy functional from Eq. (17), written for the phase components φ_c^n of a specific color channel $c \in \{R, G, B\}$. We minimize the global energy

$$(\mathcal{E}_R + \mathcal{E}_G + \mathcal{E}_B) + \beta\,(\mathcal{X}_{RG} + \mathcal{X}_{GB} + \mathcal{X}_{RB}), \tag{18}$$

where $\mathcal{X}_{c_1c_2}$ are cross-channel phase constraints, between two channels c_1 and c_2, and $\beta = 10^{-3}$. These new constraints seek similar phase solutions $\varphi_{c_1}^n(p)$ and $\varphi_{c_2}^m(p)$, for a pixel p, if the measured instantaneous frequencies $\nabla\hat{\varphi}_{c_1}^n(p)$ and $\nabla\hat{\varphi}_{c_2}^m(p)$ are similar, between channels c_1 and c_2. The indices $n = 1\ldots N_{c_1}(p)$ and $m = 1\ldots N_{c_2}(p)$ range over all detected phase components for pixel p at, respectively, color channels c_1 and c_2. The formulas are given in Appendix D.

(a) *Original photograph* **(b)** *Our multi-channel frequency adjustment* **(c)** *Spectral remapping [GO17]*

Figure 5: *Our multi-channel phase unwrapping leads to a frequency-adjusted image, in (b), that is faithful to the colors of the original photograph, in (a). The remapping algorithm of Gastal and Oliveira [GO17], in (c), reduces frequencies but is not able correctly preserve colors. Adjustment of* $\alpha = 0.2$.

As shown in Figure 5, when compared to the color-processing algorithm of Gastal and Oliveira [GO17], our multi-channel optimization leads to fewer color distortions when used for spatial frequency adjustments. This occurs because their so-called wave-alignment step is limited to processing a single color channel (either the lightness channel or one PCA-decorrelated channel). Our approach, on the other hand, allows for a joint optimization that preserves inter-channel correlations, resulting in improved frequency adjustment results. As such, we recommend the use of our multi-channel optimization in RGB space (Eqs. (17), (18) and (29)) when processing images which contain cross-channel frequency correlations.

For processing grayscale images one should use Eq. (17) directly. Similarly, for color images which contain most of their high-frequency contrast variations in the lightness channel, the combination of our phase unwrapping technique with the PCA-guided processing of Gastal and Oliveira [GO17] leads to the best results. In practice, this means extracting and unwrapping the phase of only one "channel," represented by the principal direction of color variation in RGB space. This procedure was used for generating the results shown in Figure 10, where the regions with detected high frequencies are mostly monochromatic (bluish gray and dark blue, respectively).

5. Implementation Details

Eq. (17) or (18) is solved once in a pre-processing step and the phase values $\varphi^n(p)$ are stored for subsequent real-time frequency adjustment. The resulting linear system is sparse and positive definite, which we solve using a QR factorization from SuiteSparse [Dav11]. We have also experimented with an iterative conjugate-gradient solver, but it was slower than the direct solver and stopping the iterations before convergence leads to visual artifacts during frequency adjustment (misalignment between structured patterns due to incomplete phase unwrapping). Frequency adjustment by a factor $\alpha > 0$ is obtained simply by replacing θ^n with $\alpha\varphi^n$ in Eq. (3). The resulting frequency-adjusted image is then reconstructed as

$$h_\alpha = s_\alpha + d, \quad \text{where} \quad s_\alpha(p) = \sum_{n=1}^{N(p)} B^n(p)\cos(\alpha\varphi^n(p)). \quad (19)$$

In practice, however, it is best to compute $s_\alpha(p)$ using a neighborhood Ω_p around p. This is achieved through the use of a weighted reconstruction window w, centered at p and covering Ω_p. This guarantees a smooth reconstruction for s_α, even in the presence of noisy measurements or missing phase values. The corresponding reconstruction equation, with windowing, becomes

$$s_\alpha(p) = \sum_{q\in\Omega_p}\sum_{n=1}^{N(q)} w(p-q)\,B^n(q)\cos\left(\alpha\{\varphi^n(q) + \nabla\hat{\varphi}^n(q)\cdot(p-q)\}\right). \quad (20)$$

The expression $\varphi^n(q) + \nabla\hat{\varphi}^n(q)\cdot(p-q)$ is a first order (linear) extrapolation of the phase $\varphi^n(q)$ from q to p. The appropriate reconstruction window $w(x,y)$ depends on the parameters of the Gabor decomposition being employed for space-frequency analysis, as discussed in detail by Gastal and Oliveira [GO17]. In Eq. (20), w is given by the square of an L^2-normalized Gaussian, with the same standard deviation as the Gabor atoms.

5.1. Sparse Sampling

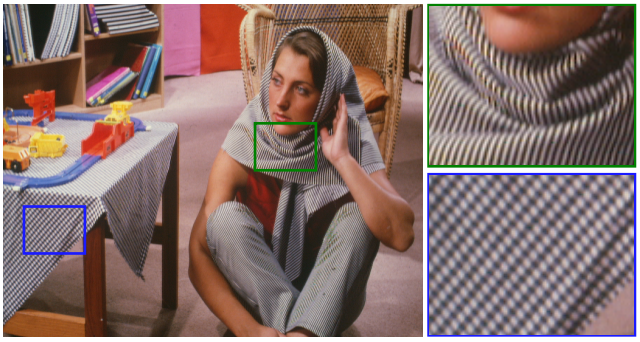
One can improve the computational and storage costs of the pre-processing step by performing a subsampled Gabor decomposition. For a properly selected subsampling step, this procedure conveys the same information as the fully-sampled alternative. This is possible because a fully-sampled Gabor space is highly redundant [Mal09]. Thus, in practice, one can compute the gradient measurements $\nabla\hat{\varphi}(p)$ only every τ pixels (in all directions). This reduces the computational and storage costs by a factor of τ^2 when processing images, and τ^3 for videos. The maximum admissible value of the stepsize $\tau \geq 1$ depends on the Gabor atom's spread [Mal09]. Note that the windowed reconstruction from Eq. (20) naturally handles subsampling: any pixel q without phase information has $N(q) = 0$, and its computed final value $s_\alpha(q)$ is obtained from the neighborhood Ω_q. In this case, however, the proper reconstruction window w depends on τ. A formula for obtaining w can be found in [GO17]. One should also adapt the integration rule in Eq. (8) to consider the step size τ by dividing the left-hand-side by τ, and also computing the shift operators $T_\circ$ using shifts of τ [Sol15].

5.2. Use with Image Resizing

Frequency adjustment is orthogonal to image resizing. For reducing the number of pixels (downscaling), two equivalent procedures exist.

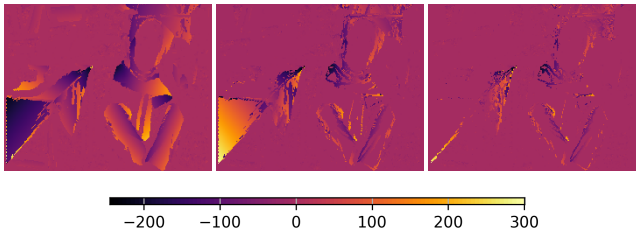
5.2.1. Procedure 1. the frequency-adjusted image (h_α in Eq. (19)) is first reconstructed at the original resolution (preferably with windowing, Eq. (20)), and subsequently downsampled using a suitable strategy, which often involves a prefilter to avoid aliasing [NH14];

5.2.2. Procedure 2. the residual image (d in Eq. (19)) is downscaled independently (by a suitable strategy), and the frequency-adjusted detail (s_α) is reconstructed *directly at the target resolution*. That is, for a downscaling factor $r < 1$, the values of $s_\alpha(p)$ are sampled every $1/r$ units. Note that Eq. (20) naturally supports such fractional sampling in its argument p (with the understanding that the pixels q, inside the summations, represent only valid pixels from the original image's resolution, i.e., located at integer coordinates). The linear extrapolation in the cosine's argument computes the correct phase for any fractional location p, based on its shift $p-q$



(a) *Reference image* (720×576)

(b) *Residual image* (d) **(c)** *Frequency-adjusted detail* (s_α)



(d) *Unwrapped phase components for the red channel* ($\varphi^1_R, \varphi^2_R, \varphi^3_R$)

Figure 6: *Decomposition used for frequency-adjustment of the reference image in (a). (b) Residual returned by the space-frequency decomposition, containing the low-frequency information of the input image. Note the absence of the stripes on the fabrics when compared to (a). (c) High-frequency detail s_α, after being frequency-adjusted with $\alpha = 0.25$ by our technique. Note that values in s_α are permitted to be negative, and the visualization in (c) maps all values to $[0,1]$ for viewing purposes only. (d) Unwrapped phases (in radians) obtained by solving the multichannel Eq. (18), used to compute s_α in (c) (only components for the red channel are shown).*

from q. For this strategy to be effective, however, the adjustment factor α must take into account the target resolution, as it defines a maximum representable spatial frequency; otherwise, fractional sampling will be prone to aliasing (see Section 5.2.3, below).

For image upscaling ($r > 1$), one must reconstruct the high-frequency details s_α directly at the target (higher) resolution. The demo provided in the supplementary materials uses the GPU's mip-map hardware for filtering and rescaling the residual image d, and uses a pixel shader to compute the frequency-adjusted detail s_α, at the target resolution using fractional sampling (either downscaled or upscaled). For additional flexibility, the provided demo allows for independent control of frequency adjustment (α) and image resizing (r) factors, with instant feedback on the resulting image. As one would expect, depending on the selected values for these parameters, aliasing may occur. Alternatively, the two factors may be "linked," automatically adjusting the frequencies in synchrony with image resizing and avoiding aliasing.

Figure 6 illustrates the decomposition $s_\alpha + d$ (Eq. (19)) for the image in (a), subject to frequency adjustment using $\alpha = 0.25$. It also shows the unwrapped phase components (for the red channel) used by our application for real-time frequency adjustment. It is important to note that the residual d may be upsampled or downsampled using any applicable strategy before being added to the reconstructed frequencies in s_α. As such, our method is orthogonal to any resampling technique (Section 6).

5.2.3. Choosing α. The selection of the frequency-adjustment factor α should consider some criteria. Thus, let $\omega_{\max} \in [0,\pi]$ be the maximum frequency (in *radians per pixel*) present in an image at its original resolution. After undergoing simultaneous resizing by a factor $r > 0$ and frequency adjustment by a factor $\alpha > 0$, $\omega_{\max}$ becomes $\omega_r = (\alpha/r)\,\omega_{\max}$. In order to avoid aliasing in the resulting image, the Nyquist limit must be observed:

$$(\alpha/r)\,\omega_{\max} \leq \pi. \tag{21}$$

If the original image encompasses the full available bandwidth, then by definition $\omega_{\max} = \pi$, and Eq. (21) results in the requirement that $\alpha \leq r$. Most commonly, however, structured patterns in images rarely represent the highest possible frequency (not considering the piecewise discontinuities that occur, for example, along object edges—the frequencies associated with these discontinuities are deliberately not included in the high-frequency detail s by the space-frequency analysis) As such, one can often select $\alpha > r$ and still obtain good (aliasing-free) results. The exact value of α used for a particular application is a subjective choice, left to the user, but should satisfy Eq. (21) to avoid aliasing.

In our demo, we provide the following options: *(i)* α freely controllable by the user; *(ii)* $\alpha = r\alpha^*$ where α^* is user-controllable (i.e., α is proportional to r with some user-adjustable proportionality constant); and *(iii)* $\alpha = \sqrt{r}\alpha^*$. We call this last (empirically-defined) option a "perceptual" selection for α, as it leads to downscaled images that better match a user's expectations (refer to Figure 7).

5.3. Processing Videos and 3-D Volumes

Extending our technique to videos is straightforward. While one could consider independently applying our method to the individual frames, this introduces visual artifacts due to discontinuities in the recovered phases for adjacent frames. A more robust solution processes the whole 3-D (x,y,t) video volume at once. For the phase unwrapping pre-processing step, this is done by adding a new set of linear constraints relating to the third dimension: that is, simply by considering $\circ \in \{x,y,t\}$ in Eqs. (13) and (17) and all associated equations. The resulting 3-D phase functions $\varphi^n(x,y,t)$ are stored and later employed for real-time frequency adjustment by combining the frequency-adjusted components $\cos(\alpha\varphi^n(x,y,t))$ according to Eq. (20) (windowed reconstruction with a 3-D neighborhood Ω_p).

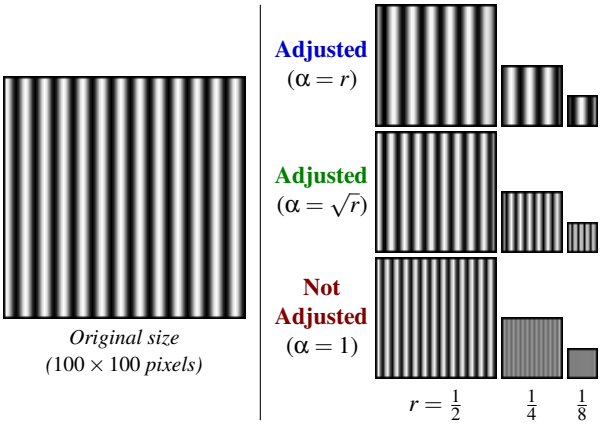


Figure 7: *Perceptual frequency adjustment. The image on the left was downscaled using factors $r \in \{\frac{1}{2}, \frac{1}{4}, \frac{1}{8}\}$. If no frequency adjustment is performed before downscaling (right-bottom row), the high-frequency information is lost at the smaller image sizes. If frequencies are adjusted using a factor $\alpha = r$ (right-top row), the striped pattern is correctly preserved at all resolutions. However, note how all resulting images have sinusoids with the same period, measured in pixels. This results in a "perceptual dissonance" since, while the image size is being reduced, the pattern is perceived as staying the same size (since the wave periods are constant). This is solved by using a "perceptual" adjustment factor $\alpha = \sqrt{r}$ (right-center row), which reduces the pattern's period lengths while still preserving the stripes.*

As in the case for images, this process is completely independent per pixel and can be computed in parallel on a GPU. Furthermore, this can be done during video playback where only the current frame has to be reconstructed. The same considerations apply for processing 3-D volumes in general (3-D scalar fields).

5.4. Complexity Analysis

The cost of the space-frequency analysis (Appendix A) for an input image with $K = w \times h$ pixels using an $L \times L$ window is $O(KL \log L)$. Phase unwrapping requires solving a sparse linear system with $O(K/\tau)$ non-zero elements, where τ is the sampling step.

To reconstruct a target image with M pixels and at most N detected phase components per pixel, it requires MN element-wise multiplications of an $L \times L$ reconstruction window by the (per-pixel) cosine functions corresponding to the unwrapped phases, resulting in a cost of $O(MNL^2)$. Using sparse sampling with a step τ in both dimensions, the cost reduces to $O(MN \lfloor L/\tau \rfloor^2)$.

The reconstruction algorithm uses as input a residual image with the same dimensions $w \times h$ as the original image, and 2-D arrays containing pre-computed phase (unwrapped), amplitude, and gradients for the N components. Each such array has dimensions $w+L \times h+L$. The extra L rows and columns are required as reconstruction windows outside the image may affect pixels inside it. When sparse sampling with a step τ, only $\lfloor (w+L)/\tau \rfloor \times \lfloor (h+L)/\tau \rfloor$ samples are stored for each array, in which case the amount of memory required by the reconstruction algorithm is $O(wh + M + 4N \lfloor (w+L)/\tau \rfloor \lfloor (h+L)/\tau \rfloor)$.

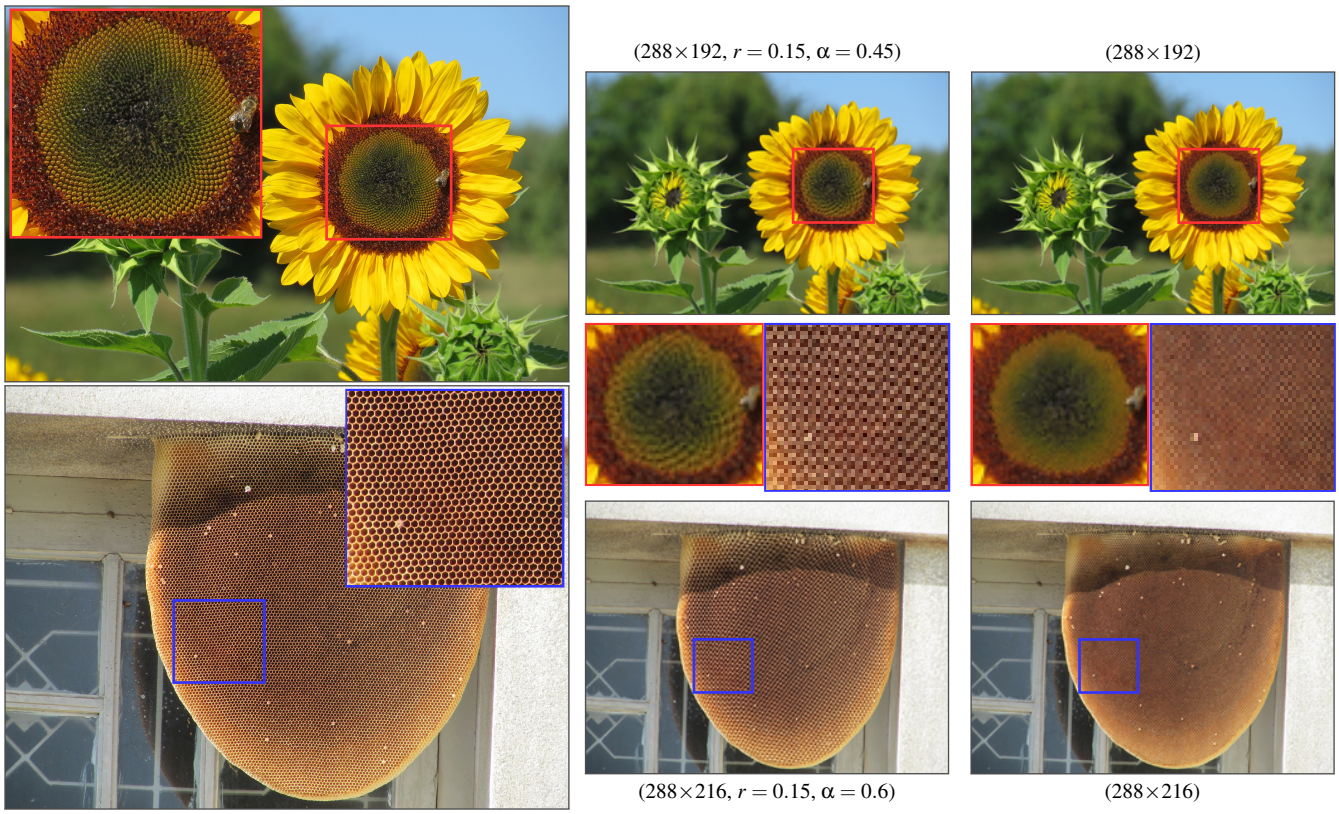
6. Results

We have applied our frequency-adjustment technique to a large number of images, including photographs, hand drawings, as well as synthetic ones. Here we demonstrate the potential and flexibility of our method through a few representative examples illustrating both increase and decrease of spatially-localized frequencies in combination with image resizing, resampling, and super-resolution techniques. Image factorization and phase unwrapping take a few seconds for a full HD image and is performed as a pre-processing. For instance, the factorization of the 1,618 × 2,048-pixel image shown in Figure 1 (a) takes about 8.8 seconds (detecting the relevant waves: 7.83 sec; building and solving an $83{,}633^2$ linear system: 0.93 sec and 0.04 sec, respectively). These measurements were made on a 6-core 3.2 GHz CPU. For all examples shown in the paper, frequency adjustment and image reconstruction are performed in real time on a pixel shader. Our technique performs frequency adjustments on the 3 megapixel image shown in Figure 1 (a) at about 70 fps using $\tau = 10$ on an AMD R9 280X GPU. Although spectral remapping can also detect relevant phases as some pre-processing, it requires building and solving a linear system for phase alignment every time the target resolution changes. For the image shown in Figure 1, these operations take about 1.5 seconds on a 6-core 3.2 GHz CPU.

In all comparisons with other techniques, we used software provided by their own authors. Our technique can generate its results directly at the target image resolution. Nevertheless, in all downscaling examples shown in the paper we perform frequency adjustment at the same resolution as the input image, followed by a resampling to the target dimensions. We do so to stress the fact that our method is orthogonal to and complements existing downscaling techniques [KSP13, NH14, OG15, SN15, WWA*16]. Thus, along the following examples we will be using different resampling techniques. We encourage the reader to inspect our results by zooming in on the full-resolution images embedded on the PDF, and by exploring them in their original sizes in the supplementary materials.

Figure 1 (a) shows a hand drawing containing regions with very high and others with low spatial frequencies. Figure 1 (e) shows the downscaling of the reference image to 135 × 171 pixels using our technique, taking 0.01 seconds for frequency adjustment and image reconstruction. Note how the structured details of the high-resolution reference image are preserved by our technique at the lower image resolution in (b). By comparison, spectral remapping [GO17], in (c), generates a similar result but is two orders of magnitude slower to compute. In both cases, the resampling to the target resolution was performed using the Cubic O-MOMS filter [NH14]. Figures 1 (d–f) show examples of frequency-adjusted results using our technique for different adjustment factors α.

Figure 8 (a) shows two photographs containing natural high frequency patterns. By reducing the frequency of these structures using our approach, one is able to preserve their appearance in a much lower resolution, as shown in (b) for a scaling factor of 0.15×. For these examples the frequency-adjusted images were reconstructed at the original resolution and subsequently downscaled using a 7-lobed



(a) *Reference (top:* 1920 × 1278*, bottom:* 1920 × 1440*)* **(b)** *Ours + Lanczos* **(c)** *Lanczos*

Figure 8: *Examples of adjusting the frequency of complex natural patterns for downscaling. (a) Reference high-resolution images. (b) Downscaled versions of the images in (a) generated by performing frequency adjustment with our technique, followed by resampling using a Lanczos kernel. (c) Downscaled versions of the images in (a) using Lanczos resampling only. Note how our result in (b) manages to preserve the structure in the sunflower's disk and the honeycomb's tiling, which are otherwise discarded by standard downscaling prefilters (c).*

Lanczos kernel. If one downscales the reference image using this same kernel, but without adjusting frequencies using our approach, the detailed patterns of the sunflower and honeycomb are lost, as shown in Figure 8 (c).

Although frequency adjustment is performed at pixel level, it potentially affects content over the entire image. Thus, additional creative control is provided by allowing the user to specify a different frequency scaling factor to individual portions of an image, or to delimit the set of pixels the frequency adjustment should be applied to. This can be achieved by providing a stencil mask specifying the target pixels (Figure 9).

Figure 10 shows the results of our technique applied to some textured fabrics. In these examples, we demonstrate the combined use of our method with Lim et al's. [LSK*17] super-resolution technique. This illustrates our method's ability to complement super-resolution strategies to obtain high-resolution images with a wider frequency bandwidth. Figures 10 (b) and (h) show downscaled versions of the corresponding reference images to 80×60 and 75×75 pixels, respectively, produced by our method. In both cases, resampling was performed using the technique of Weber et al. [WWA*16]. Figures 10 (c)/(i) show results generated by our method when increasing the frequencies by a factor $\alpha = 2$, while preserving the original resolution ($r = 1$). Note how the spatial frequencies in (c)/(i) are higher than the ones in the corresponding reference images (a)/(g). They are also higher than the ones in the results produced by the super-resolution technique of Lim et al. [LSK*17], shown in (e)/(k). Figures 10 (d)/(j) show the results obtained by combining our method with Lim et al.'s. This hybrid solution consists of applying the super-resolution technique to the residual image, which is then added to the adjusted high-frequency detail, according to Eq. (19). Note that such a procedure can be used with any super-resolution method. Figures 10 (d)/(j) show fabrics with significantly higher frequency content in comparison to the plain super-resolution results in (e)/(k), and also to standard bicubic upsampling in (f)/(l). These results demonstrate the effectiveness and simplicity of such a hybrid approach.

Figure 11 shows a photograph of a lizard exhibiting high-frequency details. Figure 11 (b) shows the downscaled version of the reference image to 220×293 pixels generated with frequency adjustments by our technique. Resampling to the target resolution was performed with the optimized quasi interpolator OQI3 [SN15]. Figures 11 (c) and (d) show results produced by our method after

Masks | *scarf:* $\alpha = 0.2$, *pants:* $\alpha = 0.6$ | *scarf:* $\alpha = 0.6$, *pants:* $\alpha = 0.2$

Figure 9: *Stencil masks used to apply independent scaling factors to the spatial frequencies of different image regions: pants (red), scarf (blue), and table cloth (green). These stencils were created using GIMP's* Intelligent Scissors *tool to segment the corresponding regions, which were further dilated by a few pixels to guarantee that no pixels at the borders of high-frequency regions were left out. Alternatively, such regions could potentially be detected by our technique as clusters of neighboring pixels whose reconstructing cosine functions have similar amplitude and phase values.*

scaling the frequencies by factors of $\alpha = 0.25$ and $\alpha = 1.25$, respectively. One should compare the details on the belly and legs of the lizards with the ones on the reference image. For the examples in Figures 11 and 12 no comparisons with the super-resolution technique of Lim et al. [LSK*17] are provided (the available implementation did not handle the sizes of the reference images).

The striped shirt in Figure 12 (a) contains a high-frequency pattern, being a challenging test case for image downscaling techniques. Figure 12 (b) compares the results of our method (top) with spectral remapping (bottom) for the case of an aggressive downscaling of the reference image to 200×150 pixels. In both cases, the actual downscaling step was performed with Lanczos filtering followed by resampling using a cubic B-spline. Figure 12 (c) shows an example of frequency increase (applied to an already high-frequency pattern). For this result, we kept the dimensions of the original image ($r = 1$) while increasing the frequency of the stripes using $\alpha = 1.5$. This example also demonstrates the effectiveness of our solution for processing color images (Section 4.3), as misalignment in the phases of the individual color channels would introduce noticeable artifacts in such a large image (see accompanying video).

Figure 13 illustrates the use of our technique for processing a video sequence. As discussed in Section 5.3, the whole 3-D (x, y, t) volume is processes simultaneously, thus preserving temporal coherence of the frequency-adjusted details. The reference and resulting processed videos are included in our supplementary materials.

6.1. Discussion and Limitations

As discussed in Appendix A, we use a Gabor space-frequency decomposition for measuring the instantaneous frequencies $\nabla\hat{\varphi}$ present in the image's spectrum [GO17]. One limitation of this approach is that the (Gaussian-windowed) Gabor atoms have a fixed bandwidth in space and frequency, controlled by a user-specified frequency threshold $\bar{\omega}$. If one wants to detect the instantaneous frequencies of very slowly varying components, the threshold $\bar{\omega}$ must be small in order to obtain a good frequency (spectral) resolution. But in turn this comes at the cost of poorer spatial resolution, since space and frequency locality are conflicting requirements [Mal09]. In practice, this means that reducing the frequency threshold leads to larger analysis and reconstruction windows for the Gabor frame. Such large windows may cause some detected high-frequencies to bleed into nearby pixels during frequency adjustment. These "blended frequencies" are often hidden by the downscaling procedure, but may be visible when the resulting image is upscaled or has its size preserved. This can be seen, for example, around the letters in Figure 10 (d) and around the buttons in Figure 12 (c). This problem could potentially be mitigated by replacing the Gabor step with a Wavelet strategy for detecting instantaneous frequencies [DEG*92]. Since wavelets adapt their spatial support based on the frequency bands of the signal, they define a multiscale approach which would use the smallest possible window for each detected frequency.

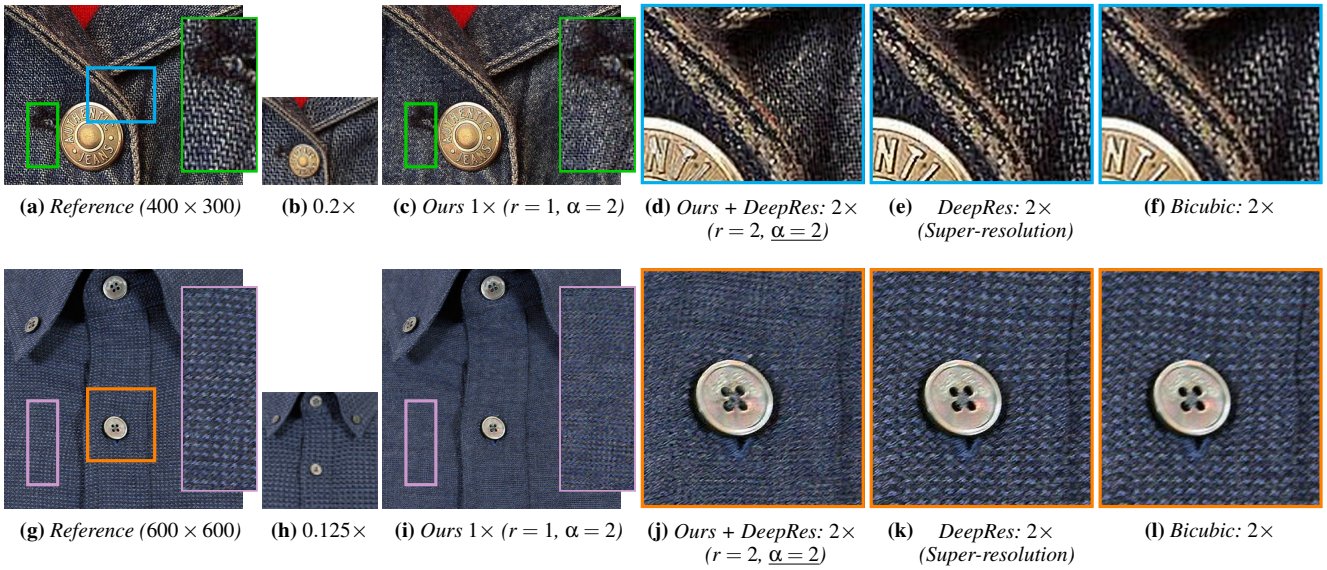
As mentioned in Section 4.3, for some color images it is best to use the PCA-based approach for phase unwrapping (processing only a single "channel"). This occurs because, in some situations, the inter-channel alignment is not enforceable without breaking intra-channel phase constraints. This is the case of the images in Figure 10 (a) and (g), which are highly monotone, and for which a multi-channel RGB alignment introduces some small chromatic distortions (Figure 14).

7. Conclusion

We presented a real-time technique for interactive adjustment of the spatial frequencies in images. It supports both decreasing and increasing of frequencies with localized spatial control, and is orthogonal to image resizing and resampling strategies. Conceptually, our method consists of factoring the input image, at a pixel level, into a set of high-frequency waves (cosines) plus some residual. Frequency adjustment is achieved by scaling the phase values of the detected waves at each pixel by some factor α. Image reconstruction is then obtained (also at pixel level) by accumulating the cosines of the scaled phase values and adding the corresponding residual.

Our paper introduces two additional contributions: *(i)* a phase-unwrapping technique to recover the phase of real waves, and *(ii)* a technique for processing color images for use with frequency adjustments. Our phase-unwrapping solution naturally handles the sign ambiguity associated with the unwrapping of real waves. To the best of our knowledge, ours is the first algorithm to solve this problem in 2-D. Our method for processing color images avoids color artifacts associated with the use of the PCA-based color reconstruction described in [GO17]. We demonstrated the effectiveness of our technique on a large number of images subject to downscaling, upscaling, as well with their dimensions preserved. We have also shown how to extend our method to videos and how to combine it with conventional super-resolution techniques to obtain novel super-resolution results.

By performing frequency adjustment with real-time feedback, our technique expands the range of artistic and technical possibilities for image and video processing. This should stimulate the development



(a) *Reference* (400 × 300) **(b)** 0.2× **(c)** *Ours* 1× ($r = 1$, $\alpha = 2$) **(d)** *Ours + DeepRes:* 2× ($r = 2$, $\alpha = 2$) **(e)** *DeepRes:* 2× *(Super-resolution)* **(f)** *Bicubic:* 2×

(g) *Reference* (600 × 600) **(h)** 0.125× **(i)** *Ours* 1× ($r = 1$, $\alpha = 2$) **(j)** *Ours + DeepRes:* 2× ($r = 2$, $\alpha = 2$) **(k)** *DeepRes:* 2× *(Super-resolution)* **(l)** *Bicubic:* 2×

Figure 10: *Examples of downscaling and upscaling with frequency adjustments. (a) and (g) Reference images. (b) and (h) Downscaled versions of (a) and (g) generate by performing frequency adjustment with our technique, followed by resampling using the method of [WWA*16]. (c) to (d) and (i) to (j) show examples of frequency adjustments using our technique. (c) and (i): Increasing spatial frequency using a scaling factor $\alpha = 2$, while keeping the dimensions of the reference image. (d) and (j) Increasing the spatial frequency using $\alpha = 2$ while resizing the image by 2× (using Lim et al.'s [LSK*17] super-resolution method for upscaling the residual). (e) and (k) Upscaling by 2× using only Lim et al.'s [LSK*17] super-resolution method. (f) and (l) Upscaling by 2× using standard bicubic upsampling. Note how our approach is able to generate high-frequency detail that is significantly different than existing classic and state-of-the-art methods.*

of new applications involving real-time exploration of visual content on small as well as on very large displays. This could include, for instance, real-time image downscaling to adapt content to thumbnail sizes or to smartphone displays. Similarly, one could combine our frequency-enhancement solution with super-resolution methods to produce upscaled images for exploration on high-definition displays.

Acknowledgements. This work was sponsored by CNPq-Brazil (fellowships and grants 131731/2019-0, 436932/2018-0, 131102/2018-4, 312975/2018-0, 423673/2016-5), and financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

References

[BH13] BARBHUIYA A. J. I., HEMACHANDRAN K.: Wavelet tranformations & its major applications in digital image processing. *Intl' Journal of Engineering Research & Technology (IJERT)* (2013), 2278–0181. 2

[CSK77] CHILDERS D., SKINNER D., KEMERAIT R.: The cepstrum: A guide to processing. *Proc. of the IEEE 65*, 10 (1977), 1428–1443. 3

[Dav11] DAVIS T. A.: Algorithm 915, SuiteSparseQR: Multifrontal Multithreaded Rank-revealing Sparse QR Factorization. 8:1–8:22. 7

[DEG*92] DELPRAT N., ESCUDIE B., GUILLEMAIN P., KRONLAND-MARTINET R., TCHAMITCHIAN P., TORRESANI B.: Asymptotic wavelet and Gabor analysis: extraction of instantaneous frequencies. *IEEE Transactions on Information Theory 38*, 2 (Mar. 1992), 644–664. 3, 11

[FF10] FREEDMAN G., FATTAL R.: Image and video upscaling from local self-examples. *ACM Trans. Graph. 28*, 3 (2010), 1–10. 3

[GKM96] GUILLEMAIN P., KRONLAND-MARTINET R.: Characterization of acoustic signals through continuous linear time-frequency representations. *Proceedings of the IEEE 84*, 4 (Apr. 1996), 561–585. 3

[GO17] GASTAL E. S. L., OLIVEIRA M. M.: Spectral remapping for image downscaling. *ACM Trans. Graph. 36*, 4 (2017), 145:1–145:16. 1, 2, 4, 6, 7, 9, 11, 13, 14, 15

[GP98] GHIGLIA D. C., PRITT M. D.: *Two-dimensional phase unwrapping: theory, algorithms, and software*. Wiley, Apr. 1998. 3, 4

[GZW88] GOLDSTEIN R. M., ZEBKER H. A., WERNER C. L.: Satellite radar interferometry: Two-dimensional phase unwrapping. *Radio Science 23*, 4 (July 1988), 713–720. 3

[HPV19] HERSZTERG I., POGGI M., VIDAL T.: Two-Dimensional Phase Unwrapping via Balanced Spanning Forests. *INFORMS Journal on Computing 31*, 3 (July 2019), 527–543. arXiv: 1812.08277. 15

[HSL*98] HUANG N. E., SHEN Z., LONG S. R., WU M. C., SHIH H. H., ZHENG Q., YEN N.-C., TUNG C. C., LIU H. H.: The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis. *Proceedings of the Royal Society of London. Series A. 454*, 1971 (1998), 903–995. 4

[Hua14] HUANG N. E.: *Hilbert-Huang transform and its applications*, vol. 16. World Scientific, 2014. 4

[Ito82] ITOH K.: Analysis of the phase unwrapping algorithm. *Applied Optics 21*, 14 (July 1982), 2470–2470. 15

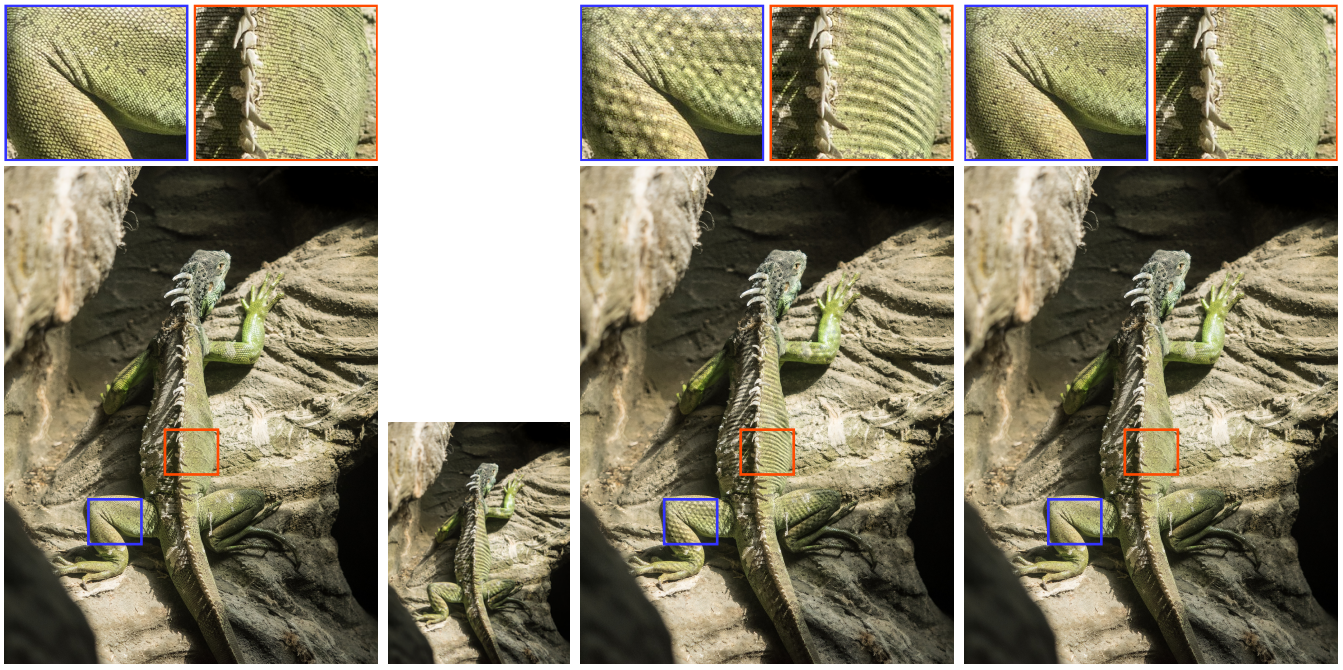
[KSP13] KOPF J., SHAMIR A., PEERS P.: Content-adaptive image downscaling. *ACM Trans. Graph. 32*, 6 (Nov. 2013), 173:1–173:8. 2, 9

[LO10] LILLY J. M., OLHEDE S. C.: On the Analytic Wavelet Transform. *IEEE Trans. on Information Theory 56*, 8 (Aug. 2010), 4135–4156. 3

[LSK*17] LIM B., SON S., KIM H., NAH S., LEE K. M.: Enhanced deep residual networks for single image super-resolution. In *IEEE CVPR Workshops* (July 2017). 3, 10, 11, 12

[Mal09] MALLAT S. G.: *A Wavelet Tour of Signal Processing: The Sparse Way*, 3rd ed. Elsevier, 2009. 2, 3, 4, 5, 7, 11

[Mer79] MERTZ L. N.: Speckle imaging, photon by photon. *Applied Optics 18*, 5 (Mar. 1979), 611–614. 4



(a) *Reference* (1,536 × 2,048) **(b)** ($r = \alpha = 0.14$) **(c)** *Ours* ($r = 1$, $\alpha = 0.25$) **(d)** *Ours* ($r = 1$, $\alpha = 1.25$)

Figure 11: *Picture of a lizard (courtesy of William Warby). (a) Reference image. (b) Downscaled version of (a) generate with our technique for frequency adjustment and resampling to* 220×293 *using the method of [SN15]. (c) and (d) Examples of spatial frequency adjustment generated by our technique. The images were kept at the original resolution* ($r = 1$)*: (c) Frequency decreasing obtained with a scaling factor* $\alpha = 0.25$*. (d) Frequency increasing obtained with a scaling factor* $\alpha = 1.25$*. Note the adjusted frequencies of the details on legs and belly of the lizard.*

(a) *Reference* (1,600 × 1,200) **(b)** 0.125× **(c)** *Ours* ($r = 1$, $\alpha = 1.5$) (1,600 × 1,200)

Figure 12: *Shirt. (a) Reference image. (b) Downscaled versions of (a): (top) generated with our technique and (bottom) obtained with spectral remapping [GO17]. In both cases, resampling to* 200×150 *pixels was done applying Lanczos filtering followed by resampling using a cubic B-spline. (c) Result produced by our techniques with same resolution as the reference image* ($r = 1$)*, but with spatial frequency increased by a factor* $\alpha = 1.5$*. Please zoom-in to images (a) and (b) to avoid aliasing artifacts caused by the PDF reader's resampling filter.*

(a) *Frame from the reference video* (760×720) **(b)** *Frames from our frequency-adjusted video (downscaled to* 228×216 *using Lanczos filtering)*

Figure 13: *Example of video frequency adjustment. (a) The reference video contains high-frequency patterns on the jacket, which move around from frame to frame following the person's movement. (b) Our frequency adjustment technique is applied to the whole 3-D space-time video volume (Section 5.3), reducing the frequency of the patterns and preserving them at the target (lower) resolution.*

Figure 14: *Chromatic distortions might happen when enforcing multi-channel alignment for images that are mostly monotone (see the button on the left and the shirt texture on the right). Images obtained using* $\alpha = 0.5$ *and* $r = 1$.

[NH14] NEHAB D., HOPPE H.: A fresh look at generalized sampling. *Foundations and Trends in C.G. and Vision 8*, 1 (2014), 1–84. 1, 2, 7, 9

[OG15] ÖZTIRELI A. C., GROSS M.: Perceptually based downscaling of images. *ACM Trans. Graph. 34*, 4 (July 2015), 77:1–77:10. 2, 9

[OS10] OPPENHEIM A. V., SCHAFER R. W.: *Discrete-time signal processing*. Pearson Higher Education, 2010. 2, 3, 4

[PI97] PRASAD L., IYENGAR S.: *Wavelet Analysis with Applications to Image Processing*. Taylor & Francis, 1997. 2

[RAGS01] REINHARD E., ADHIKHMIN M., GOOCH B., SHIRLEY P.: Color transfer between images. *IEEE Computer Graphics and Applications 21*, 5 (July 2001), 34–41. 6

[SF85] SCIVIER M. S., FIDDY M. A.: Phase ambiguities and the zeros of multidimensional band-limited functions. *JOSA A 2*, 5 (May 1985), 693–697. 15

[SN15] SACHT L., NEHAB D.: Optimized Quasi-Interpolators for Image Reconstruction. *IEEE TIP 24*, 12 (2015), 5249–5259. 2, 9, 10, 13

[SNPG95] SONG S. M.-H., NAPEL S., PELC N., GLOVER G.: Phase unwrapping of MR phase images using Poisson equation. *IEEE TIP 4*, 5 (1995), 667–676. 15

[Sol15] SOLOMON J.: *Numerical algorithms: methods for computer vision, machine learning, and graphics*. CRC Press, Taylor & Francis, 2015. 5, 7

[TT88] TAKAJO H., TAKAHASHI T.: Least-squares phase estimation from the phase difference. *JOSA A 5*, 3 (Mar. 1988), 416–425. 15

[WRDF13] WADHWA N., RUBINSTEIN M., DURAND F., FREEMAN W. T.: Phase-based video motion processing. *ACM Trans. Graph. 32*, 4 (2013). 4

[WRDF14] WADHWA N., RUBINSTEIN M., DURAND F., FREEMAN W.: Quaternionic representation of the Riesz pyramid for video magnification. MIT-CSAIL-TR-2014-009. 5

[WWA*16] WEBER N., WAECHTER M., AMEND S. C., GUTHE S., GOESELE M.: Rapid, detail-preserving image downscaling. *ACM Trans. Graph. 35*, 6 (2016), 205:1–205:6. 2, 9, 10, 12

[YWHM10] YANG J., WRIGHT J., HUANG T. S., MA Y.: Image super-resolution via sparse representation. *IEEE TIP 19*, 11 (Nov 2010), 2861–2873. 3

[ZL98] ZEBKER H. A., LU Y.: Phase unwrapping algorithms for radar interferometry: residue-cut, least-squares, and synthesis algorithms. *JOSA A 15*, 3 (Mar. 1998), 586–598. 15

Appendix A: Estimating the phase gradient field for one real wave

The pre-processing step of our technique begins by obtaining a non-harmonic space-frequency decomposition of the image h. For this purpose, we employ the recent algorithm of Gastal and Oliveira [GO17], which separates the image as $h = s + d$, where s contains most of the high-frequency information of h, for a certain predefined frequency threshold (Figure 6). This decomposition approximates the neighborhood around each pixel as a collection of plane waves of the form $A\cos(ax+by+c)$, where $(a,b) \in [-\pi,\pi)^2$ are constant horizontal and vertical frequencies, $c \in [0,2\pi)$ is a constant phase shift, and $A > 0$ is a constant amplitude. In other words, assuming a single phase component (the case of several components is discussed in Section 4.2), a tuple of measurements (a,b,c) is computed for each pixel (x_0,y_0), and the approximation

$$s(x_0+\delta_x, y_0+\delta_y) \approx A\cos(\hat{\omega}) = A\cos(a\,\delta_x + b\,\delta_y + c) \qquad (22)$$

is the best linear-phase approximation of the image's high-frequency content s around pixel (x_0,y_0). The (unknown) true phase value at

(x_0, y_0) matches the measured phase shift c up to modulo 2π (i.e., c is a wrapped measurement).

By definition, a function and its best linear approximation at point (x_0, y_0) have matching first derivative at such point. Thus, for the desired representation $s(x,y) = A\cos(\theta(x,y))$, with a possibly non-linear phase function $\theta(x,y)$, and given Eq. (22), with (a,b) computed for a particular pixel (x_0, y_0), one has

$$\nabla\theta(x_0,y_0) = \nabla\hat{\omega} = (a,b). \tag{23}$$

Moreover, since $\cos(\hat{\omega}) = \cos(-\hat{\omega})$, it is also true that

$$s(x+\delta_x, y+\delta_y) \approx A\cos(-\hat{\omega}) = A\cos(-a\,\delta_x - b\,\delta_y - c); \tag{24}$$

so it could also be the case that we have a sign flip:

$$\nabla\theta(x_0,y_0) = -\nabla\hat{\omega} = (-a,-b). \tag{25}$$

In other words, $\nabla\theta(x_0,y_0) = \sigma(x_0,y_0)\,\nabla\hat{\varphi}(x_0,y_0)$ for an unknown sign $\sigma(x_0,y_0) \in \{-1,+1\}$, where the measured phase gradient

$$\nabla\hat{\varphi}(x_0,y_0) = (a,b) \tag{26}$$

is given by the horizontal and vertical plane-wave frequencies a and b, detected at pixel (x_0, y_0), by a non-harmonic Gabor space-frequency analysis [GO17].

Appendix B: Unwrapping the phase of one complex wave

The phase-unwrapping literature deals with wrapped phase values $\hat{\theta}(x,y) \in [-\pi,\pi)$ that are related to the true phase θ according to Eq. (5). In practice, this is equivalent to a phase obtained from a *complex* wave $\exp(i\theta(x,y))$. Note that the wrapped phase recovered from a complex wave is given by $\hat{\theta} = \arg(\exp(i\theta)) \in [-\pi,\pi)$ [Ito82], thus no sign ambiguity occurs. Since the integer shifts $k(x,y)$ in Eq. (5) define a piecewise-constant function, the unwrapped and wrapped gradients are such that

$$\nabla\theta(x,y) = \nabla\hat{\theta}(x,y) \quad \text{almost everywhere.} \tag{27}$$

If the vector field $\nabla\hat{\theta}(x,y)$ is conservative, then the scalar potential field $\theta(x,y)$ is directly determined by a path integral (up to some integration constant) [TT88].

In the discrete setting, the wrapped phase $\hat{\theta}(x,y)$ is only sampled at integer pixel locations, and the discrete analogue to Eq. (27) is given by finite differences. Some extra care must be taken, however, to handle the discontinuities caused by the integer shifts $k(x,y)$. Appendix C reviews how this is done in the phase-unwrapping literature [Ito82]. The end result is an estimated discrete gradient field $\nabla\tilde{\theta}$ that approximates the true field $\nabla\theta$ at each pixel (x,y), and the unwrapped phase $\theta(x,y)$ is obtained by integrating $\nabla\tilde{\theta}$. While, in theory, this may be performed by a cumulative sum over any path covering all pixels in the image, this process is sensitive to noise and errors in the measurements [ZL98], in addition to ambiguities inherent to the phase-unwrapping problem [SF85]. Thus, in practice, the integration process becomes path dependent and a good choice of integration path must be made. Modern approaches obtain good results by the use of combinatorial optimization with carefully-selected heuristics [HPV19]. Unfortunately, the time complexity of these methods scale as $O(N^3)$ in the number of pixels.

A second class of algorithms work by integrating the field $\nabla\tilde{\theta}$ in the least-squares sense [TT88, SNPG95, ZL98]. This is equivalent to solving a discrete Poisson problem. The benefit of these approaches is that they recover smoothly-varying phase information even in regions of low signal-to-noise ratio, but at the cost of some distortion in the recovered phase (which may not exactly match the wrapped measurements) [ZL98]. For our frequency-adjustment technique, obtaining a smooth phase field is of utmost importance, as any discontinuity tends to introduce objectionable visual artifacts. As such, we opt for a least-squares solution, discussed in Sections 4.1.1 and 4.2. Since this amounts to solving a linear system, the unwrapping computation takes at most a few seconds for a typical image at full HD resolution on commodity hardware. The actual time depends on the image resolution and its frequency content. Note, however, that recovering $\theta(x,y)$ can be done as pre-processing, after which one can perform spatial frequency adjustments in real-time using independent per-pixel operations (Section 5).

Appendix C: Common approach for estimating the wrapped gradient field for one complex wave

The following relation holds between the phase θ of a complex wave and its wrapped version $\hat{\theta}(x,y) = \arg(\exp(i\theta(x,y)))$ [Ito82]:

$$\Delta\theta(x,y) = \text{wrap}(\Delta\hat{\theta}(x,y)) \quad \text{if} \quad \Delta\theta(x,y) \in [-\pi,\pi)^2; \tag{28}$$

where $\Delta = (\Delta_x, \Delta_y)$ gives the discrete gradient, for

$$\Delta_x\theta(x,y) = \theta(x,y) - \theta(x-1,y), \text{ and}$$
$$\Delta_y\theta(x,y) = \theta(x,y) - \theta(x,y-1).$$

The operator $\text{wrap}(\theta) = \theta + 2\pi\lfloor(\pi-\theta)/2\pi\rfloor$ operates elementwise when applied to a vector [HPV19]. The identity in Eq. (28) holds everywhere if the horizontal and vertical variations in θ are bounded by π radians [Ito82]. This restriction, that $\Delta\theta(x,y) \in [-\pi,\pi)^2$, is trivially satisfied for sampled complex signals. The continuous signal reconstruction with sinc interpolation is naturally band-limited to the Nyquist frequency interval $[-\pi,\pi)^2$. This implies that the components of the instantaneous frequency $\nabla\theta(x,y)$ are in $[-\pi,\pi)$, and consequently $\Delta\theta(x,y) \in [-\pi,\pi)^2$. As such, given measurements $\hat{\theta}(x,y)$ of the wrapped phase, the unwrapped phase $\theta(x,y)$ for one complex wave can be obtained by integrating $\text{wrap}(\Delta\hat{\theta}(x,y))$ [Ito82, SNPG95].

Appendix D: Cross-channel phase constraints

The cross-channel constraints $\mathcal{X}_{c_1c_2}$ from Eq. (18) are expressed as

$$\mathcal{X}_{c_1c_2} = \sum_{n=1}^{N_{c_1}}\sum_{m=1}^{N_{c_2}} \eta^{nm}_{c_1c_2}\,(\varphi^n_{c_1} - \Delta^{nm}_{c_1c_2}\sigma\,\varphi^m_{c_2})^2, \tag{29}$$

where, as before, $\Delta^{nm}_{c_1c_2}\sigma(p)$ determines if there is an expected change in sign between the gradient vectors $\nabla\hat{\varphi}^n_{c_1}(p)$ and $\nabla\hat{\varphi}^m_{c_2}(p)$. Similarly, the weight $\eta^{nm}_{c_1c_2}(p)$ measures the similarity of such vectors, according to their orientations and magnitudes:

$$\eta^{nm}_{c_1c_2} = \text{Ang}\,(\nabla\hat{\varphi}^n_{c_1}, \nabla\hat{\varphi}^m_{c_2})\,\text{Mag}\,(\nabla\hat{\varphi}^n_{c_1}, \nabla\hat{\varphi}^m_{c_2}). \tag{30}$$

The orientation similarity is given by dot product

$$\text{Ang}(v_1,v_2) = \frac{|\,v_1\cdot v_2\,|}{\|v_2\|\,\|v_2\|}, \tag{31}$$

and the magnitude similarity by

$$\text{Mag}(v_1,v_2) = \exp\left(-\tfrac{1}{2}\,\frac{(\|v_1\|-\|v_2\|)^2}{(\pi/3)^2}\right). \tag{32}$$