

Blur Processing Using Double Discrete Wavelet Transform

Yi Zhang Keigo Hirakawa

Department of Electrical & Computer Engineering, University of Dayton

<http://campus.udayton.edu/~ISSL>

Abstract

We propose a notion of double discrete wavelet transform (DDWT) that is designed to sparsify the blurred image and the blur kernel simultaneously. DDWT greatly enhances our ability to analyze, detect, and process blur kernels and blurry images—the proposed framework handles both global and spatially varying blur kernels seamlessly, and unifies the treatment of blur caused by object motion, optical defocus, and camera shake. To illustrate the potential of DDWT in computer vision and image processing, we develop example applications in blur kernel estimation, deblurring, and near-blur-invariant image feature extraction.

1. Introduction

Image blur is caused by a pixel recording lights from multiple sources. Illustrated in Figure 1 are three common types of blur: defocus, camera shake and object motion. Defocus blur is caused by a wide aperture that prevents light rays originating from the same point from converging. Camera motion during the exposure produces global motion blur where the same point on the scene is observed by multiple moving pixel sensors. Object motion causes each pixel to observe multiple points on the scene that produces spatially-variant motion blur. Assuming Lambertian surfaces, blur is typically represented by the implied blur kernel that acts on the unobserved sharp in-focus image.

Blur is a part of everyday photography. On one hand, long exposure is needed to overcome poor lighting conditions, but it increase the risk of camera shake and object motion blurs that severely deteriorate the sharpness of the image. Automatic or manual focus is also a challenge when the scene covers a wide range of depths or is rapidly changing (e.g. sports photography), often causing unwanted defocus blur. On the other hand, professional photographers use well-controlled blur to enhance the aesthetics of a photograph. Thus the ability to manipulate blur in post-processing would offer a greater flexibility in consumer and professional photography.

Blur is also valuable for computer vision. Blur may vary across the spatial location (e.g. a scene with multiple mov-



(a) Defocus (b) Camera shake (c) Object motion

Figure 1. Variations on the types of image blur.

ing objects or multiple depths) or be global (e.g. camera shake). We may learn the temporal state of the camera and the scene from blur caused by camera shake or object motion, respectively. The defocus blur kernel varies with the object distance/depth, which can be useful for three dimensional scene retrieval from a single camera[23]. Blur also interferes with recognition tasks, as feature extraction from blurry image is a real challenge.

In this paper, we propose a novel framework to address the analysis, detection, and processing of blur kernels and blurry images. Central to this work is the notion of double discrete wavelet transform (DDWT) that is designed to sparsify the blurred image and the blur kernel simultaneously. We contrast DDWT with the work of [3], which regularizes image and blur kernel in terms of their sparsity in linear transform domain. The major disadvantage of regularization approach is that the image/blur coefficients are not directly observed, hence requiring a computationally taxing search to minimize some “cost” function. On the other hand, our DDWT provides a way to observe the wavelet coefficients of image and blur kernel directly. This gives DDWT coefficients a very intuitive interpretation, and simplify the task of decoupling the blur from the signal, regardless of why the blur occurred (e.g. object motion, defocus, and camera shake) or the type of blur (e.g. global and spatially varying blur). In this sense, DDWT is likely to impact computer vision and image processing applications broadly. Although the primary goal of this article is to develop the DDWT as an analytical tool, we also show example applications in blur kernel estimation, deblurring, and near-blur-invariant image feature extraction to illustrate the potential of DDWT.

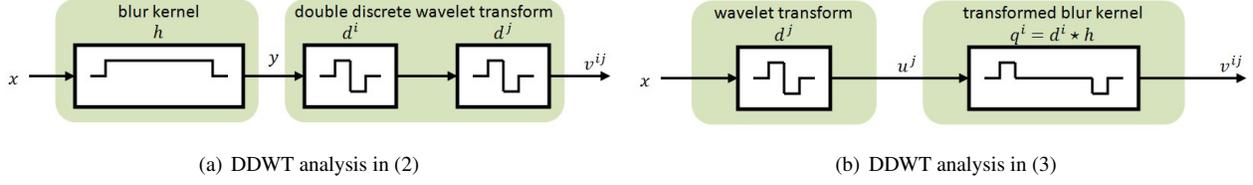


Figure 2. The two processing pipelines above are equivalent. Though (a) is the direct result of applying DDWT ($d^i \star d^j$) to the observed blurry image y , (b) is the interpretation we give to the DDWT coefficients.

2. Related Work

Recent advancements on blind and non-blind deblurring have enabled us to handle complex uniform blur kernels (e.g. [20, 19, 12, 3, 2, 21, 22, 9]). By comparison, progress in blind and non-blind deblurring for spatially varying blur kernel (e.g. [16, 17, 4, 6]) has been slow since there are limited data availability to support localized blur kernel. For this reason, it is more common to address this problem using multiple input images [1, 7] and additional hardware [21, 18, 8]. Approaches to computational solutions include supervised [16] or unsupervised [17, 10] foreground/background segmentation, statistical modeling [4], homography based blur kernel modeling methods [24, 15] and partial differential equation (PDE) methods [6]. In particular, sparsifying transforms have played key roles in the detection of blur kernels—gradient operator [4, 6, 10, 9] and wavelet/framelet/curvelet transforms [3, 11] have been used for this purpose.

However, existing works have shortcomings, such as problems with ringing artifact in deblurring [20, 5] or inability to handle spatially varying blur [19, 12, 3, 2, 21, 22, 9]. It is also common for deblurring algorithms to require iteration [3, 20, 5, 12, 24, 15], which is highly undesirable for many real-time applications. Besides PDE, authors are unaware of any existing framework that unify analysis, detection, and processing of camera shake, object motion, defocus, global, and spatially varying blurs.

3. Double Discrete Wavelet Transform

3.1. Definitions

We begin by defining *single discrete wavelet transform* (DWT) and the proposed *double discrete wavelet transform* (DDWT). Both transforms are defined to take a over-complete (a.k.a. undecimated) form and invertible.

Definition 1 (DWT). *Let $y : \mathbb{Z}^2 \rightarrow \mathbb{R}$ be an image signal and $\mathbf{n} \in \mathbb{Z}^2$. Denote by $d^j : \mathbb{Z}^2 \rightarrow \mathbb{R}$ a wavelet analysis filter of j th subband. Then*

$$w^j(\mathbf{n}) := \{d^j \star y\}(\mathbf{n})$$

is the j th subband, \mathbf{n} th location over-complete single discrete wavelet transform coefficient of an image $y(\mathbf{n})$, where \star denotes a convolution operator.

Definition 2 (DDWT). *The over-complete double discrete wavelet transform is defined by the relation*

$$v^{ij}(\mathbf{n}) := \{d^i \star w^j\}(\mathbf{n}),$$

where $v^{ij}(\mathbf{n})$ is the transform coefficient of the image $y(\mathbf{n})$ in the (i, j) th subband and location \mathbf{n} .

In the special case that $d^j(\mathbf{n})$ is a 1D horizontal wavelet analysis filter and $d^i(\mathbf{n})$ is a vertical one, then $v^{ij}(\mathbf{n})$ is an ordinary separable 2D wavelet transform. In our work, however, we allow the possibility that $d^j(\mathbf{n})$ and $d^i(\mathbf{n})$ to be arbitrarily defined (e.g. both horizontal). Technically speaking, the DWT/DDWT definitions above may apply to non-wavelet transforms d^j and d^i , so long as they are invertible.

3.2. DDWT Analysis Of Blurred Image

Assuming Lambertian reflectance, let $x : \mathbb{Z}^2 \rightarrow \mathbb{R}$ be latent sharp image, $y : \mathbb{Z}^2 \rightarrow \mathbb{R}$ is the observed blurry image, and $\mathbf{n} \in \mathbb{Z}^2$ is the pixel location index. Then the observation y is assumed to be given by:

$$y(\mathbf{n}) = \{x \star h_{\mathbf{n}}\}(\mathbf{n}) + \epsilon(\mathbf{n}) \quad (1)$$

where $\epsilon : \mathbb{Z}^2 \rightarrow \mathbb{R}$ is measurement noise. The point spread function $h_{\mathbf{n}} : \mathbb{Z}^2 \rightarrow \mathbb{R}$ denotes a (possibly local) blur kernel acting at pixel location \mathbf{n} , which may not be known *a priori*. However, $h_{\mathbf{n}}$ may take a parametric form in the case of motion blur (by object speed and direction) or defocus blur (by aperture radius or depth). In order for the convolution model of (1) to hold, the Lambertian reflectance assumption is necessary since objects may be observed from a different angle (e.g. as the camera or objects move). Although the degree of deviation of the non-Lambertian reflectance from the model of (1) depends on the properties of surface material (such as Fresnel constant), it is a common practice in the blur/deblur literature to approximate the real world reflectance with (1) (as we do as well in this paper). Where understood, the subscript \mathbf{n} is omitted from $h_{\mathbf{n}}(\mathbf{n})$.

When DDWT is applied to the observation y , the corresponding DDWT coefficients v^{ij} is related to the latent sharp image x and the blur kernel h by:

$$v^{ij}(\mathbf{n}) = \{d^i \star d^j \star y\}(\mathbf{n}) \quad (2)$$

$$\begin{aligned} &= \{d^i \star h\} \star \{d^j \star x\}(\mathbf{n}) + \{d^i \star d^j \star \epsilon\}(\mathbf{n}) \\ &= \{q^i \star u^j\}(\mathbf{n}) + \eta^{ij}(\mathbf{n}), \end{aligned} \quad (3)$$

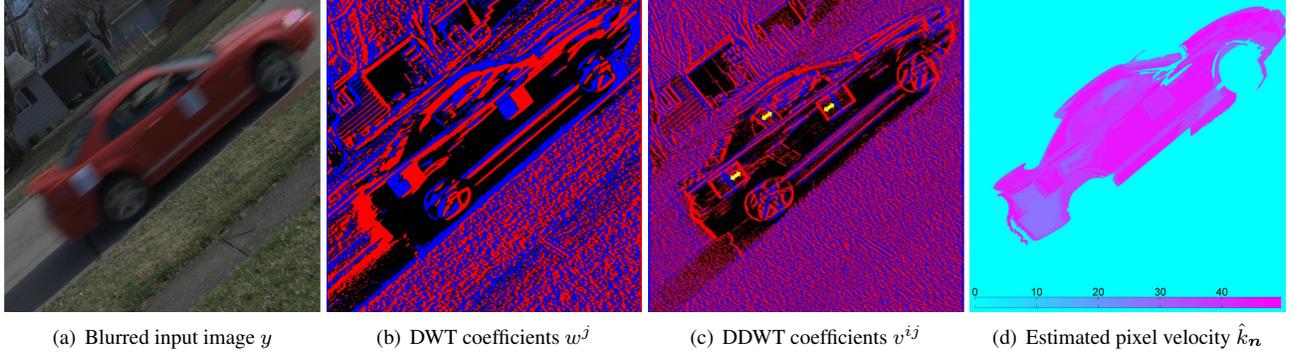


Figure 3. Example of DWT and DDWT coefficients using real camera sensor data. The motion blur manifests itself as a double edge in DDWT, where the distance between the double edges (yellow arrows in (c)) correspond to the speed of the object. Average velocity of the moving pixels in (d) is 38 pixels, and the direction of motion is 40 degrees above the horizontal.

where $u^j := d^j \star x$ and $q^i := d^i \star h$ are the DWT decompositions of x and h , respectively; and $\eta^{ij} := d^i \star d^j \star h$ is noise in the DDWT domain. The relation between (2) and (3) is also illustrated in Figure 2. By the commutativity and associativity of convolution, the processes in 2(a) and 2(b) are equivalent—2(a) is the direct result of applying DDWT on the observed blurry image,¹ but 2(b) is the interpretation we give to the DDWT coefficients (though 2(b) is not directly computable).

Suppose that u^j and q^i are sufficiently sparse. Then DDWT coefficients v^{ij} is the result of applying a “sparse filter” q^i to a “sparse signal” u^j . For a filter q^i supported on $\mathbf{n} \in \{\mathbf{n}_1, \dots, \mathbf{n}_K\}$, we have

$$v^{ij}(\mathbf{n}) = \sum_{k=1}^K q^i(\mathbf{n}_k) u^j(\mathbf{n} - \mathbf{n}_k). \quad (4)$$

When K is small, v^{ij} is nothing more than a sum of a K DWT coefficients u^j . Thanks to sparsity, many of u^j are already zeros, and so v^{ij} is actually a sum of only a few (far less than K) DWT coefficients u^j . In this paper, we call a DDWT coefficient *aliased* if v^{ij} is a sum of more than one “active” u^j coefficients. One can reduce the risk of aliasing when the choice of d^j and d^i makes u^j and q^i as sparse as possible. By symmetry, one may also interpret DDWT coefficients as $v^{ij} = \{q^j \star u^i\} + \eta^{ij}$ —this is equally valid. But in practice, the “confusion” between (q^i, u^j) and (q^j, u^i) does not seem to be a concern for algorithm development when q^i is more sparse than q^j .

Recovery of u^j from v^{ij} leads to image deblurring, while reconstructing q^i is the blur kernel detection problem. Clearly, it is easy to decouple u^j and q^i if v^{ij} is unaliased, and reasonably uncomplicated when u^j and q^i are sufficiently sparse. In the subsequent sections, we demonstrate

¹Images captured by color image sensor undergoes demosaicking, which makes (1) void. One can circumvent this problem by using demosaicking method in [13] designed to recover DWT coefficients w^j from color filter array data directly.

the power of DDWT by analyzing specific blur types and designing example applications. Owing to page limit, we describe object motion blur processing at length. DDWT treatment of other blur types are brief, but their details follow the examples of object motion processing closely. Where understood, the super scripts i and j are omitted from v^{ij} , w^j , u^j , and q^i .

4. Object Motion Blur

4.1. DDWT Analysis

Consider for the moment the horizontal motion blur kernel. Assuming constant velocity of the object during exposure, the blur kernel can be modeled as:

$$h(\mathbf{n}) = \frac{\text{step}(\mathbf{n} + (\frac{0}{k/2})) - \text{step}(\mathbf{n} - (\frac{0}{k/2}))}{k}, \quad (5)$$

where k is the speed of the object. Letting d^i denote a Haar wavelet transform $[-1, 1]$, the DWT coefficient q^i is just a difference of two impulse functions:

$$q(\mathbf{n}) = \frac{\delta(\mathbf{n} + (\frac{0}{k/2})) - \delta(\mathbf{n} - (\frac{0}{k/2}))}{k}. \quad (6)$$

Hence $\{q \star u\}$ is a “difference of two DWT coefficients” placed k pixels apart:

$$\{q \star u\}(\mathbf{n}) = \frac{u(\mathbf{n} + (\frac{0}{k/2})) - u(\mathbf{n} - (\frac{0}{k/2}))}{k}. \quad (7)$$

Figure 3 shows an example of DDWT coefficients. Recall that DWT coefficients u typically captures directional image features (say vertical edges). By (7), we intuitively expect DDWT of moving objects to yield “double edges” where the distance between the two edges² correspond exactly to the speed of the moving object. Indeed, detection

²Red lines denote positive DDWT coefficients while blue lines are negative.

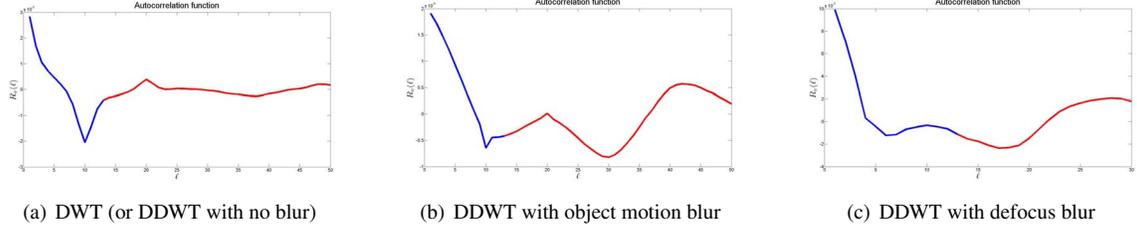


Figure 4. Autocorrelation examples. The minimum coincides with the pixel speed/blur radius we seek. Red lines denote $\ell \geq L$ or $s \geq S$.

of local object speed simplifies to the task of these detecting double edges. Deblurring is equally straightforward: double edges in DDWT coefficients v are the copies of u .

4.2. Object Motion Detection

Naturally, human eye is well suited for the task of identifying replicated DWT coefficients u in DDWT coefficients v —in military applications, for example, a human “analyst” can easily detect complex motion (such as rotation) when presented with a picture of DDWT coefficients (such as Figure 3(c)). For *computational* object motion detection, our ability to detect complex motion is limited primarily by the capabilities of the computer vision algorithms to extract local image features from DDWT, and find similar features that is located unknown (k pixel) distance away. There are a number of ways that this can be accomplished—we emphasize that the correlation-based technique we present below should not be taken as the *best* way, but rather a proof-of-concept on DDWT. A more advanced image feature extraction strategy founded on computer vision principles would likely improve the overall performance of the object motion detection—we leave this as future research plan.

Assume that u in (7) is wide sense stationary. A typical autocorrelation function $R_u(\ell) := \mathbb{E}u(\mathbf{n} - (\ell/2))u(\mathbf{n} + (\ell/2))$ of u is shown in Figure 4(a). When $v(\mathbf{n}) = u(\mathbf{n}) + \eta(\mathbf{n})$ (i.e. no blur),

$$R_v(\ell) = R_u(\ell) + R_\eta(\ell). \quad (8)$$

On the other hand, in the presence of motion blur, $R_v(\ell)$ is a linear combination of $R_u(\ell)$ and $R_\eta(\ell)$, hence $R_v(\ell) =$

$$(2R_u(\ell) - R_u(\ell - k) - R_u(\ell + k))k^{-2} + R_\eta(\ell). \quad (9)$$

As illustrated by Figure 4(b), the maximum of $R_v(\ell) \approx \sigma_u^2/k^2 + \sigma_\eta^2$ occurs at $\ell = 0$. The two minimums of $R_v(\ell)$ coincide with the minimum of $R_u(\ell)$ and with $\ell = \pm k$ caused by the $R_u(\ell \pm k)$. Hence the candidate ℓ which produces smallest secondary autocorrelation yields the estimation of the blur kernel length:

$$\hat{k} = \arg \min_{\ell \geq L} R_v(\ell) \quad (10)$$

where $\ell \in [L, \infty)$ is the candidate searching range (because the first minimum is expected to live in $\ell \in [0, L)$). The

autocorrelation function of (9) is essentially the indicator function for the double edges evidenced in Figure 3(c).

To estimate the local object motion, autocorrelation needs to be localized also. Expectation operator in (8) can be approximated by an local weighted average $R_v(\mathbf{n}, \ell) \approx$

$$\frac{\sum_{\mathbf{m} \in \Lambda} a(\mathbf{n} + \mathbf{m}, \ell)v(\mathbf{n} + \mathbf{m} + (\ell/2))v(\mathbf{n} + \mathbf{m} - (\ell/2))}{\sum_{\mathbf{m} \in \Lambda} a(\mathbf{n} + \mathbf{m}, \ell)} \quad (11)$$

where Λ defines the local neighborhood, \mathbf{n} is the center pixel location, and $a(\mathbf{n}, \ell)$ denotes the averaging weight at location \mathbf{n} . Drawing on the principles of bilateral filtering, weights $a(\mathbf{n}, \ell)$ promote averaging of $v(\mathbf{n} + \mathbf{m} + (\ell/2))v(\mathbf{n} + \mathbf{m} - (\ell/2))$ when $y(\mathbf{n} + \mathbf{m} + (\ell/2))$ and $y(\mathbf{n} + \mathbf{m} - (\ell/2))$ are similar to $y(\mathbf{n})$; and limit contributions of the DDWT coefficient unlikely to be associated with the object at \mathbf{n} . Borrowing the idea of *image simplification*, the bilateral filtering on $R_v(\mathbf{n}, \ell)$ can be repeated multiple times to yield a smoothed $R_v(\mathbf{n}, \ell)$ that favors piecewise-constant object motion speed over more complex ones.

For non-horizontal/vertical motion, we use image shearing to skew the input image y by angle $\phi \in [0, \pi]$ (compared to image rotation, shearing avoids interpolation error). Denoting by $R_v(\mathbf{n}, \phi, \ell)$ the autocorrelation function of $v(\mathbf{n})$ in the sheared direction ϕ , we detect the local blur angle θ and length k by:

$$(\hat{\theta}_n, \hat{k}_n) = \arg \min_{(\phi, \ell) \in [0, \pi] \times [L, \infty)} R_v(\mathbf{n}, \phi, \ell). \quad (12)$$

Figure 3(d) shows the result of estimating the angle θ and the length k of the blur at every pixel location, corresponding to the input image Figure 3(a).

4.3. Object Motion Deblurring

Complementary to the detection of q^i from v^{ij} is the notion of recovering u^j from v^{ij} . When inverse DWT is applied to the recovered u^j , the reconstructed image is the latent sharp image x (i.e. deblurred image). In the discussion below, we assume that the motion blur angle θ and length k are already known via (12). We continue to assume horizontal blur—non-horizontal follows from image shearing.

Recall the relation in (3) and (7). We first note that noise η^{ij} can be accounted for by applying one of many standard wavelet shrinkage operators to DDWT coefficients v^{ij} [14].



Figure 5. Result of object motion deblurring using real camera sensor data with (top row) global and (bottom row) spatially varying blurs. Since methods in [5, 20, 22] cannot handle non-global blur, top row allows for a fair comparison of the reconstruction quality, while the bottom row shows a more realistic scenario. The bottom row was rendered with average velocity of moving pixels for [5, 20, 22] and using Figure 3(d) for the proposed deblurring method.

Such procedure effectively removes noise η^{ij} in v^{ij} to yield a robust estimate $\hat{v} \approx \{q^i \star u^j\}(\mathbf{n}) = k^{-1} \{u(\mathbf{n} + \binom{0}{k/2}) - u(\mathbf{n} - \binom{0}{k/2})\}$ for low and moderate noise. Hence the main deblurring task is the estimation of $u(\mathbf{n})$ from $\hat{v}(\mathbf{n})$.

Key insight we exploit for deblurring is that denoised DDWT coefficients $\hat{v}(\mathbf{n} + \binom{0}{k/2})$ and $\hat{v}(\mathbf{n} - \binom{0}{k/2})$ share the same DWT coefficient $u(\mathbf{n})$. But $u(\mathbf{n})$ in $\hat{v}(\mathbf{n} + \binom{0}{k/2})$ and $\hat{v}(\mathbf{n} - \binom{0}{k/2})$ may be contaminated by $u(\mathbf{n} + \binom{0}{k})$ and $u(\mathbf{n} - \binom{0}{k})$, respectively. It follows from the usual DWT arguments that DWT coefficients u of a natural image x are indeed sparse, and thus it is a rare event that contaminants $u(\mathbf{n} + \binom{0}{k})$ and $u(\mathbf{n} - \binom{0}{k})$ are both active at the same time. To this effect, we have the following result.

Claim 1 (Robust Regression). *Let*

$$\hat{v}(\mathbf{n}) = u(\mathbf{n} + \binom{0}{k/2}) - u(\mathbf{n} - \binom{0}{k/2}).$$

Suppose further that the probability density function of u is symmetric (with zero mean), and $P[u(\mathbf{n}) = 0] = \rho$ (u is said to be “ ρ -sparse”). Then

$$P\left[u(\mathbf{n} + k) = 0 \mid \|\hat{v}(\mathbf{n} + \binom{0}{k/2})\| < \|\hat{v}(\mathbf{n} - \binom{0}{k/2})\|\right] \geq \rho.$$

The proof is provided in the supplementary document. By the above claim, the following reconstruction scheme is “correct” with probability greater than ρ , $\hat{u}^j(\mathbf{n}) =$

$$\begin{cases} k\hat{v}^{ij}(\mathbf{n} + \binom{0}{k/2}) & \text{if } \|\hat{v}(\mathbf{n} + \binom{0}{k/2})\| < \|\hat{v}(\mathbf{n} - \binom{0}{k/2})\| \\ k\hat{v}^{ij}(\mathbf{n} - \binom{0}{k/2}) & \text{otherwise.} \end{cases} \quad (13)$$

Since this deblurring scheme improves if $P[u(\mathbf{n}) = 0] = \rho \approx 1$ (i.e. more sparse), the choice of sparsifying transform d^j is the determining factor for the effectiveness of the proposed DDWT-based blur processing.

We highlight a few notable features of the proposed deblurring scheme. First, the recovery of u^j in (13) is simple, and works regardless of whether the blur kernel is global or spatially varying (simply replace k with $k_{\mathbf{n}}$). Second, the deblurring technique in (13) is a single-pass method. Contrast this to the overwhelming majority of existing deconvolution techniques that require iteration [20, 5, 12]. Third, owing to the fact that no DDWT coefficient $v(\mathbf{n})$ can influence the reconstruction of the DWT coefficient $u(\mathbf{n})$ that is more than $\binom{0}{k/2}$ pixels away, the proposed method is not prone to ringing artifacts. Finally, one can easily incorporate any wavelet domain denoising scheme into the design of the deblurring algorithm. Reconstructions using real camera sensor data in Figure 5 shows superiority of the proposed DDWT approach.

5. Optical Defocus Blur

In this section, we extend DDWT analysis to optical defocus blur. The support of the defocus blur kernel takes the shape of the aperture opening, which is a circular disk in most typical cameras ($\text{supp}\{h\} = \{\mathbf{n} : \|\mathbf{n}\| \leq r\}$ where r is the radius of the disk). Though $h(\mathbf{n})$ may not be known exactly, consider the following approximation:

$$h(\mathbf{n}) \approx \begin{cases} \frac{1}{\pi r^2} & \text{if } \|\mathbf{n}\| \leq r \\ 0 & \text{otherwise.} \end{cases} \quad (14)$$

Letting d^i denote a Haar wavelet transform $[-1, 1]$, the corresponding sparse blur kernel q^i is (see Figure 6(a))

$$q^i(\mathbf{n}) \approx \begin{cases} \frac{1}{\pi r^2} & \text{if } \|\mathbf{n}\| = r \text{ and } n_2 > 0 \\ \frac{-1}{\pi r^2} & \text{if } \|\mathbf{n}\| = r \text{ and } n_2 < 0 \\ 0 & \text{otherwise.} \end{cases} \quad (15)$$

As it turns out, the crude approximation in (14) is acceptable for DDWT-based blur processing—with the discontinuities at the disk boundary and smoothness of $h(\mathbf{n})$ inside the disk, the sparse blur kernel q^i is similar to (15).

Drawing parallels between (15) and (6), DDWT-based processing of optical defocus blur requires only minor modifications to (12) and (13). For detection, one would re-define autocorrelation function to integrate over the circumference of the circle in (14):

$$R_v(\mathbf{n}, s) = \frac{\mathbb{E} \int_{-\pi/2}^{\pi/2} v(\mathbf{n} - s(\cos(\theta)))v(\mathbf{n} + s(\cos(\theta)))d\theta}{s}.$$

The estimated defocus blur radius is given by

$$\hat{r}_n = \arg \min_{s \geq S} R_v(\mathbf{n}, s),$$

where $s \in [S, \infty)$ is the candidate search range. The modified autocorrelation function is shown in Figure 4(c), where its second minimum corresponds to the detected blur radius. Figure 6(b) shows the estimation of defocus blur radius r_n at every location, corresponding to the input Figure 1(a).

For deblurring, we recover the latent wavelet coefficient u^j by comparing $\hat{v}^{ij}(\mathbf{n} - r(\frac{\cos(\theta)}{\sin(\theta)}))$ and $\hat{v}^{ij}(\mathbf{n} + r(\frac{\cos(\theta)}{\sin(\theta)}))$ —it is an unlikely event that both are aliased. As such,

$$\begin{aligned} \hat{u}^j(\mathbf{n}, \theta) &= \pi r^2 [\beta(\mathbf{n}, \theta) \hat{v}^{ij}(\mathbf{n} - r(\frac{\cos(\theta)}{\sin(\theta)})) \\ &\quad - (1 - \beta(\mathbf{n}, \theta)) \hat{v}^{ij}(\mathbf{n} + r(\frac{\cos(\theta)}{\sin(\theta)}))] \\ \beta(\mathbf{n}, \theta) &= \frac{|\hat{v}^{ij}(\mathbf{n} + r(\frac{\cos(\theta)}{\sin(\theta)}))|}{|\hat{v}^{ij}(\mathbf{n} - r(\frac{\cos(\theta)}{\sin(\theta)}))| + |\hat{v}^{ij}(\mathbf{n} + r(\frac{\cos(\theta)}{\sin(\theta)}))|} \end{aligned}$$

is a possible reconstruction of u based on a pair of DDWT coefficients. Figure 7 shows obtained final deblurring result by marginalizing out θ :

$$\hat{u}^j(\mathbf{n}) = \left(\int_{-\pi/2}^{\pi/2} \hat{u}^j(\mathbf{n}, \theta) d\theta \right) (\pi r)^{-1}. \quad (16)$$

6. Camera Shake Blur

In this section, we develop a non-blind deblurring algorithm for camera shake based on the DDWT framework. Camera shake differs from the object motion and defocus blurs because it is difficult to parameterize the blur kernel. Although various work has shown that DWT sparsify most

“reasonable” functions, certain types of transform seem to be well suited for modeling q^i [9]. Define reverse-ordered sparse filter $q'(\mathbf{n}) = q(-\mathbf{n})$. If DWT successfully decorrelates the blur kernel h and image signal x , then following approximations will hold: $\mathbb{E}q(\mathbf{n})q(\mathbf{m}) \approx 0$, $\forall \mathbf{n} \neq \mathbf{m}$. By this approximation, we have the following property:

$$\begin{aligned} \mathbb{E}[\{q' \star \hat{v}\}(\mathbf{n})|u] &= \mathbb{E} \left[\sum_{\mathbf{m}, \mathbf{m}'} q(\mathbf{m} - \mathbf{n})q(\mathbf{m} - \mathbf{m}')u(\mathbf{m}') \middle| u \right] \\ &= u(\mathbf{n}) \mathbb{E} \left[\sum_{\mathbf{m}} q(\mathbf{m})^2 \right]. \end{aligned}$$

Hence the following is an unbiased estimator for the latent DWT coefficient u :

$$\hat{u}(\mathbf{n}) = (\{q' \star \hat{v}\}(\mathbf{n})) \left(\sum_{\mathbf{m}} q(\mathbf{m})^2 \right)^{-1}. \quad (17)$$

The generalization of (17) to the spatially varying blur kernel case is straightforward (simply replace q^i with q_n^i). On the other hand, if h is a global blur kernel then $y \mapsto v$, $\hat{v} \mapsto \hat{u}$, and $\hat{u} \mapsto \hat{x}$ are a completely linear, shift-invariant processing. When the wavelet denoising step $v \mapsto \hat{v}$ can be ignored, then the entire deblur procedure $y \mapsto v \mapsto \hat{u} \mapsto \hat{x}$ may be reduced to a single convolution operation, which significantly reduces the computational complexity:

$$\hat{x} = \left\{ \sum_j e^j \star \left\{ \frac{q^i}{\sum_{\mathbf{m}} q^i(\mathbf{m})} \right\} \star \{d^i \star d^j\} \right\} \star y, \quad (18)$$

where e^j is the j th synthesis filter corresponding to d^j .

One weakness to the proposed deblurring scheme is the lack of adaptivity to the image content. Unlike (13) or (16), the method in (17) makes no distinction between aliased and non-aliased coefficients \hat{v} . For example, suppose we apply (17) to the problem of object motion blur (instead of camera shake). In this case, deblurring procedure would simply average $v(\mathbf{n} + (\frac{0}{k/2}))$ and $v(\mathbf{n} - (\frac{0}{k/2}))$ instead of choosing the one with far lower risk of aliasing (*a la* (13)). Deblurred image is likely to suffer from minor ringing artifacts near edges due to the fact that aliasing was not resolved. Signal-adaptive reconstruction in DDWT domain will be addressed in our future work—the main challenge is that the DWT coefficients q^i of blur kernel is more difficult to parameterize than the object motion and defocus ones. Nevertheless, (17) is a simple, non-iterative deblurring method yielding reasonable output image quality, and it clearly validates the overall DDWT blur processing framework.

7. Image Recognition

Blur interferes with recognition tasks, as feature extraction from blurry image is a real challenge. For example, a license plate shown in Figure 9(a) is blurred by the motion

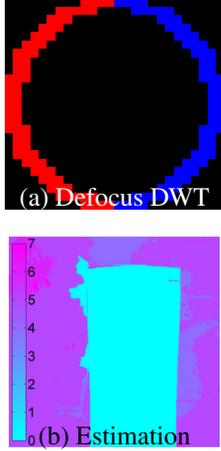


Figure 6. (a) is the DWT of defocus blur kernel. Distance between double edges is 9. (b) is defocus diameter estimation of figure 1(a) with average diameter of 7.



Figure 7. Result of optical defocus deblurring using real camera sensor data with (top row) global and (bottom row) varying depth. Although methods in [5, 20] cannot handle non-global blur, top row is a fair comparison of the reconstruction quality, while the bottom row shows a more realistic scenario. The bottom row was rendered with background blur for [5, 20] and using Figure 6(b) for the proposed deblurring method.



Figure 8. Result of camera shake deblurring using synthetic data.

of the car. As evidenced by Figure 9(b), edge detection fails to yield meaningful features because the image lacks sharp transitions. Character recognition on Figure 9(b) would also likely fail, not only due to degraded image quality but also because letters are elongated in the horizontal direction by an arbitrary amount. One obvious way to cope with this is to deblur as a pre-processing step to the computer vision algorithms. Analysis in Section 3 suggests an alternative approach of extracting near-blur-invariant image features.

Specifically, consider DWT and DDWT of input image y , shown in Figures 9(c-d). Though the conventional interpretation of Figure 9(c) is that this is a wavelet decomposition of an *image*, one can equally understand this as a sparse

filter applied to a sharp image, as follows:

$$w^j(\mathbf{n}) := \{q^j \star x\}(\mathbf{n}) + \{d^j \star \epsilon\}(\mathbf{n}).$$

If Haar wavelet transform $[-1, 1]$ is used, above reduces to a difference of latent sharp image x :

$$w^j(\mathbf{n}) := x(\mathbf{n} + \binom{0}{k/2}) - x(\mathbf{n} - \binom{0}{k/2}) + \{d^j \star \epsilon\}(\mathbf{n}).$$

Hence the characteristics of the latent image x is well preserved in DWT coefficients w^j . Indeed, characters in Figure 9(c) are more readable—their appearance is sharp with strong edges, and they have not been elongated in the direction of the motion. However, each character appears twice (two 7's, etc.), so a character recognition on w^j would require a post-processing step to prevent double counting of



Figure 9. Example of license plate identification using DDWT using real camera sensor data.

detected characters. DDWT shown in Figure 9(d) essentially performs edge detection on Figure 9(c). Compared to the failed edge detection of Figure 9(b), this is clearly an improvement. Note also that we were never required to detect the blur kernel for producing Figures 9(c-d).

The above analysis raises the possibility of carrying out recognition tasks on blurry images without image deblurring. DWT and DDWT are near-blur-invariant representations of the latent image x , and computer vision algorithms can be trained to work with them directly.

8. Conclusions

We proposed *double discrete wavelet transform*—a novel analytical tool for blur processing. DDWT sparsifies the latent sharp image and blur kernel simultaneously using DWT. Sparse representation is key to decoupling blur and image signals, enabling blur kernel recovery and deblurring to occur in the wavelet domain. This framework also inspires a new generation of blur-tolerant recognition tasks aimed at exploiting the near-blur-invariant properties of DDWT coefficients. We validated the power of DDWT framework via example applications and experiments using real camera sensor data, but further development in blur kernel detection, deblurring, and recognition tasks are possible. Potential applications of DDWT include object velocity and defocus blur estimation, which are useful for making inferences on the object activities or the depths.

Acknowledgments We thank the authors of [3, 4, 5, 22, 24] for providing their code. This work was funded in part by Texas Instrument and University of Dayton Graduate School Summer Fellowship program.

References

- [1] L. Bar, B. Berkels, M. Rumpf, and G. Sapiro. A variational framework for simultaneous motion estimation and restoration of motion-blurred video. *ICCV*, 2007.
- [2] M. Ben-Ezra and S. Nayar. Motion deblurring using hybrid imaging. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 1, pages 1–657. IEEE, 2003.
- [3] J. Cai, H. Ji, and Z. Shen. Blind motion deblurring from a single image using sparse approximation. *CVPR*, 62:291–294, January 2009.
- [4] A. Chakrabarti, T. Zickler, and W. T. Freeman. Analyzing spatially-varying blur. *CVPR*, 2010.
- [5] S. Chan, R. Khoshabeh, K. Gibson, P. Gill, and T. Nguyen. An augmented Lagrangian method for total variation video restoration. *Image Processing, IEEE Transactions on*, 20(11):3097–3111, 2011.
- [6] S. Chan and T. Nguyen. Single image spatially-variant out-of-focus blur removal. *ICIP*, 2011.
- [7] J. Chen, L. Yuan, C. Tang, and L. Quan. Robust dual motion deblurring. In *Computer Vision and Pattern Recognition (CVPR), 2008 IEEE Conference on*, pages 1–8. IEEE, 2008.
- [8] T. Cho, A. Levin, F. Durand, and W. Freeman. Motion blur removal with orthogonal parabolic exposures. In *Computational Photography (ICCP), 2010 IEEE International Conference on*, pages 1–8. IEEE, 2010.
- [9] T. Cho, S. Paris, B. Horn, and W. Freeman. Blur kernel estimation using the radon transform. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 241–248. IEEE, 2011.
- [10] S. Dai and Y. Wu. Motion from blur. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.
- [11] D. Donoho and M. Raimondo. A fast wavelet algorithm for image deblurring. *ANZIAM Journal*, 46:C29–C46, 2005.
- [12] R. Fergus, B. Singh, A. Hertzmann, S. Roweis, and W. Freeman. Removing camera shake from a single photograph. *ACM SIGGRAPH*, 2006.
- [13] K. Hirakawa, X. Meng, and P. Wolfe. A framework for wavelet-based analysis and processing of color filter array images with applications to denoising and demosaicing. In *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, volume 1, pages 1–597. IEEE, 2007.
- [14] K. Hirakawa and P. Wolfe. Skellam shrinkage: Wavelet-based intensity estimation for inhomogeneous poisson data. *Information Theory, IEEE Transactions on*, 58(2):1080–1093, 2012.
- [15] M. Hirsch, C. J. Schuler, S. Harmeling, and B. Scholkopf. Fast removal of non-uniform camera shake. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 463–470. IEEE, 2011.
- [16] J. Jia. Single image motion deblurring using transparency. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.
- [17] A. Levin. Blind motion deblurring using image statistics. *Advances in Neural Information Processing Systems*, 19:841, 2007.
- [18] A. Levin, P. Sand, T. Cho, F. Durand, and W. Freeman. Motion-invariant photography. In *ACM Transactions on Graphics (TOG)*, volume 27, page 71. ACM, 2008.
- [19] A. Levin, Y. Weiss, F. Durand, and W. Freeman. Understanding and evaluating blind deconvolution algorithms. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1964–1971. IEEE, 2009.
- [20] L. Lucy. Bayesian-based iterative method of image restoration. *Journal of Astronomy*, 79(745-754), 1974.
- [21] S. Nayar and M. Ben-Ezra. Motion-based motion deblurring. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(6):689–698, 2004.
- [22] Q. Shan, J. Jia, and A. Agarwala. High-quality motion deblurring from a single image. *ACM Transactions on Graphics (SIGGRAPH)*, 2008.
- [23] M. Subbarao and G. Surya. Depth from defocus: a spatial domain approach. *International Journal of Computer Vision*, 13(3):271–294, 1994.
- [24] O. Whyte, J. Sivic, A. Zisserman, and J. Ponce. Non-uniform deblurring for shaken images. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 491–498. IEEE, 2010.