ANZIAM J. 58(2017), 247–255
doi:10.1017/S1446181117000219

IMAGE INPAINTING FROM PARTIAL NOISY DATA BY DIRECTIONAL COMPLEX TIGHT FRAMELETS

YI SHEN✉[1,2,3], BIN HAN[2] and ELENA BRAVERMAN[3]

(Received 12 June, 2016; accepted 15 October, 2016; first published online 26 May 2017)

Abstract

Image inpainting methods recover true images from partial noisy observations. Natural images usually have two layers consisting of cartoons and textures. Methods using simultaneous cartoon and texture inpainting are popular in the literature by using two combined tight frames: one (often built from wavelets, curvelets or shearlets) provides sparse representations for cartoons and the other (often built from discrete cosine transforms) offers sparse approximation for textures. Inspired by the recent development on directional tensor product complex tight framelets (TP-$\mathbb{C}$TFs) and their impressive performance for the image denoising problem, we propose an iterative thresholding algorithm using tight frames derived from TP-$\mathbb{C}$TFs for the image inpainting problem. The tight frame TP-$\mathbb{C}\mathrm{TF}_6$ contains two classes of framelets; one is good for cartoons and the other is good for textures. Therefore, it can handle both the cartoons and the textures well. For the image inpainting problem with additive zero-mean independent and identically distributed Gaussian noise, our proposed algorithm does not require us to tune parameters manually for reasonably good performance. Experimental results show that our proposed algorithm performs comparatively better than several well-known frame systems for the image inpainting problem.

2010 *Mathematics subject classification*: primary 42C40; secondary 42C15, 65T60, 94A08.

Keywords and phrases: noisy data, image inpainting, directional tensor product complex tight framelets, sparse representation, iterative scheme.

1. Introduction and motivations

The image inpainting problem occurs when the pixels of an image are missing or corrupted by various types of noise during image acquisition, storage, or transmission.

[1]Department of Mathematics, Zhejiang Sci-Tech University, Hangzhou 310028, China; e-mail: yshen@zstu.edu.cn.
[2]Department of Mathematical and Statistical Sciences, University of Alberta, Edmonton, Alberta T6G 2G1, Canada; e-mail: bhan@ualberta.ca.
[3]Department of Mathematics and Statistics, University of Calgary, 2500 University Drive N.W., Calgary, Alberta T2N 1N4, Canada; e-mail: maelena@ucalgary.ca.

Let $\Omega \subseteq \{1, \dots, d\}$ be a nonempty given observable region. Define a $d \times d$ diagonal matrix $\mathcal{P}_\Omega$ by

$$[\mathcal{P}_\Omega]_{j,k} = \begin{cases} 1, & j = k \text{ with } j \in \Omega, \\ 0, & j = k \text{ with } j \notin \Omega \text{ or } j \neq k, \end{cases} \qquad j, k = 1, \dots, d. \tag{1.1}$$

Image inpainting is generally formulated as the following: the observed image $\boldsymbol{y} = (y_1, \dots, y_d)^\mathsf{T} \in \mathbb{R}^d$ is given by

$$\boldsymbol{y} = \mathcal{P}_\Omega \boldsymbol{x} + \boldsymbol{n}, \tag{1.2}$$

where $\boldsymbol{x} = (x_1, \dots, x_d)^\mathsf{T} \in \mathbb{R}^d$ is an unknown clean image to be restored and $\boldsymbol{n}$ is the additive zero-mean independent and identically distributed Gaussian noise. Quite often, the observation $\boldsymbol{y}$ in (1.2) is only available on a subset Ω of $\{1, \dots, d\}$, while $y_j, j \notin \Omega$, are not available.

The goal of image inpainting is to recover the pixels of $\boldsymbol{x}$ on Ω by suppressing the noise of $\boldsymbol{y}$ in the observable region Ω. Methods for image inpainting can be classified into three groups: patch based, partial differential equation/variational based, and sparse representation based methods. The reader may see a detailed list of references for many different methods to study the image inpainting problem in the literature [1, 6, 7]. Among those image inpainting methods, one popular approach is to employ sparse representations and convex minimization schemes with regularization (see [2–4, 6–8, 12, 13] and the references therein). Roughly speaking, one expects that the unknown clean image has a sparse representation under a basis, or a frame, or more generally a dictionary. Then one hopes to recover the unknown image $\boldsymbol{x}$ by finding the few dominating large coefficients of $\boldsymbol{x}$ in the transform domain from (1.2) by minimization schemes with some sparsity constraints. Due to the energy-preserving property and computational efficiency, orthonormal bases and tight frames are often used. For example, the orthonormal basis in the discrete cosine transform (DCT) has many applications in image processing and has been known to be effective for sparsely approximating the texture part of an image. Tight frames built from wavelets (called tight framelets in this paper) or from curvelets/shearlets are claimed to provide sparse approximations for natural images and they are particularly attractive for capturing the cartoon part of an image [2, 5, 9, 11, 13].

Inspired by the recent development on directional tensor product complex tight framelets (TP-$\mathbb{C}$TF) and their impressive performance for the image denoising problem in [10, 11], we use two tight frames derived from TP-$\mathbb{C}$TFs for the image inpainting problem. One is a single tight frame system which can handle both the cartoon part and texture part well. The other tight frame, which can represent the cartoon part well, is suggested to be used together with local DCT.

Before going further, we introduce some notation. The inner product of two vectors $\boldsymbol{x}, \boldsymbol{y}$ is defined to be $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \sum_{j=1}^{d} x_j y_j$. The l_2-norm of $\boldsymbol{x}$ is defined by $\|\boldsymbol{x}\|_2 = \sqrt{\langle \boldsymbol{x}, \boldsymbol{x} \rangle}$. If a $d \times n$ matrix $\mathcal{D}$ is a tight frame, that is, $\mathcal{D}\mathcal{D}^\mathsf{T} = I$,

$$\boldsymbol{x} = \mathcal{D}\mathcal{D}^\mathsf{T}\boldsymbol{x} = \mathcal{D}\boldsymbol{c} = \sum_{j=1}^{n} c_j \mathcal{D}_j \quad \text{with } \boldsymbol{c} = (c_1, \dots, c_n)^\mathsf{T} = \mathcal{D}^\mathsf{T}\boldsymbol{x}.$$

https://doi.org/10.1017/S1446181117000219 Published online by Cambridge University Press

Here $c_j = \langle \boldsymbol{x}, \mathcal{D}_j \rangle$ are called the frame coefficients. An image $\boldsymbol{x}$ often has sparse frame coefficients under some tight frames. The bivariate shrinkage function was introduced by Sendur and Selesnick [14]. To take advantage of the cross-scale relations in the wavelet tree of frame coefficients, we use the bivariate shrinkage function η_λ^{bs} as follows:

$$\eta_\lambda^{bs}(c) = \eta_{\lambda_c}^{\text{soft}}(c) = \begin{cases} c - \lambda_c c/|c|, & |c| > \lambda_c, \\ 0, & \text{otherwise} \end{cases} \quad \text{with } \lambda_c = \frac{\sqrt{3}\sigma_n^2}{\sigma_c \sqrt{1 + |c_p/c|^2}}, \tag{1.3}$$

where $\sigma_n = \lambda \|b\|_2$ with b being the high-pass filter inducing the frame coefficient c, the frame coefficient c_p is the parent coefficient of c in the immediate higher scale, and

$$\sigma_c = \begin{cases} \sqrt{\breve{\sigma}_c^2 - \sigma_n^2}, & \breve{\sigma}_c > \sigma_n, \\ 0, & \text{otherwise} \end{cases} \quad \text{with } \breve{\sigma}_c^2 = \frac{1}{|N_c|} \sum_{j \in N_c} |c_j|^2,$$

where $|N_c|$ is the cardinality of the set N_c (details of the choice of N_c can be found in [14]).

2. Image inpainting algorithm using TP-$\mathbb{C}$TF$_6$

We now provide full details in Algorithm 1 for our proposed image inpainting algorithm. The output $\boldsymbol{x}_{\ell+1}$ of Algorithm 1 is the restored (or inpainted) image. The projection operator $\mathcal{P}_\Omega$ and the thresholding η_λ^{bs} are defined in (1.1) and (1.3), respectively. Note that r is the percentage of missing pixels, that is, the ratio between the number of missing pixels in the inpainting mask and the number of all pixels in an image. We now discuss how to generate the thresholding values Λ_1 and Λ_2. Algorithm 1 uses decreasing thresholding values $\Lambda_1 \cup \Lambda_2$ from $[\lambda_{\min}, \lambda_{\text{mid}}] \cup [\lambda_{\text{mid}}, \lambda_{\max}]$, where we set

$$\lambda_{\min} = \max\{1, \sigma(1 - r^2/2)\}, \quad \lambda_{\max} = 512, \tag{2.1}$$

and

$$\lambda_{\text{mid}} = \min\{\max\{2\lambda_{\min} + 10, 20\}, \lambda_{\max}\}.$$

The sequence Λ_1 of decreasing thresholding values on $[\lambda_{\text{mid}}, \lambda_{\max}]$ and the sequence Λ_2 of decreasing thresholding values on $[\lambda_{\min}, \lambda_{\text{mid}}]$ are given by

$$\begin{aligned} \Lambda_1(i) &= r_1^{(i-N_1)/(N_1-1)} \lambda_{\text{mid}}, \quad i = 1, \ldots, N_1, \\ \Lambda_2(i) &= r_2^{(i-N_2)/N_2} \lambda_{\min}, \quad i = 1, \ldots, N_2, \end{aligned} \tag{2.2}$$

where $r_1 = \lambda_{\text{mid}}/\lambda_{\max}$ and $r_2 = \lambda_{\min}/\lambda_{\text{mid}}$. Note that $0 < r_1, r_2 < 1$ and

$$\begin{aligned} \Lambda_1(1) = \lambda_{\max}, \quad \Lambda_1(N_1) = \lambda_{\text{mid}}, \\ \Lambda_2(1) = r_2^{1/N_2} \lambda_{\text{mid}}, \quad \Lambda_2(N_2) = \lambda_{\min}. \end{aligned}$$

Consider the image inpainting model (1.2). Let n_j be the additive independent and identically distributed zero-mean Gaussian noise with zero mean and standard

https://doi.org/10.1017/S1446181117000219 Published online by Cambridge University Press

Algorithm 1 Frame-based image inpainting algorithm using TP-$\mathbb{C}$TF$_6$

Require: The tight frame $\mathcal{D} \in \mathbb{R}^{d\times n}$ built from TP-$\mathbb{C}$TF$_6$, an inpainting mask $\{1,\dots,d\}\backslash\Omega$ (that is, Ω is a given observable region), standard deviation σ of independent and identically distributed zero-mean Gaussian noise, and an observed partial image $\boldsymbol{y} \in \mathbb{R}^d$ on the observable region Ω.

Initialization: $\boldsymbol{x}_{-1} = \boldsymbol{x}_0 = 0$, $i = \ell = 1$, $r = 1 - (\#\Omega)/n$, where $\#\Omega$ denotes the cardinality of Ω. Generate thresholding values Λ_1, Λ_2 by (2.2) and iteration parameters $N_1, N_2, \mathrm{tol}_1, \mathrm{tol}_2$ by Table 2.

Initialize thresholding value: $\lambda = \Lambda_1(1)$.

while $(i \leqslant N_1 + N_2)$ **do**

  $\boldsymbol{y}_\ell = \mathcal{P}_\Omega(\boldsymbol{y}) + (I - \mathcal{P}_\Omega)(\boldsymbol{x}_\ell)$.

  $\boldsymbol{c}_{\ell+1} = \eta_\lambda^{bs}(\mathcal{D}^\mathsf{T}\boldsymbol{y}_\ell)$.

  $\boldsymbol{x}_{\ell+1} = \mathcal{D}\boldsymbol{c}_{\ell+1}$.

  $\mathrm{error} = \|(I - \mathcal{P}_\Omega)(\boldsymbol{x}_{\ell+1} - \boldsymbol{x}_\ell)\|_2 / \|\mathcal{P}_\Omega\boldsymbol{y}\|_2$.

  if $(\mathrm{error} < \mathrm{tol}_1)$ **and** $(i < N_1)$ **then**

    $i = i + 1$.

    $\lambda = \Lambda_1(i)$.

  else if $(\mathrm{error} < \mathrm{tol}_2)$ **and** $(N_1 \leqslant i < N_1 + N_2)$ **then**

    $i = i + 1$.

    $\lambda = \Lambda_2(i - N_1)$.

  else if $(\mathrm{error} < \mathrm{tol}_2)$ **and** $(i = N_1 + N_2)$ **then**

    Break.

  end if

  $\ell = \ell + 1$.

end while

return $\boldsymbol{x}_{\ell+1}$ as the restored image.

---

deviation σ. Suppose that the pixels of the image are missing uniformly and randomly with probability r. For any given tight frame, the noise standard deviation of each frame coefficient (after normalization of the filters) is approximately $\sigma\sqrt{1-r}$. The boundary of the inpainting mask often creates artificial jumps between the observable region and the missing region. Hence, the boundary of the inpainting mask also contributes as another source of noise. As a consequence, we should set the positive minimal thresholding value $\lambda_{\min}$ greater than $\sigma\sqrt{1-r}$. For simplicity, we take $\lambda_{\min}$ as in (2.1) and note that $\lambda_{\min} \geqslant \sigma\sqrt{1-r}$.

3. Numerical experiments

Real images usually have two components referring to the cartoon part (the piecewise-smooth part of the image) and the texture part (the oscillating pattern part of the image). Both these components have sparse approximations under some tight frame systems. The real part and imaginary part of TP-$\mathbb{C}$TF$_4$ have six edge-like

https://doi.org/10.1017/S1446181117000219 Published online by Cambridge University Press

TABLE 1. Simultaneous cartoon and texture image inpainting using different tight frames with local DCT.

	Text 3	Text 4	50%	80%
TP-$\mathbb{C}$TF$_4$+Local DCT	*35.37*	*31.85*	*33.24*	*26.89*
[3]	34.52	30.88	32.71	26.60
[8]	35.07	31.49	32.76	26.43

TABLE 2. Choices of iteration parameters N_1, N_2 and stopping tolerances $\text{tol}_1, \text{tol}_2$ for Algorithm 1.

r	N_1	tol_1	N_2	tol_2
$0 < r < 0.5$	5	5×10^{-3}	8	10^{-4}
$0.5 \leqslant r < 1$	8	5×10^{-3}	5	10^{-3}

directional elements, which are expected to handle the cartoon part well. To illustrate this property, we test the cartoon and texture inpainting by using the complex tight frame TP-$\mathbb{C}$TF$_4$. To represent the texture part of the image, we use the local DCT, which is a standard transform (see, for example, [3, 8]). We compare the complex tight frame TP-$\mathbb{C}$TF$_4$ with the algorithms of Cai et al. [3] and Elad et al. [8]. Let $\boldsymbol{x}_{t,0} = 0$ and $\boldsymbol{x}_{c,0} = 0$; we use the following iterative scheme:

$$\boldsymbol{x}_{t,\ell} = \mathcal{D}_t \eta_{\lambda_t}^{\text{soft}} \{\mathcal{D}_t^{\top} (\mathcal{P}_\Omega \boldsymbol{y} + (I - \mathcal{P}_\Omega) \boldsymbol{x}_{t,\ell-1} - \mathcal{P}_\Omega \boldsymbol{x}_{c,\ell-1})\},$$
$$\boldsymbol{x}_{c,\ell} = \mathcal{D}_c \eta_{\lambda_c}^{bs} \{\mathcal{D}_c^{\top} (\mathcal{P}_\Omega \boldsymbol{y} + (I - \mathcal{P}_\Omega) \boldsymbol{x}_{c,\ell-1} - \mathcal{P}_\Omega \boldsymbol{x}_{t,\ell-1})\},$$
$$\lambda_t = \rho \lambda_t, \quad \lambda_c = \rho \lambda_c$$

to recover the corrupted image without noise. Here $\mathcal{D}_t$ denotes the local DCT and $\mathcal{D}_c$ denotes the complex tight frame TP-$\mathbb{C}$TF$_4$. In applications, the parameter ρ is set to be 0.95 and $\lambda_t = 255$, $\lambda_c = 2.5\lambda_t$. The iterative scheme stops when $\lambda_c < 1$. For the above-mentioned algorithms [3, 8], we use the codes kindly provided by the authors. We choose the image Barbara of size 512×512 as the test image; the comparison results are given in Table 1.

To test our proposed algorithm, we use four test images and two inpainting masks in Figure 1, which are from http://pan.baidu.com/s/1slwa2Ln. We compare the performance of our proposed algorithm with several frame-based iterative image inpainting algorithms including a spline tight framelet based image inpainting algorithm by Cai et al. [2]; an adaptive inpainting algorithm based on undecimated transform using the DCT–Haar wavelet filters by Li et al. [12]; and an image inpainting algorithm based on undecimated transform using compactly supported nonseparable shearlets by Lim [13]. The implementations of all those related frame-based image inpainting algorithms are kindly provided by their own authors or downloaded from their home pages. We run all the related image inpainting algorithms with their default parameter values, which have been given in the source codes by their respective authors. The parameters of Algorithm 1 are given in Table 2.

Some zoomed-in views of comparison results are shown in Figure 2. We observe that our proposed algorithm produces the restoration results with fewer artefacts than

(a) Hill (b) Man (c) Boat

(d) Barbara (e) Text 1 (f) Text 2

FIGURE 1. (a)–(d) are test images of size 512×512. (e) and (f) are inpainting masks of size 512×512. The observable region Ω is the complement of an inpainting mask.

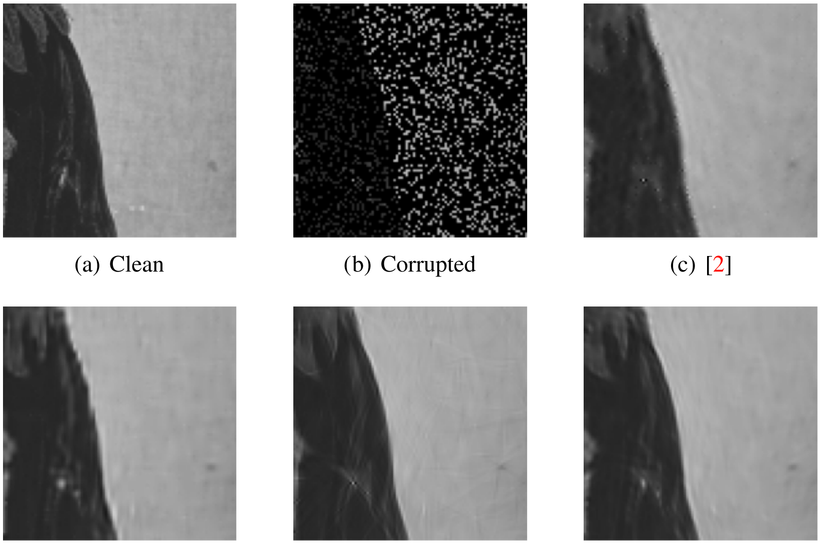
TABLE 3. Performance in terms of PSNR values of several image inpainting algorithms without noise for four 512×512 test images in (a)–(d) of Figure 1. Top left table for inpainting mask Text 3 in (e) of Figure 1. Top right table for inpainting mask Text 4 in (f) of Figure 1. Bottom left table for 50% randomly missing pixels. Bottom right table for 80% randomly missing pixels.

	Hill	Man	Boat	Barbara	Hill	Man	Boat	Barbara
[2]	35.08	34.50	33.10	31.89	31.63	30.70	29.29	29.04
[12]	35.73	34.82	34.62	35.03	32.85	31.34	30.35	31.51
[13]	35.69	35.11	34.66	35.17	32.13	31.52	30.65	32.45
Algorithm 1	*36.03*	*35.52*	*34.94*	*36.59*	*32.54*	*31.92*	*30.76*	*32.65*
[2]	28.90	28.18	27.02	24.32	28.93	28.06	27.03	24.32
[12]	34.43	33.45	34.08	33.85	28.99	27.69	27.87	26.39
[13]	33.11	32.81	33.07	34.13	29.07	28.42	28.01	28.08
Algorithm 1	*34.53*	*34.25*	*34.42*	*35.69*	*29.59*	*29.15*	*28.56*	*28.11*

by other methods. Compared to the peak signal-to-noise ratio (PSNR) presented in Tables 1 and 3, we also notice that TP-$\mathbb{C}$TF$_6$ produces better results than TP-$\mathbb{C}$TF$_4$ combined with local DCT does. One reason is that TP-$\mathbb{C}$TF$_6$ provides a multi-resolution decomposition, while local DCT only provides one-level decomposition.

The inpainting algorithm by Li et al. [12] and the algorithm by Lim [13] only consider the noiseless case. For the algorithms proposed by Cai et al. [2] and the algorithm by Elad et al. [8], the parameters of these iterative algorithms are often

https://doi.org/10.1017/S1446181117000219 Published online by Cambridge University Press



(a) Clean (b) Corrupted (c) [2]

(d) [15] (e) [16] (f) Algorithm 1

FIGURE 2. Zoomed-in portion of the clean image Man in Figure 1 (b) in (a), the corrupted image in (b) by 50% randomly missing pixels without noise, and the inpainted images in (c) by [2], (d) by [12], (e) by [13], and (f) by our Algorithm 1.

TABLE 4. Performance in terms of PSNR values of our proposed image inpainting algorithm under independent and identically distributed zero-mean Gaussian noise with noise standard deviation $\sigma = 10, 30, 50$ for four 512×512 test images in (a)–(d) of Figure 1. Top left table for inpainting mask Text 3 in (e) of Figure 1. Top right table for inpainting mask Text 4 in (f) of Figure 1. Bottom left table for 50% randomly missing pixels. Bottom right table for 80% randomly missing pixels.

σ	Hill	Man	Boat	Barbara	Hill	Man	Boat	Barbara
10	31.45	31.43	31.04	31.81	29.90	29.64	28.77	29.84
30	27.84	27.58	27.41	27.17	27.00	26.66	26.20	26.24
50	26.18	25.77	25.55	24.89	25.56	25.11	24.75	24.28
10	30.70	30.64	30.65	31.10	27.90	27.55	27.08	26.66
30	27.04	26.82	26.64	25.93	25.33	24.92	24.42	23.30
50	25.36	25.00	24.71	23.56	23.95	23.39	22.90	21.85

manually set and depend on the noise levels and test images. Here we only report the performance of our proposed Algorithm 1 for image inpainting with noise. We choose the independent and identically distributed zero-mean Gaussian noise with three different standard deviations $\sigma = 10, 30, 50$. The results are summarized in Table 4. See more numerical experiments and details of TP-$\mathbb{C}$TF$_s$ in the supplementary file [15].

https://doi.org/10.1017/S1446181117000219 Published online by Cambridge University Press

4. Conclusions

We propose a frame-based image inpainting algorithm. Numerical results show that our proposed inpainting algorithm can restore corrupted images with better quality than those recovered by the state-of-the-art frame-based iterative inpainting algorithms [2, 3, 8, 12, 13]. Moreover, our proposed algorithm performs well for image inpainting under noise. In future, we expect that our results on image inpainting could be further improved by exploring the freedom in the design of directional tensor product complex tight framelets [10, 11] or by using directional nonseparable tight framelets [9].

Acknowledgements

The authors thank Zhenpeng Zhao for providing us with his source matlab code implementing the discrete framelet transform using TP-$\mathbb{C}$TF$_6$. This research was supported in part by the Natural Sciences and Engineering Research Council of Canada under grants 05865 and 261351 and a Pacific Institute for the Mathematical Sciences (PIMS) CRG grant. Research of Yi Shen was also supported in part by a PIMS postdoctoral fellowship, the NSF of China under grant 11671358, the NSAF of China under grant U1630116, and the Key Project of NSF of China under grant 11531013.

References

[1] M. Bertalmio, G. Sapiro, V. Caselles and C. Ballester, "Image inpainting", in: *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques: SIGGRAPH'00* (ACM Press/Addison Wesley Publ. Co., New York, 2000) 417–424; doi:10.1145/344779.344972.

[2] J.-F. Cai, R. Chan and Z. W. Shen, "A framelet-based image inpainting algorithm", *Appl. Comput. Harmon. Anal.* **24** (2008) 131–149; doi:10.1016/j.acha.2007.10.002.

[3] J.-F. Cai, R. Chan and Z. W. Shen, "Simultaneous cartoon and texture inpainting", *Inverse Probl. Imaging* **4** (2010) 379–395; doi:10.3934/ipi.2010.4.379.

[4] J.-F. Cai, S. Osher and Z. W. Shen, "Split Bregman methods and frame based image restoration", *Multiscale Model. Simul.* **8** (2009) 337–369; doi:10.1137/090753504.

[5] E. J. Candès and D. L. Donoho, "New tight frames of curvelets and optimal representations of objects with piecewise C^2 singularities", *Comm. Pure Appl. Math.* **57** (2004) 219–266; doi:10.1002/cpa.10116.

[6] T. F. Chan and J. Shen, *Image processing and analysis: variational, PDE, wavelet, and stochastic methods*. 1st edn (SIAM, Philadelphia, PA, 2005).

[7] M. Elad, *Sparse and redundant representations: from theory to applications in signal and image processing*, 1st edn (Springer, New York, 2010).

[8] M. Elad, J. L. Starck, P. Querre and D. L. Donoho, "Simultaneous cartoon and texture image inpainting using morphological component analysis (MCA)", *Appl. Comput. Harmon. Anal.* **19** (2005) 340–358; doi:10.1016/j.acha.2005.03.005.

[9] B. Han, "Nonhomogeneous wavelet systems in high dimensions", *Appl. Comput. Harmon. Anal.* **32** (2012) 169–196; doi:10.1016/j.acha.2011.04.002.

[10] B. Han, "Properties of discrete framelet transforms", *Math. Model. Nat. Phenom.* **8** (2013) 18–47; doi:10.1051/mmnp/20138102.

[11] B. Han and Z. P. Zhao, "Tensor product complex tight framelets with increasing directionality", *SIAM J. Imaging Sci.* **7** (2014) 997–1034; doi:10.1137/130928558.

[12] Y. R. Li, L. X. Shen and B. W. Suter, "Adaptive inpainting algorithm based on DCT induced wavelet regularization", *IEEE Trans. Image Process.* **22** (2013) 752–763; doi:10.1109/TIP.2012.2222896.

[13] W.-Q. Lim, "Nonseparable shearlet transform", *IEEE Trans. Image Process.* **22** (2013) 2056–2065; doi:10.1109/TIP.2013.2244223.

[14] L. Sendur and I. W. Selesnick, "Bivariate shrinkage with local variance estimation", *IEEE Signal Process. Lett.* **9** (2002) 438–441; doi:10.1109/LSP.2002.806054.

[15] Y. Shen, B. Han and E. Braverman, "Image inpainting using directional tensor product complex tight framelets", Preprint, 2014, arXiv:1407.3234.