Reversible, Fast, and High-Quality Grid Conversions

Laurent Condat, *Member, IEEE*, Dimitri Van De Ville, *Member, IEEE*, and Brigitte Forster-Heinlein, *Member, IEEE*

***Abstract*—A new grid conversion method is proposed to resample between two 2-D periodic lattices with the same sampling density. The main feature of our approach is the *symmetric reversibility*, which means that when using the same algorithm for the converse operation, then the initial data is recovered exactly. To that purpose, we decompose the lattice conversion process into (at most) three successive shear operations. The translations along the shear directions are implemented by 1-D fractional delay operators, which revert to simple 1-D convolutions, with appropriate filters that yield the property of symmetric reversibility. We show that the method is fast and provides high-quality resampled images. Applications of our approach can be found in various settings, such as grid conversion between the hexagonal and the Cartesian lattice, or fast implementation of affine transformations such as rotations.**

***Index Terms*—Fractional delay filters, hexagonal grid, resampling, rotation, shears, 2-D lattices.**

I. Introduction

DIGITAL images are almost exclusively available on the Cartesian lattice, despite the existence of other periodic lattices with attractive properties. Probably, the main reasons behind this omnipresence are the trivial correspondence between pixel position and pixel index, and the availability of display devices with Cartesian geometry.

Resampling procedures come into play when converting from one lattice to another. Mathematically, a 2-D lattice [1]–[3] is a regular set of points of the plane, characterized by two linearly independant vectors $\mathbf{r}_1, \mathbf{r}_2 \in \mathbb{R}^2$, conveniently grouped in a 2×2 matrix $\mathbf{R} = [\mathbf{r}_1\ \mathbf{r}_2]$. Lattice sites have coordinates $\mathbf{Rk}, \mathbf{k} \in \mathbb{Z}^2$, and the lattice sampling density can be obtained as $1/|\det(\mathbf{R})|$ (lattice sites per unit surface). For resampling 2-D data from one lattice, with matrix $\mathbf{R}$, onto another lattice, with matrix $\mathbf{R}'$, typical approaches are based on reconstruction: a continuous-domain representation lying in a shift-invariant function space is constructed, that estimates the underlying (unknown) function $f(\mathbf{x})$ by means of its samples $f(\mathbf{Rk})$. Specifically, a generator function at every lattice site is weighted by a coefficient, after which the reconstruction is sampled on the target lattice. On the Cartesian lattice, generator functions are typically obtained as tensor-product extensions of 1-D versions. On non-Cartesian lattices, such as the hexagonal one, intrinsically 2-D generator functions are deployed; e.g., hex-splines [4] or box-splines [5]. The use of high-quality versions of these methods (say, beyond linear interpolation) requires a prefiltering step that can be computationally expensive. We highlight two main properties of these approaches: 1) successive resampling operations increasingly degrade the image quality; 2) the process is not invertible by a similar procedure. The latter point is related to the fact that the generator function is intricately linked to the source lattice, only. For instance, when using a linear tensor-product B-spline for Cartesian-to-hexagonal resampling, one would consider the three-directional linear box-spline [5]–[7] as an "equivalent" way to return from the hexagonal to Cartesian lattice. However, the original image will not be recovered after cascading both operations.

In this paper, we propose an alternative approach that is driven by the property of "symmetric reversibility"; i.e., the conversion between two arbitrary lattices with the same sampling density has an exact inverse operation, that is achievable with the same algorithm. Formally, we pursue

$$\mathcal{C}_{\mathbf{R}' \to \mathbf{R}} \circ \mathcal{C}_{\mathbf{R} \to \mathbf{R}'} = \mathcal{I}d, \quad \forall \mathbf{R}, \mathbf{R}'. \tag{1}$$

Our method is based on two fundamental observations: 1) a lattice can be turned into another one by at most three successive shear operations; 2) shear operations can be carried out by 1-D fractional delay operators that ensure the "symmetric reversibility" property. By construction, this approach will be separable and simple to implement; i.e., through 1-D operations along selected directions. Additionally, 1-D translators of arbitrary high order can be designed to guarantee high quality results.

Many applications can benefit from this type of grid conversion; in particular we propose the following settings.

Hexagonal-to-Cartesian Resampling: The hexagonal lattice of the so-called first type [8], depicted in Fig. 1, is characterized by the matrix

$$\mathbf{R}_{\text{hex}} = \sqrt{\frac{2}{\sqrt{3}}} \begin{bmatrix} 1 & 1/2 \\ 0 & \sqrt{3}/2 \end{bmatrix}. \tag{2}$$

Hexagonal sampling has numerous attractive properties, such as more efficient representation of isotropic band-limited 2-D signals [1], [9]. Recent technical progress has brought hexagonal sampling into the domain of consumer electronics and there now exist imaging sensors that acquire on the hexagonal lattice

Manuscript received September 7, 2007; revised July 27, 2008. This work was supported by the Marie Curie Excellence Team Grant MEXT-CT-2004-013477, Acronym MAMEBIA, funded by the European Commission. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Peyman Milanfar.

L. Condat and B. Forster-Heinlein are with the Institute of Biomathematics and Biometry, Helmholtz Zentrum München—German Research Center for Environmental Health, 85764 Neuherberg, Germany (e-mail: laurent.condat@helmholtz-muenchen.de; brigitte.forster@helmholtz-muenchen.de).

D. Van De Ville is with the Biomedical Imaging Group (BIG), École Polytechnique Fédérale de Lausanne (EPFL), CH-1015 Lausanne, Switzerland (e-mail: dimitri.vandeville@epfl.ch).

Digital Object Identifier 10.1109/TIP.2008.919361

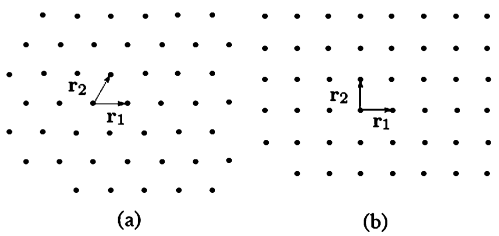

Fig. 1. (a) Regular hexagonal lattice and (b) the square lattice, with the same sampling density equal to 1.

[10], [11]. This is likely to foster a renewed interest for image processing on this lattice. A non-Cartesian acquisition lattice may also be constrained for technical reasons, e.g., in medical imaging devices. However, data sampled on such a lattice generally have, at some stage of their processing chain, to be resampled on the Cartesian lattice, typically for visualization.

Hexagonal-to-Cartesian resampling can be performed fast and efficiently with our new approach using 1-D filtering only. The reversibility property ensures that no loss of information is introduced during this conversion operation. One particular useful application of the reversibility is when hexagonally-sampled data needs to be stored or transmitted on a Cartesian lattice. Our approach will produce a high-quality intermediate representation of the data (that can be directly visualized), which can be reconverted into the original data on the hexagonal lattice at the receiver side. That way, existing infrastructure can be used to handle hexagonally-sampled data.

Cartesian-to-Hexagonal Resampling: Cartesian-to-hexagonal lattice conversion can be beneficial for low-level image processing tasks. It is known that the better isotropic properties of hexagonal sampling like twelve-fold symmetry and six-connectivity [12] can be successfully exploited for various image processing tasks [13]–[18]. Specifically, morphological operations can take great advantage of the hexagonal lattice [19]–[22]. In computer graphics also, various algorithms have been adapted to the hexagonal lattice, with a persistant result that a better quality can be obtained in comparison with the same operation on a square lattice of same density, for the same or lower computational cost [22]–[25]. In this context, reversibility is reassuring, since it guarantees that the image does not suffer from iterated conversions between the Cartesian and the hexagonal lattice.

Image Rotation: Resampling procedures are also commonly required for geometric transformations; e.g., image rotation. Let us consider the case of rotating the Cartesian lattice by an angle θ; i.e., $\mathbf{R}$ is the identity matrix and $\mathbf{R}' = \mathbf{R}_{-\theta}$ is the rotation matrix defined as

$$\mathbf{R}_\theta = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}. \tag{3}$$

For the particular setting of image rotation, our decomposition of the conversion process into three shears has been reported already in the literature—see [26] and references therein. However, our specific choice of fractional delay operations allows our method to satisfy the original and appealing property of symmetric reversibility

$$\mathcal{R}_{-\theta} \circ \mathcal{R}_\theta = \mathcal{I}d, \quad \forall\, \theta \in \mathbb{R}. \tag{4}$$

It is satisfying to know that one can rotate an image without losing any information and that the same operation applied in opposite sense will exactly recover the initial image.

Outline: This paper is organized as follows. We present in Section II our decomposition in shears. This suggests an easy and fast algorithm, detailed in Section III. We stress the fact that the proposed method is essentially discrete and only relies on 1-D shifts along rows or columns of the image. There is no underlying continuous model fitted on the image, as is the case with interpolation methods. On the other hand, our approach is limited to the conversion between lattices having the same sampling density, because we only use operations that leave the sampling density invariant. In Section IV, we present an in-depth analysis of our approach, including the "frequency shuffle" theorem that demonstrates how high-frequency components are affected by our method. Finally, in Section V, we show the experimental results illustrating the reversibility property, the efficiency of the implementation, and the quality compared to interpolation techniques.

II. Lattice Conversion: Decomposition Into Successive Shears

A. Mathematical Principle

In this section, we show that a succession of at most three well-chosen shears can convert data between two arbitrary lattices. Vectors of $\mathbb{R}^2$ are written in bold lowercase in this paper, e.g., $\mathbf{x} = [x_1\ x_2]^{\mathrm{T}}$. We consider two given lattice matrices $\mathbf{R} = [\mathbf{r}_1\ \mathbf{r}_2]$ and $\mathbf{R}' = [\mathbf{r}'_1\ \mathbf{r}'_2]$, with equal sampling density such that $|\det(\mathbf{R})| = |\det(\mathbf{R}')| = 1$.

Definition 1 (Shear): We define a shear operation along the direction $\mathbf{a}$ as the displacement of each point $\mathbf{x}$ in the direction of $\mathbf{a}$ with magnitude proportional to $\langle \mathbf{x}, \mathbf{a}^\perp \rangle$ where $\mathbf{a}^\perp = [-a_2\ a_1]^{\mathrm{T}}$ is orthogonal to $\mathbf{a}$. The linear shear operator

$$\mathbf{x} \mapsto \mathbf{x} + \lambda \langle \mathbf{x}, \mathbf{a}^\perp \rangle \mathbf{a} \tag{5}$$

with magnitude $\lambda \in \mathbb{R}$, has the associated matrix representation

$$\mathbf{S} = \mathbf{I} + \lambda \mathbf{a}\mathbf{a}^{\perp\mathrm{T}} = \begin{bmatrix} 1 - \lambda a_1 a_2 & \lambda a_1^2 \\ -\lambda a_2^2 & 1 + \lambda a_1 a_2 \end{bmatrix}. \tag{6}$$

It can be easily checked that for any shear, $\det(\mathbf{S}) = 1$. Its eigenvalues are 1 too. This indicates that the sampling density is preserved when applying a shear to a lattice.

Also, one fundamental property of a shear operation is that its converse is the same shear in the opposite direction; i.e., $\mathbf{S}^{-1} = \mathbf{I} - \lambda \mathbf{a}\mathbf{a}^{\perp\mathrm{T}}$. Designing discrete shear operators while maintaining this property is at the heart of the symmetric reversibility property of the conversion process between lattices.

Given a lattice with matrix $\mathbf{R}$, we are interested in shears applied along "natural" directions of the lattice, that is, along lines joining two lattice sites. To this purpose, it is more convenient to characterize a shear by its matrix $\mathbf{M}$ in the basis of the vectors of the lattice, instead of by its matrix $\mathbf{S}$ in the canonical basis. Both matrices are related by the change of basis $\mathbf{S}=\mathbf{R}\mathbf{M}\mathbf{R}^{-1}$. If the shear operation is performed along the first coordinate axis of the lattice, then the shear matrix $\mathbf{M}$ turns into a unit upper-triangular matrix

$$\mathbf{U}=\begin{bmatrix}1 & \mu\\ 0 & 1\end{bmatrix}. \tag{7}$$

Similarly, shearing along the second coordinate axis is associated to a unit lower-triangular matrix

$$\mathbf{L}=\begin{bmatrix}1 & 0\\ \mu & 1\end{bmatrix}. \tag{8}$$

Such shears are preferable, since they correspond to applying simple and efficient row-by-row or column-by-column operations on the data.

We now define the unitary matrix $\mathbf{P}$ that characterizes the conversion between the two lattices as

$$\mathbf{P}=\begin{bmatrix}p_{11} & p_{12}\\ p_{21} & p_{22}\end{bmatrix}=\mathbf{R}^{-1}\mathbf{R}'. \tag{9}$$

The possible decompositions of $\mathbf{P}$ in shear operations depend on the mutual orientations between the respective basis vectors of both source and target lattices. Depending on them, we can propose the following decompositions:

Decomposition 1 (ULU Factorization): If the vectors $\mathbf{r}_1$ and $\mathbf{r}'_1$ are not colinear, that is $p_{21}\neq 0$ in (9), then the conversion matrix can be decomposed in three shears

$$\begin{aligned}\mathbf{P}&=\mathbf{U}_1\mathbf{L}_2\mathbf{U}_3 &(10)\\ &=\begin{bmatrix}1 & a\\ 0 & 1\end{bmatrix}\begin{bmatrix}1 & 0\\ b & 1\end{bmatrix}\begin{bmatrix}1 & c\\ 0 & 1\end{bmatrix}. &(11)\end{aligned}$$

The values of a,b,c can be uniquely identified as

$$a=\frac{p_{11}-1}{p_{21}},\quad b=p_{21},\quad c=\frac{p_{22}-1}{p_{21}}. \tag{12}$$

A more in-depth mathematical treatment of this decomposition has been given in [27].

To provide a better insight into the decomposition process, let us describe the successive shears of the decomposition in more detail.

- The first shear operates on the initial lattice with matrix $\mathbf{R}$. We are looking for a shear operating along the first vector $\mathbf{r}_1$ of this lattice, thus with matrix $\mathbf{S}_1=\mathbf{R}\mathbf{U}_1\mathbf{R}^{-1}$, where $\mathbf{U}_1$ has the form

$$\mathbf{U}_1=\begin{bmatrix}1 & a\\ 0 & 1\end{bmatrix}. \tag{13}$$

- After the first shear, the lattice has matrix $\mathbf{R}\mathbf{U}_1$. We look for a shear operating along the second direction of this lattice, thus with matrix $\mathbf{S}_2=(\mathbf{R}\mathbf{U}_1)\mathbf{L}_2(\mathbf{R}\mathbf{U}_1)^{-1}$, where $\mathbf{L}_2$ has the form

$$\mathbf{L}_2=\begin{bmatrix}1 & 0\\ b & 1\end{bmatrix}. \tag{14}$$

- The lattice has now matrix $\mathbf{R}\mathbf{U}_1\mathbf{L}_2$. We look for a shear operating along the first vector of this lattice, thus with matrix $\mathbf{S}_3=(\mathbf{R}\mathbf{U}_1\mathbf{L}_2)\mathbf{U}_3(\mathbf{R}\mathbf{U}_1\mathbf{L}_2)^{-1}$ where $\mathbf{U}_3$ has the form

$$\mathbf{U}_3=\begin{bmatrix}1 & c\\ 0 & 1\end{bmatrix}. \tag{15}$$

At the end, the lattice has matrix $\mathbf{R}\mathbf{U}_1\mathbf{L}_2\mathbf{U}_3=\mathbf{R}'$, or, equivalently

$$\mathbf{S}_3\mathbf{S}_2\mathbf{S}_1\mathbf{R}=\mathbf{R}'. \tag{16}$$

The last factorization (16) provides the geometric interpretation of the decomposition, using shears along 1-D directions of the supporting lattices. On the other hand, the factorization (11), when read from the left to the right, provides the practical way for implementing the conversion between the two lattices, as detailed in the next section, since the operations are directly expressed in the basis of the pixel indices.

Decomposition 2 (LUL Factorization): If the vectors $\mathbf{r}_1$ and $\mathbf{r}'_1$ are colinear, but $\mathbf{r}_2$ and $\mathbf{r}'_2$ are not, that is $p_{12}\neq 0$ in (9), then the decomposition of the conversion process in three shears can be performed by switching the order of the directions along which the shears are performed. Specifically, we obtain

$$\begin{aligned}\mathbf{P}&=\mathbf{L}_1\mathbf{U}_2\mathbf{L}_3 &(17)\\ &=\begin{bmatrix}1 & 0\\ a & 1\end{bmatrix}\begin{bmatrix}1 & b\\ 0 & 1\end{bmatrix}\begin{bmatrix}1 & 0\\ c & 1\end{bmatrix} &(18)\end{aligned}$$

where a,b,c can be uniquely identified as

$$a=\frac{p_{22}-1}{p_{12}},\quad b=p_{12},\quad c=\frac{p_{11}-1}{p_{12}}. \tag{19}$$

Decomposition 3 (SUL Factorization): In the particular case where both $\mathbf{r}_1$ and $\mathbf{r}'_1$ and $\mathbf{r}_2$ and $\mathbf{r}'_2$ are colinear, then $\mathbf{P}$ has the form

$$\mathbf{P}=\begin{bmatrix}\sigma & 0\\ 0 & 1/\sigma\end{bmatrix} \tag{20}$$

and the lattice conversion actually consists in contracting the image in the horizontal direction with factor σ and dilating it in the vertical direction with factor σ. In this special case, a decomposition of the form (11) or (18) is not possible. One possibility is the so-called LULU decomposition [28], which consists of four shears

$$\begin{aligned}\mathbf{P}&=\mathbf{L}_1\mathbf{U}_2\mathbf{L}_3\mathbf{U}_4\\ &=\begin{bmatrix}1 & 0\\ -1 & 1\end{bmatrix}\begin{bmatrix}1 & 1-\frac{1}{\sigma}\\ 0 & 1\end{bmatrix}\begin{bmatrix}1 & 0\\ \sigma & 1\end{bmatrix}\begin{bmatrix}1 & \frac{1-\sigma}{\sigma^2}\\ 0 & 1\end{bmatrix}.\end{aligned}$$

Instead, we propose an alternative in three shears, using a first shear along the diagonal direction $\mathbf{r}_1+\mathbf{r}_2$

$$\begin{aligned}\mathbf{P}&=\mathbf{S}_1\mathbf{U}_2\mathbf{L}_3 &(21)\\ &=\begin{bmatrix}1-a & a\\ -a & 1+a\end{bmatrix}\begin{bmatrix}1 & b\\ 0 & 1\end{bmatrix}\begin{bmatrix}1 & 0\\ c & 1\end{bmatrix} &(22)\end{aligned}$$

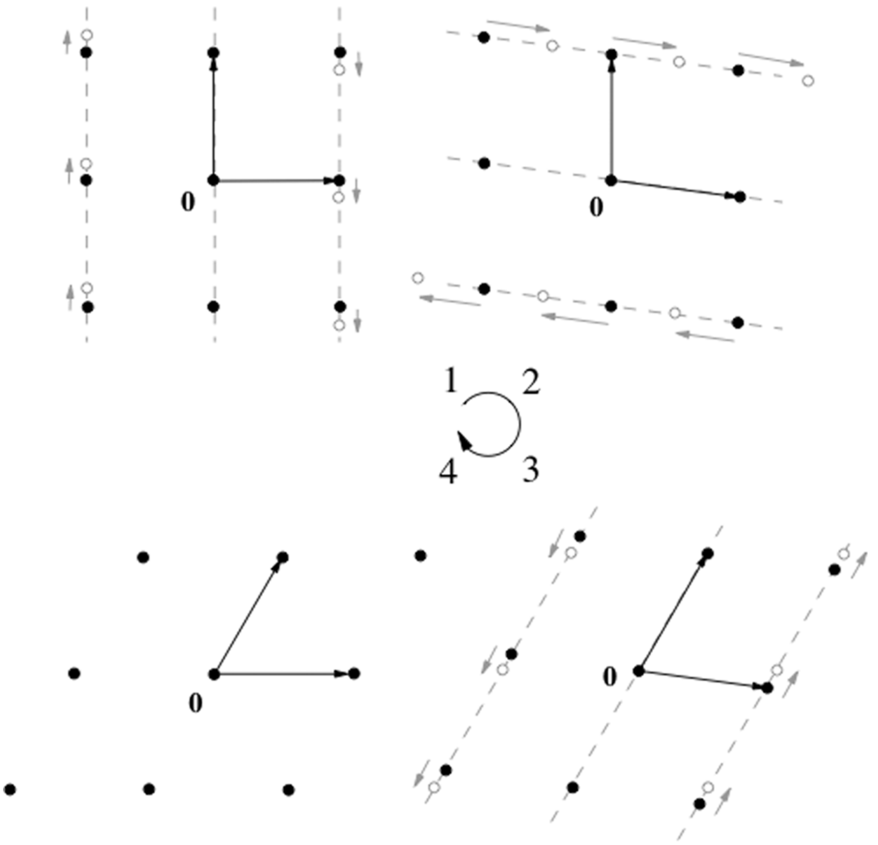

Fig. 2. Conversion from the Cartesian to the hexagonal lattice can be decomposed in three steps, using two intermediate lattices. Each of these three conversions consists in a shear along one of the two vectors generating the lattice on which the shear is applied. The converse operation of hexagonal to Cartesian conversion is simply obtained by reverting the order of the operations. On each subfigure, the displacements of the lattice sites caused by the shear are indicated by arrows; each black dot is moved to the next gray one in the direction of the shear.

with

$$a = 1 - \sigma, \quad b = 1 - \frac{1}{\sigma}, \quad c = 1 - \sigma^2. \tag{23}$$

We notice that the decomposition in three shears is not unique; i.e., other shear directions will result in other factorizations. For the directions we have chosen, horizontal, vertical, and diagonal, depending on the case, the decompositions are unique and can be advantageously performed along the lattices vectors. Assuming that the pixels are indexed accordingly, the process turns into 1-D filtering along rows and columns of the image. This suggests the efficient algorithm detailed in the following. If other addressing schemes are used for indexing the pixels, like the one of Middleton *et al.* [16], [18], this does not have any influence on the performances of the algorithm, as they also guarantee an easy access to the neighbors of a given pixel.

B. Conversion Between Cartesian and Hexagonal Lattices

Let us first illustrate our approach by considering the conversion from the Cartesian to the hexagonal lattice. Since the vector $\mathbf{r}_1$ of the hexagonal lattice is colinear with the Cartesian vector $[1\ 0]^T$, we use the decomposition (18); that is

$$\mathbf{R}_{\text{hex}} = \begin{bmatrix} 1 & 0 \\ a & 1 \end{bmatrix} \begin{bmatrix} 1 & b \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ c & 1 \end{bmatrix} \tag{24}$$

with $a = \sqrt{3} - \sqrt{6/\sqrt{3}}, b = \sqrt{\sqrt{3}/6}, c = 2 - \sqrt{6/\sqrt{3}}$.

The three successive shears convert the data between the successive lattices depicted in Fig. 2. The intuition behind this is that two shears are required to transform the vector $\mathbf{r}_2$ of the hexagonal lattice into the Cartesian vector $[0\ 1]^{\mathrm{T}}$ and a third shear then transforms $\mathbf{r}_1$ into $[0\ 1]^{\mathrm{T}}$. The practical process consists of three steps that only involve 1-D operations.

- On each column of the image s with index k_1, perform a translation of magnitude $-a \cdot k_1$ (in the direction $\uparrow$). This amounts to estimating the sample values $f(\mathbf{L}_1\mathbf{k})$ from the available pixel values $f(\mathbf{k})$.
- On each row of the image with index k_2, perform a translation of magnitude $-b \cdot k_2$ (in the direction $\rightarrow$). The pixel values of the obtained image are estimates of the sample values $f(\mathbf{L}_1\mathbf{U}_2\mathbf{k})$.
- On each column of the image with index k_1, perform a translation of magnitude $-c \cdot k_1$ (in the direction $\uparrow$). The pixel values of the final image v approximate the desired samples $f(\mathbf{R}_{\text{hex}}\mathbf{k})$.

Following the same analysis, hexagonal to Cartesian conversion is performed the same way as Cartesian to hexagonal conversion, but in reversing the order of the translations and making them in the opposite directions. This yields the three steps.

- On each column of the image s with index k_1, perform a translation of magnitude $c \cdot k_1$ (in the direction $\uparrow$).
- On each row of the image with index k_2, perform a translation of magnitude $b \cdot k_2$ (in the direction $\rightarrow$).

- On each column of the image with index k_1, perform a translation of magnitude $a \cdot k_1$ (in the direction $\uparrow$).

Clearly, the symmetric reversibility of the conversion process is obtained if and only if the inverse operation of a discrete translation with magnitude τ is the translation with magnitude $-\tau$. In Section III, we design the 1-D translators such that this property is satisfied.

Along with the conversion process in three shears we just described, there is a way for converting an image from the Cartesian lattice to a uniform hexagonal lattice in only two shears. Of course, this hexagonal lattice is not the same as in the previous section, but a rotated version: none of its natural directions are horizontal or vertical. In fact

$$\mathbf{R} = [\mathbf{u}_1, \ \mathbf{u}_2] = \begin{bmatrix} 1 & a \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ b & 1 \end{bmatrix} \tag{25}$$

with $a = \sqrt{2/\sqrt{3} - 1}$ and $b = (1 - \sqrt{2\sqrt{3} - 3})/2$, corresponds to the matrix of a uniform hexagonal lattice with density 1. In a framework where an image would be resampled from the Cartesian to an hexagonal lattice in order to benefit from the advantages of hexagonal sampling, it would be interesting to opt for this original lattice, since there is generally no reason to enforce the horizontal or vertical direction as being part of the lattice. Since only two resampling operations are required instead of three with the classical hexagonal lattice, the distortions introduced during the conversion will be of lower magnitude. Conversely, when designing hexagonally-arranged CCD sensors, it could be interesting to rotate them in order to align them with the lattice proposed here, keeping in mind that only two shears will then enable the conversion to the Cartesian lattice.

Finally, we mention that we proposed a different decomposition for the conversion from the hexagonal to the Cartesian lattice in [29], using the three natural directions of the hexagonal lattice. This amounts to performing translations along not only the rows and columns, but also the diagonals of the image. The decomposition described in the present paper is easier to implement, and turns out to provide images with slightly better quality.

C. Application to Image Rotation

Now, when applied to the problem of image rotation, our decomposition reverts to the one already known in the literature [26], [30]–[32]. The matrix $\mathbf{P} = \mathbf{R}_{-\theta}$ is the rotation matrix with angle $-\theta$, that can be decomposed as

$$\mathbf{P} = \begin{bmatrix} 1 & -\tan\left(\frac{\theta}{2}\right) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \sin(\theta) & 1 \end{bmatrix} \begin{bmatrix} 1 & -\tan\left(\frac{\theta}{2}\right) \\ 0 & 1 \end{bmatrix}. \tag{26}$$

We remark that only rotations with angle $\theta \in [-\pi/4, \pi/4]$ have to be processed by this decomposition, since rotations with larger magnitude can be decomposed as the composition of such a rotation and a rotation with angle $-\pi/2, \pi/2$ or π, which are trivially performed.

Decomposing the rotation in three shears provides algorithms—for the same quality—being computationally less complex than the full 2-D counterparts [26]. However, the method proposed in [26], implementing the 1-D translations using spline interpolation, is not reversible, in the following sense: the inverse of the rotation with angle θ exists, but it does not correspond to rotation with angle $-\theta$. Basically, the rotation introduces some blur, while its inverse operates some contrast enhancement. Our symmetrically reversible approach, to the contrary, does not introduce any blur, as discussed later.

We note that for particular angles, 2-D nearest neighbor interpolation is actually symmetrically reversible, and can be computed efficiently using the modulo transform proposed in [33]. The quality of this method is very poor; however, 2-D sinc interpolation (for which an efficient implementation was given in [34]) is not reversible, since the Nyquist regions of the source and target lattices do not coincide. It should also be stressed that other decompositions of rotations have been proposed in the literature [35]–[37], but they use scaling operations in addition to shears; hence, they cannot be made reversible, since some loss of information is always implied by a down-scaling.

D. Extensions to Higher Dimensions

An interesting question is whether the decomposition in three shears for the 2-D case proposed in this work can be extended to higher dimensions. In any dimension, a shear matrix along a canonical direction has unit determinant and differs from the identity by only one row. As a consequence of the factorization result in [27], a unit square matrix of size $N \times N$ can be factorized in at most $3N - 3$ such shear matrices, since it can be decomposed in 3 triangular matrices, each of which is simply decomposed in $N - 1$ shear matrices. To our knowledge, it is still an open question whether one can achieve a general factorization with less shears. The authors of [38] conjecture that a decomposition in $N + 1$ shears can be obtained. They base their hypothesis on their result that in the 3-D case, a decomposition of a matrix in four shears operating along one of the three canonical vectors is always possible (except if $\mathbf{P}$ is a diagonal matrix) [38]. This decomposition has been applied for implementing rotations in 3-D [39]. Therefore, applied to our context, one can convert data between two 3-D lattices using four shears along well-chosen directions. For instance, the conversion from the body-centered-cubic lattice to the Cartesian lattice $\mathbb{Z}^3$ is a practical problem in computer graphics [40].

III. One-Dimensional Fractional Delay Operators for Image Translation

A. Design Constraints for 1-D Translators

The important feature of our lattice-conversion procedure is that it only involves 1-D translations. Since several shears are successively applied to the image, the translation (a.k.a. *shift* or *delay*) operators should be designed with the greatest possible care. This section is devoted to the study of proper ways for translating a discrete 1-D signal $s = (s[k])_{k \in \mathbb{Z}}$, while respecting the property of symmetric reversibility.

The translation operator $\mathcal{T}_\tau : s \mapsto s'$ is defined such that

$$s'[k] = s[k - \tau] \quad \forall\, k \in \mathbb{Z}. \tag{27}$$

While this operation is straightforward when τ is an integer, we are faced with an interpolation problem in the general case.

Assuming the translation process is linear and shift-invariant, it is implemented by a discrete convolution

$$s' = s * h_\tau. \tag{28}$$

where h_τ is called a *fractional delay filter*. These filters have a long history and play a key role in many signal processing and communication systems, see [41] and references therein. The shears require each row or column of the image to be translated with a different shift τ, so we have to design a family of filters h_τ, for $\tau \in \mathbb{R}$. Further on, we denote the $\mathcal{Z}$-transform of a filter h by $H(z) = \sum_{k\in\mathbb{Z}} h[k]z^{-k}$ and its Fourier transform $\hat{h}(\omega) = H(e^{j\omega})$. We pursue two design principles.

1) *Symmetry*. Changing the sign of τ should correspond to the same operation but in reverse order on the data

$$H_{-\tau}(z) = H_\tau(z^{-1}). \tag{29}$$

For integer τ, the exact translation is achieved by a simple shift, that is, $H_\tau(z) = z^{-\tau}$. For the fractional part of the delay, we only have to design the filters h_τ for $\tau \in [0, 1/2]$. Then, for $\tau \in \mathbb{R}$, we select

$$H_\tau(z) = z^{-d} H_{|\tau - d|}\left(z^{\operatorname{sgn}(\tau - d)}\right) \tag{30}$$

where $d = \operatorname{sgn}(\tau)\lceil|\tau| + (1/2)\rceil - 1$.

2) *Symmetric reversibility*. As long as the filter $H_\tau(z)$ has no roots on the unit circle, it will be invertible. However, we want the inverse to be the translation in the opposite direction. This condition can be expressed as

$$H_{-\tau}(z) = \frac{1}{H_\tau(z)}. \tag{31}$$

Formally, $\mathcal{T}_{-\tau} \circ \mathcal{T}_\tau = \mathcal{I}d$ for every $\tau \in \mathbb{R}$. This makes the discrete shears and the whole conversion process inherit the symmetric reversibility.

B. Two Extreme Cases

1) Nearest Neighbor: The most straightforward approach for performing a translation is nearest neighbor interpolation, which consists in assigning to $s'[k]$ the value of the pixel closest to the position $k - \tau$. This corresponds to

$$H_\tau(z) = 1 \quad \forall \tau \in [0, 1/2]. \tag{32}$$

This elementary method is symmetrically reversible. Moreover, the set of the pixel values of the initial and converted images are the same, since this conversion method just re-orders pixels of the initial image at different locations. For example, a binary image will remain binary after conversion on a different lattice with this method, which can be advantageous in some applications. However, as shown in Section IV, this "shuffling" method does not provide very high quality.

2) Sinc: At the opposite side, assuming a band-limited hypothesis on the underlying continuous process, Shannon's theorem [42] tells us that the ideal fractional delay operator is defined as

$$H_\tau(e^{j\omega}) = e^{-j\omega\tau}. \tag{33}$$

which correspond to interpolation-based translation with the sinc kernel

$$s'[k] = \sum_{l\in\mathbb{Z}} s[l] \operatorname{sinc}(k - l - \tau) \tag{34}$$

where $\operatorname{sinc}(x) = \sin(\pi x)/(\pi x)$. The idea of using this ideal translator in combination with a decomposition in shears was proposed in [26] and [43] for 2-D and 3-D rotations, respectively. However, the infinitely long response of sinc interpolation requires much computation time and is prone to the introduction of unwanted oscillations (ringing), due to the fact that the band-limited hypothesis is actually false for most natural images.

When applying the ideal translation to a signal s with finite length T, it is usual to implicitly consider that the underlying process is T-periodic. Using the FFT then provides a convenient way for exactly performing this translation in the Fourier domain. However, as already pointed out in [43], this has to be done carefully. The method simply consists in computing the FFT of s, then in multiplying each FFT coefficient by a complex value of the form $e^{-j\tau k\delta/T}$ with appropriate value δ, and finally going back in the spatial domain using the inverse FFT. However, if T is even, the last FFT coefficient, which corresponds to the energy component at the Nyquist frequency, is real valued. In order for the delayed signal to be real, its Nyquist coefficient should remain real. This means that the Nyquist coefficient should not be multiplied by a complex shift and should instead be left unchanged. This particularity comes from the fact that in a discrete real signal, the phase of its component at the Nyquist frequency can not be determined. It is assumed to be zero by convention, so the phase has to remain zero after translation.

Moreover, the FFT has a computation cost proportional to $T\log(T)$. This is more than the linear time corresponding to implementing the translation in the spatial domain with short realizable filters, as we present in the following.

C. All-Pass Fractional Delay Flters

Let us go back to the general case. In fact, (31) is equivalent for the delay filter to be an all-pass filter, that is, to have magnitude 1 in the frequency domain

$$|H_\tau(e^{j\omega})| = 1, \quad \forall \omega \in \mathbb{R}. \tag{35}$$

So, an all-pass filter h_τ is entirely characterized by its phase response $\theta_h(\omega)$ such that

$$\hat{h}_\tau(\omega) = e^{j\theta_h(\omega)}. \tag{36}$$

Ideally, $\theta_h(\omega) = -\tau\omega$ for every $\omega \in (-\pi, \pi)$.

We are interested in realizable filters having rational transfer functions, in order to implement them in the spatial domain. A realizable all-pass filter can be written under the form

$$H_\tau(z) = z^N \frac{D(z)}{D(z^{-1})} \tag{37}$$

for some unique $N \in \mathbb{Z}$ and polynomial D such that $D(0) = 1$. If $-N$ is greater or equal than the degree of D, then h_τ is a causal filter; that is, $h_\tau[k] = 0$ for every $k < 0$. In order to evaluate the quality of a realizable filter h_τ, we have to study its phase properties. Two measures are classically used: the phase delay $-\theta_h(\omega)/\omega$ and the group delay $-\mathrm{d}\theta(\omega)/\mathrm{d}\omega$. In the ideal case, both should be constant and equal to τ.

The only known class of realizable filters having explicit formulas of their coefficients as a function of τ is the class of Thiran filters [41], [44].

The Thiran filter t_τ^N of order N is causal, hence it takes the form

$$T_\tau^N(z) = \frac{a_N + a_{N-1}z^{-1} + \cdots + a_1 z^{-N+1} + z^{-N}}{1 + a_1 z^{-1} + \cdots + a_{N-1}z^{-N+1} + a_N z^{-N}} \quad (38)$$

with

$$a_k = (-1)^k \binom{N}{k} \prod_{n=0}^{N} \frac{\tau - N + n}{\tau - N + n + k} \quad \forall k \in 1 \ldots N \quad (39)$$

where $\binom{N}{k} = N!/(k!(N-k)!)$ is the binomial coefficient. The Thiran filters are designed to have maximally flat group delay at the zero frequency. This is an asymptotic design constraint that privileges the accurate translation of the low frequency part of the signal. This design is in fact particularly relevant for images, since it is well known that images have their energy content essentially localized around the origin in the frequency domain.

Thiran filters t_τ^N are stable (that is, its poles, the roots of the denominator of $T_\tau^N(z)$ are inside the complex unit circle) if and only if $\tau > N - 1$. In this stability interval, it achieves its best performances for τ close to N. Moreover, its phase properties are better for $N - \epsilon$ than for $N + \epsilon$. For our purpose, we want to design delay filters for $\tau \in [0, 1/2]$, which does not suit with the stability interval of the Thiran filters. Moreover, the constraint for the filter to be causal is not necessary for 1-D treatments on the rows and columns of images. Thus, we design new all-pass filters having maximally flat group delay, with a rational form that is not causal, and adapted for $\tau \in [0, 1/2]$.

Definition 2 (Type I Filters): We define the type I causal/anti-causal filters $h_\tau^{1,N}$ by

$$\begin{aligned} H_\tau^{1,N}(z) &= z^{-N} T_{N-\tau}^N(z^{-1}) \quad (40)\\ &= \frac{1 + b_1 z^{-1} + \cdots + b_{N-1}z^{-N+1} + b_N z^{-N}}{b_N z^N + b_{N-1}z^{N-1} + \cdots + b_1 z^1 + 1} \quad (41) \end{aligned}$$

with

$$b_k = (-1)^k \binom{N}{k} \prod_{n=0}^{N} \frac{\tau - n}{\tau - n - k} \quad \forall k \in 1 \ldots N. \quad (42)$$

The numerator of $H_\tau^{1,N}(z)$ is causal and its denominator is anticausal. This design reverts to matching the interval $\tau \in [N - 1/2, N]$, where the Thiran filter t_τ^N has optimal performance, in the interval $\tau \in [0, 1/2]$.

The filtering step on a signal s of finite length T with $h_\tau^{1,N}$ can be easily achieved in one single in-place backward pass

```
for i from $T - 1$ down to 0 do
  for k from 1 to N do
    $s[i] + = b\_k * (s[i - k] - s[i + k])$ .
```

In practice, the signals to be manipulated have finite length. For the translation to be reversible without expanding the size of the signal, periodic boundary conditions have to be used. Then, an initialization step is required for the filtering pass, that can be performed the same way as in [45], where an algorithm based on inverse filtering is proposed for spline interpolation. That is, the initialization is performed on a few terms of the signals so that the error is below some prescribed level of precision.

Definition 3 (Type II Filter): We define the type II centered filters $h_\tau^{2,N}$ by

$$H_\tau^{2,N}(z) = \frac{a_N z^{N/2} + a_{N-1}z^{N/2-1} + \cdots + a_1 z^{-N/2+1} + z^{-N/2}}{z^{N/2} + a_1 z^{N/2-1} + \cdots + a_{N-1}z^{-N/2+1} + a_N z^{-N/2}} \quad (43)$$

when N is even, with the a_k given by (39), and otherwise

$$\begin{aligned} H_\tau^{2,N}(z) &= z^{-1} T_{1-\tau}^N(z^{-1}) \quad (44)\\ &= \frac{z^{(N-1)/2} + c_1 z^{(N-3)/2} + \cdots + c_N z^{(-N-1)/2}}{c_N z^{(N+1)/2} + \cdots + c_1 z^{(-N+2)/2} + z^{(-N+1)/2}} \quad (45) \end{aligned}$$

with

$$c_k = (-1)^k \binom{N}{k} \prod_{n=0}^{N} \frac{1 - \tau - n}{1 - \tau - n - k} \quad \forall k \in 1 \ldots N. \quad (46)$$

We remark that $h_\tau^{1,1} = h_\tau^{2,1}$. These type II filters revert to using Thiran filters outside of their stability interval, where they have optimal performances. The idea behind this design is that the filter $h_\tau^{1,N}$ is actually a combination of two filters (the numerator and the denominator) that each translate the signal by $\tau/2$. A $N + 1$-tap FIR delay filter will perform at its best if its coefficients are equally positioned around 0. Intuitively, the best way of predicting the sample value at location $\tau \in [0, 1/2]$ is to use for the computation the pixel values at locations $-N/2, \ldots, N/2$ and not $0, \ldots, N$.

However, the implementation of an all-pole filter that is neither causal nor anti-causal is problematic. The spectral factorization method that consists in factorizing a filter in stable causal and anti-causal parts, as proposed in [45], requires the computation of the roots of a polynomial of degree N, whose coefficients depend on τ. An explicit expression of these roots only exists if $N \leq 4$, but already for $N = 3$, the formulas become impracticable. For arbitrary values of N, we can use an algebraic approach, that amounts to solving a linear system using a fast *LU* decomposition for band-diagonal matrices, similarly to the approach described in [46]. This is the method we adopted. In the case $N=2$, however, we can provide a faster implementation, based on the following factorization:

$$H_\tau^{2,2}(z)=\frac{(1-\alpha_{-\tau}z^{-1})(1-\alpha_\tau z)}{(1-\alpha_\tau z^{-1})(1-\alpha_{-\tau}z)} \tag{47}$$

where

$$\alpha_\tau=\frac{-4+\tau^2+\sqrt{12-3\tau^2}}{3\tau+2+\tau^2}. \tag{48}$$

This yields the fast implementation

```
for i from 0 to $T-1$ do
    $\mathrm{s[i]}+=\mathrm{alpha\_tau}*(\mathrm{s[i-1]}-\mathrm{s[i+1]})$;
for i from $T-1$ down to 0 do
    $\mathrm{s[i]}+=\mathrm{alpha\_(-tau)}*(\mathrm{s[i+1]}-\mathrm{s[i-1]})$.
```

For all these implementations, boundaries have to be handled carefully. However, it is common to first extend (by padding or mirror extension) the image to a larger one before applying the conversion process, in order to avoid the shuffling of some parts of the images at the boundaries, caused by the periodic conditions used in the shears. Such an extension of the image is typically required for rotation, since the rotated domain extends beyond the frame of the initial image. In that case, it is not useful to consider the exact computation of the few pixels on the boundaries, that will be cropped afterwards.

Finally, we note that the two extreme cases described in Section III-B can be seen as two limiting cases of the proposed approach; i.e., for $N=0$ and $N\to\infty$, respectively. Consequently, we denote these two methods by h_τ^0 and h_τ^∞ in the following. In fact, as N increases, the phase delay of the filters $h_\tau^{1,N}$ and $h_\tau^{2,N}$ approaches more and more the ideal phase delay, which is τ within $(-\pi,\pi)$ and 0 at $\pm\pi$, as can be seen in Fig. 3.

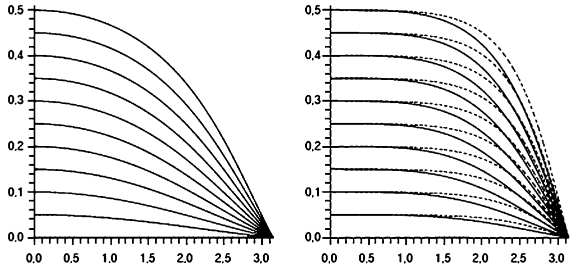

Fig. 3. Phase delay of (a) $h_\tau^{1,1}$, (b) $h_\tau^{1,2}$ (solid line) and $h_\tau^{2,2}$ (dotted line), for several τ in [0,0.5].

IV. In-Depth Analysis

A. Quasi-Translation

It is important to recognize that our method does not introduce traditional blurring during the conversion process. The use of all-pass filters leaves unchanged the magnitude of all frequency components along the shear directions. Even more, our filters $h_\tau^{1,N}$ and $h_\tau^{2,N}$ are quasi-translation filters, also called Neville filters [47], of order $2N+1$. A filter h is a Neville filter of order N and shift τ if $\hat{h}(\omega)=e^{-j\omega\tau}+o(\omega^{N-1})$. These filters perfectly translate polynomials of degree at most $N-1$. Specifically, every polynomial signal $P(\mathbb{Z})$, where $P(X)$ is a polynomial of degree at most $2N$, is transformed into $P(\mathbb{Z})*h_\tau^{i,N}=P(\mathbb{Z}-\tau)$. In other words, the conversion process between lattices is exact if the pixel values are samples of a 2-D polynomial with total degree at most $2N$.

B. Orthogonality

The conversion process is also orthogonal. For two images u_1 and u_2 that are converted onto another lattice, forming the images v_1 and v_2, respectively, we have: $\langle u_1,u_2\rangle=\langle v_1,v_2\rangle$. More formally, the conversion operator $\mathcal{C}$ is such that $\mathcal{C}^{-1}=\mathcal{C}^*$, the adjoint operator of $\mathcal{C}$, which is another way to see the property of symmetric reversibility.

This property can be of great interest in applications. For instance, quantizing a pixel value before or after the conversion introduces the same distortion. Denoising could be performed after the image has been converted onto another lattice and before it is turned back onto the original lattice. This method could even be applied locally; i.e., have an image block converted to an adaptive lattice with one vector in the direction of maximum local regularity, and then applying a separable treatment that differentiates along and across the edges.

C. Shuffle the Frequencies

We first have to introduce some more notions on 2-D signals and lattices. We define the Fourier transform of a function $f\in L_2(\mathbb{R}^2)$ as $\hat{f}(\boldsymbol{\omega})=\int_{\mathbb{R}^2}f(\mathbf{x})\exp(-j\boldsymbol{\omega}^{\mathrm{T}}\mathbf{x})\,\mathrm{d}\mathbf{x}$. A discrete image $s=(s[\mathbf{k}])_{\mathbf{k}\in\mathbb{Z}^2}$, whose pixels are localized on the lattice with matrix $\mathbf{R}$, can be interpreted in the continuous domain as a Dirac "brush" distribution $s_\delta(\mathbf{x})=\sum_{\mathbf{k}\in\mathbb{Z}^2}s[\mathbf{k}]\delta(\mathbf{x}-\mathbf{Rk})$, where $\delta(\mathbf{x})$ is the 2-D Dirac distribution. So, we define the Fourier transform of the image accordingly, as

$$\hat{s}(\boldsymbol{\omega})=\sum_{\mathbf{k}\in\mathbb{Z}^2}s[\mathbf{k}]\exp(-j\boldsymbol{\omega}^{\mathrm{T}}\mathbf{Rk}). \tag{49}$$

To a lattice with matrix $\mathbf{R}$, we also associate its reciprocal lattice with matrix $2\pi\hat{\mathbf{R}}=2\pi(\mathbf{R}^{-1})^{\mathrm{T}}$. The effect of sampling a function $f(\mathbf{x})$ on a lattice is to replicate its Fourier transform at the dual lattice sites $2\pi\hat{\mathbf{R}}\mathbf{k}$, as described by the Poisson summation formula (see [4])

$$\sum_{\mathbf{k}\in\mathbb{Z}^2}f(\mathbf{Rk})\exp(-j\boldsymbol{\omega}^{\mathrm{T}}\mathbf{Rk})=\sum_{\mathbf{k}\in\mathbb{Z}^2}\hat{f}(\boldsymbol{\omega}-2\pi\hat{\mathbf{R}}\mathbf{k}). \tag{50}$$

Therefore, for a given lattice, we can define its Nyquist region as the Voronoi cell of its reciprocal lattice; that is, the region of the frequency plane consisting of all points closer to the origin $\mathbf{0}$ than to any other dual lattice site. Indeed, if $\hat{f}$ is nonvanishing only in the Nyquist region, sampling f on the lattice does not create aliasing, since the replicas $\hat{f}(\boldsymbol{\omega}-2\pi\hat{\mathbf{R}}\mathbf{k})$ do not overlap.

To interpret the effect of the conversion process on the Fourier transform of the image, we state the following Theorem. Basically, the main effect consists in shuffling the (high-)frequency content of the image, so as to tile the frequency plane differently. No information is lost in the process.

Theorem 1 (Frequency Shuffle): Let us consider an image u sampled on a lattice $\Lambda=\{\mathbf{Rk}\,|\,\mathbf{k}\in\mathbb{Z}^2\}$ with matrix $\mathbf{R}$ and

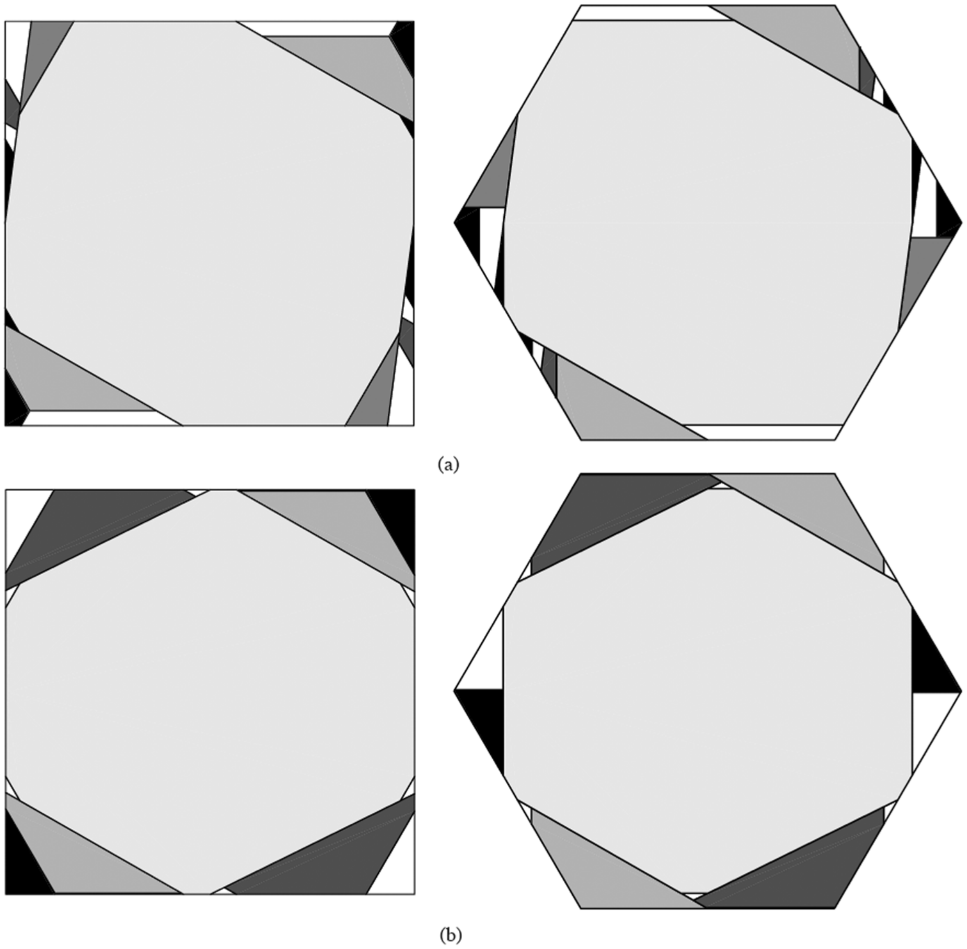

Fig. 4. Nyquist regions of the (left) Cartesian and (right) hexagonal lattices, respectively. The parts that are moved during the conversion process between these two lattices are indicated in the same gray tint. (a) Decomposition in three shears proposed in this work. (b) Decomposition as we proposed in [29], which includes one diagonal shear.

$\mathbf{a} \in \Lambda$, a vector of the lattice having minimal length (that is, $r\mathbf{a} \notin \Lambda$ for every $r \in (0,1)$). The image u is converted to the lattice $\mathbf{S}\Lambda$, where $\mathbf{S}$ is a single shear with direction $\mathbf{a}$ and magnitude λ, as given by (6). We assume that the translations are performed using 1-D sinc interpolation. So, by adapting (34), the pixel values of the target image v are defined by

$$v[\mathbf{k}] = \sum_{l\in\mathbb{Z}} u[\mathbf{k}+l\mathbf{R}^{-1}\mathbf{a}]\mathrm{sinc}(l-\lambda\langle\mathbf{R}\mathbf{k},\mathbf{a}^{\perp}\rangle). \qquad (51)$$

Then, the Fourier transforms of u and v are related as follows:

$$\hat{v}(\boldsymbol{\omega}) = \hat{u}\left(\boldsymbol{\omega} + 2\pi\lambda\left\lfloor\frac{\langle\boldsymbol{\omega},\mathbf{a}\rangle}{2\pi}+\frac{1}{2}\right\rfloor\mathbf{a}^{\perp}\right) \qquad (52)$$

where $\lfloor\cdot\rfloor$ is the integer part operator.

Proof: In the spatial domain, the conversion process can be decomposed in two steps: first, a convolution of u_δ with the function $\mathrm{sinc}(\langle\mathbf{x},\mathbf{a}\rangle/||\mathbf{a}||^2)\mathrm{sinc}(\langle\mathbf{x},\mathbf{a}^{\perp}\rangle)$ and then, resampling the obtained function on $\mathbf{S}\Lambda$. In the Fourier domain, this convolution amounts to multiplying $\widehat{u_\delta}(\boldsymbol{\omega}) = \hat{u}(\boldsymbol{\omega})$ by the function $\mathbb{1}_{(-\pi,\pi)}(\langle\boldsymbol{\omega},\mathbf{a}\rangle)\mathbb{1}_{(-\pi,\pi)}(\langle\boldsymbol{\omega},\mathbf{a}^{\perp}\rangle/||\mathbf{a}||^2)$. Then, resampling amounts to replicating this product at every dual lattice site $2\pi\widehat{\mathbf{SR}}\mathbf{k} = 2\pi\mathbf{S}^{-\mathrm{T}}\mathbf{R}^{-\mathrm{T}}\mathbf{k} = 2\pi\hat{\mathbf{R}}\mathbf{k} - 2\pi\lambda\langle\mathbf{R}^{-\mathrm{T}}\mathbf{k},\mathbf{a}\rangle\mathbf{a}^{\perp}$, for $\mathbf{k} \in \mathbb{Z}^2$. Using the $2\pi\hat{\mathbf{R}}$-periodicity of u, this yields

$$\hat{v}(\boldsymbol{\omega}) = \sum_{l\in\mathbb{Z}} \hat{u}(\boldsymbol{\omega}+2\pi l\lambda\mathbf{a}^{\perp})\mathbb{1}_{(-\pi,\pi)}(\langle\boldsymbol{\omega},\mathbf{a}\rangle-2\pi l). \qquad (53)$$

Since we can also write

$$\hat{u}(\boldsymbol{\omega}) = \sum_{l\in\mathbb{Z}} \hat{u}(\boldsymbol{\omega})\mathbb{1}_{(-\pi,\pi)}(\langle\boldsymbol{\omega},\mathbf{a}\rangle-2\pi l) \qquad (54)$$

(52) follows by identifying (53) and (54). □

This theorem states that converting an image from one lattice to another by using sinc translation does not modify its frequency content, but displaces it. More precisely, the conversion amounts to applying block-wise shears to the Fourier transform of the image. The accumulated displacements for the three shears are illustrated in Fig. 4(a), for the conversion between the Cartesian and hexagonal lattices. Parts of the Nyquist region containing high-frequency components are shuffled around, so as to tile the plane with another periodic pattern. Different choices for the directions of the shears yield different shuffles. For example, in Fig. 4(b), the decomposition that uses one diagonal shear, as proposed in [29], is shown, resulting in a more symmetrical repartition of the Nyquist region.

TABLE I
PSNR RESULTS OBTAINED AFTER CONVERSION OF AN IMAGE FROM THE HEXAGONAL LATTICE TO THE CARTESIAN LATTICE

	interpolation		proposed conversion method in three shears							
image	trilinear	cubic box-spline	$h_\mathcal{T}^0$	$h_\mathcal{T}^{1,1}$	$h_\mathcal{T}^{1,2}$	$h_\mathcal{T}^{1,3}$	$h_\mathcal{T}^{2,2}$	$h_\mathcal{T}^{2,3}$	$h_\mathcal{T}^{2,4}$	$h_\mathcal{T}^{\infty}$
zoneplate	32.70	52.68	20.52	37.00	43.95	48.58	48.47	56.33	**63.13**	59.95
phantom	37.93	**42.01**	28.57	37.61	38.76	39.21	39.24	39.70	39.90	39.71

We notice that the central part of the Nyquist region is left unchanged. This domain is defined as

$$\Omega = \{\boldsymbol{\omega} \,|\, \langle \boldsymbol{\omega}, \mathbf{a}_i \rangle \in (-\pi, \pi); i = 1, \ldots, N\} \tag{55}$$

where the $\{\mathbf{a}_i\}_{i=1,\ldots,N}$ are the N vectors along which the shears are applied. This means that for band-limited signals with frequency components within Ω, three shears using sinc translators is equivalent to 2-D sinc interpolation. Consequently, we have a practical implementation of resampling using 2-D sinc interpolation, for this class of band-limited signals. Moreover, this property provides a way to compare the quality of decompositions in three shears along different directions; i.e., we can associate to each decomposition the radius of the greatest circle included in Ω. This value is the maximum frequency for isotropic band-limited signals, so that they are unaltered by the conversion process when using sinc translators. For instance, our choice of shears for the Cartesian to hexagonal conversion, given by (24), provides a circle with radius $\sqrt{\sqrt{3}/2}\pi \approx 0.93\pi$, which is close to the maximal value of π achievable by true 2-D sinc interpolation. This value is also about 4% larger than the radius associated to the decomposition proposed in [29], which uses one diagonal shear. So according to this criterion, the decomposition in (24) is superior. This is confirmed by the experimental results in the next section.

Finally, when using one of the proposed filters—without ideal phase response—additional (and much harder to quantify) distortions of the frequency content occur.

V. Experimental Results

A. Hexagonal-to-Cartesian Resampling

In order to evaluate the quality of the resampled image in a hexagonal-to-Cartesian setting, we need a ground truth that is available on both lattices. To this end, we use synthetic images defined by an analytical function $f(\mathbf{x})$. In the Appendix, we propose the zoneplate and an adapted version of the Shepp–Logan phantom. We use them to produce images of size 512×512 and 256×256, respectively.

The experimental protocol goes as follows. First, we build the initial image by sampling f on the hexagonal lattice. Then, we convert it onto the Cartesian lattice by means of one of the methods in the comparison. The obtained image is then compared[1] to the ground truth, obtained by sampling f on the Cartesian lattice directly.

The experimental results are reported in Table I and are illustrated in Fig. 5. We tested our method using different filters, as well as three-directional box-spline interpolation [5] of approximation order 2 and 4. As can be observed, the quality increases rapidly with the order of the filter, but even the simple order 1 filter achieved a satisfactory quality. For high orders, our filters outperformed the sinc method, and also for the zoneplate image, the fourth-order box-spline interpolation, which can be considered as high-quality state-of-the-art for hexagonal-to-Cartesian resampling. However, for the Shepp–Logan image, which fits less well in the band-limited assumption than the zoneplate does, the fourth-order box-spline interpolation obtained the best results. However, the box-spline interpolation has a much higher computational complexity. In general, the quality of the images resampled with our method was very good for order 2 delay filters.

It should be noted that our method is not stationary; i.e., the procedure changes with the local neighborhood, and the exact symmetry properties of the initial signal are lost after conversion. This is visible in Fig. 5 where the error images for the zoneplate pattern obtained with our method only have a symmetry around the origin, while interpolation with symmetric kernels provide a four-fold symmetry. The asymmetry is low, however. With respect to this specific issue, the decomposition proposed in [29] turns out to provide more symmetrical results, but the same resampling experiments give PSNRs lower by about 1 dB in comparison with the decomposition proposed in the present work, without diagonal shear.

B. Cartesian-to-Hexagonal Resampling

In Fig. 6, we provide examples of images converted from the Cartesian onto the hexagonal lattice. Even with the simple order 1 filter, the converted image is free of visible artifacts. Increasing the order of the filter makes no visible difference for this image. For comparison, we also provide the image resampled using bicubic interpolation. It appears slightly blurred in comparison with the initial image and with the image converted using our method.

Numerical results obtained by resampling the zoneplate and phantom images from the Cartesian to the hexagonal lattice, not reproduced here, are very similar to the ones in Table I.

C. Successive Image Rotations

In a final experiment, we performed nine successive rotations of angle $2\pi/9$ on natural images. The cumulative effect was then observed by comparing the final image against the initial one. Notice that for the proposed technique, the final image can be rotated back nine times to recover the original image again.

The standard test images all had sizes of 512×512, except 256×256 for the *camera* image. They were first extended within a larger support, before applying the rotations, and

[1] The PSNR is computed using the floating point values of the pixels for both the reference and resampled image, without quantization. If the pixel values were rounded to integers in the range [0, 255] before computing the PSNR, an overall improvement of about 1 dB could be observed. Such a gain is caused by clipping, for high-contrast images like the zoneplate and the phantom, which have many pixels having gray level 0 or 255. Here, however, we believe that there is no reason to introduce such a bias in the results.

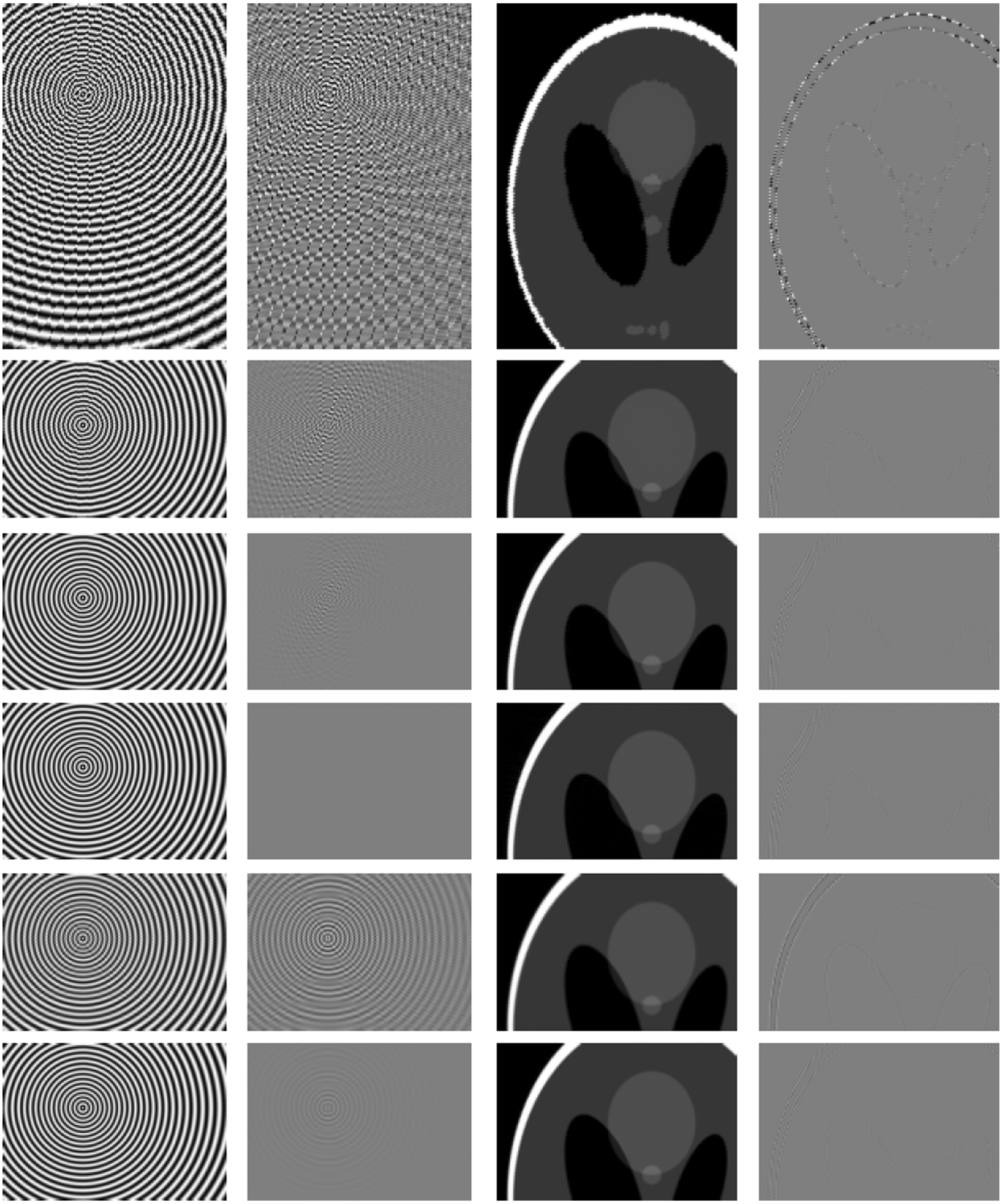

Fig. 5. Results of conversion from the hexagonal to the Cartesian lattice. From left to right: Resampled image from the zoneplate image, difference between the resampled image and the ground truth (gray is zero), resampled image from the phantom image, difference between the resampled image and the ground truth. From top to bottom: First four rows are obtained with our decomposition in three shears with the filters $h^0_\tau, h^{1,1}_\tau, h^{2,2}_\tau, h^\infty_\tau$; the last two rows are obtained by interpolation using the trilinear order 2 box-spline and the order 4 box-spline.

cropped to the initial size afterwards. The origins of the lattices were placed at the centers of the images.

The results are reported in Table II and are illustrated in Fig. 7. Performing the translations using nearest neighbor interpolation clearly provides a poor quality. For the more advanced delay filters, the method significantly outperforms the classical bilinear interpolation, which provides blurred images. The bicubic interpolation [48] is also outperformed for $h^{1,2}_\tau$ or higher order, while

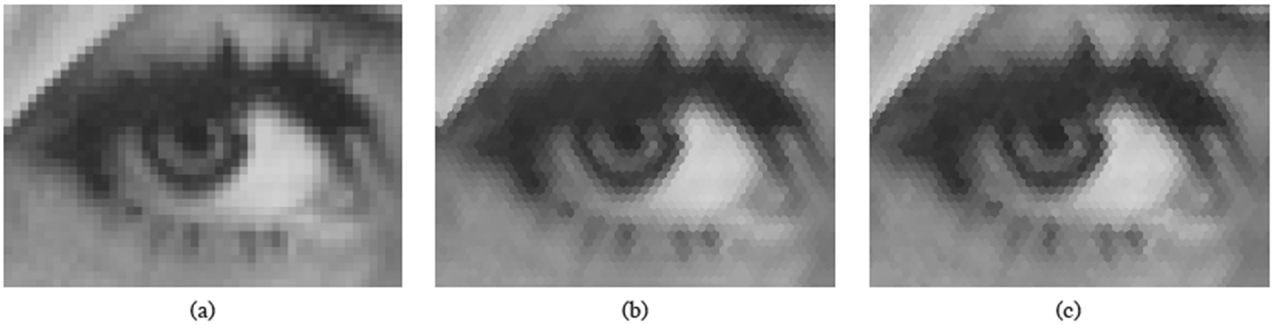

Fig. 6. "Eye of Lena" defined on the (a) Cartesian lattice is resampled on the (b) hexagonal lattice using bicubic interpolation and (c) our conversion method in three shears with the filter $h_{\tau}^{1,1}$. The image (a) can be perfectly recovered from (c), but not from (b).

TABLE II
PSNR RESULTS FOR ROTATIONS EXPERIMENTS ON SEVERAL CLASSICAL IMAGES OF THE LITERATURE. NINE ROTATIONS OF ANGLE $2\pi/9$ ARE PERFORMED

	interpolation				3 shears	conversion in three shears with proposed filters							
image	bilinear	bicubic	SP3	O-MOMS	+SP3 [26]	h_{τ}^{0}	$h_{\tau}^{1,1}$	$h_{\tau}^{1,2}$	$h_{\tau}^{1,3}$	$h_{\tau}^{2,2}$	$h_{\tau}^{2,3}$	$h_{\tau}^{2,4}$	h_{τ}^{∞}
lena	31.57	37.13	40.26	**41.87**	38.94	24.74	34.74	36.90	37.89	37.93	39.11	39.79	40.95
barbara	24.67	27.55	31.29	34.40	29.32	20.30	25.11	27.67	29.36	29.46	32.06	33.81	**36.67**
baboon	23.13	26.46	29.16	**30.83**	27.61	18.86	23.92	25.69	26.61	26.66	27.86	28.63	30.01
lighthouse	24.49	29.29	33.20	35.89	31.35	19.40	27.11	29.76	31.23	31.29	33.25	34.45	**36.31**
goldhill	29.97	34.01	36.89	**38.75**	35.47	24.38	31.57	33.49	34.54	34.59	36.05	37.02	38.94
boat	28.17	32.80	35.44	**36.82**	34.24	22.26	30.26	32.03	32.87	32.91	33.96	34.60	35.73
camera	24.84	28.91	31.75	**33.48**	30.41	19.53	26.61	28.48	29.42	29.46	30.66	31.41	32.92
peppers	30.98	35.68	37.98	**39.01**	35.28	23.98	31.80	33.37	34.14	34.18	35.17	35.79	36.98
time	0.28	0.63	0.65	0.65	0.10	0.05	0.08	0.09	0.10	0.14	0.59	0.73	1.17

$h_{\tau}^{2,4}$ or higher order provides similar results to the cubic spline interpolation [45] (denoted SP3), often considered as the reference for resampling tasks. However, even with large orders, we do not achieve the quality of the cubic O-MOMS interpolation [49], which can be considered as the state-of-the-art method for interpolation on the Cartesian lattice. A remarkable exception are the images with high frequency patterns (trousers of Barbara and fence in Lighthouse). We also report the results obtained by Unser *et al.* [26] with the same decomposition in three shears, but using cubic spline interpolation to perform the translations. This last method is not symmetrically reversible, and provides similar results to our approach with $h_{\tau}^{2,2}$ or $h_{\tau}^{2,3}$.

Interpolation methods achieve a tradeoff between blurring, aliasing and ringing, while our method does not introduce any blur. Using the sinc filter h_{τ}^{∞} creates ringing artifacts that spread over the entire image. On the other hand, since our filters have rational transfer functions, the artifacts have magnitude that decay exponentially fast away from edges. The artifacts created by our method appear as a mix of noise and jittering effects near edges. Their magnitude decreases with the order of the filter, but they also tend to spread further away from edges at the same time. We have, in some sense, to choose a tradeoff between aliasing and ringing. Since these artifacts may be considered more disturbing than blur, the practitioners not interested by reversibility, who want the best tradeoff between quality and speed, may use the decomposition in shears in combination with 1-D spline interpolation, as proposed in [26] for image rotation.

D. Computational Complexity

Resampling data between lattices essentially amounts to computing samples whose locations fall between the sample locations of the initial image. Each new value has to be calculated as a suitable weighted average of the available samples on the initial lattice, surrounding the new sample. With classical resampling methods, for each new sample, the weighting coefficients have to be computed by multiple evaluations of some interpolation kernel, and then, the weighted average itself is computed. This makes nonseparable 2-D methods of high orders impracticable, because of the increase in the size of the local neighborhood involved in the calculation of each new sample, but also because the design of high-order interpolation kernels is complex.

Even in the simplest case of bilinear interpolation on the Cartesian lattice, three multiplications and eight additions per pixel are required to convert an image on another lattice. The proposed method with $h_{\tau}^{1,N}$ requires only $3N$ multiplications and $6N+3$ additions per pixel, independently of the lattices. The computation of the filters coefficients is negligible, because it is done once for each row or column and not for every pixel. In other words, we have to compute a new 1-D filter for each row or column, versus a new 2-D filter for each pixel when using classical resampling methods.

In Table II, we report the computation time in seconds for one rotation with each proposed method, for C-code running on an Apple Mac. Dual 2.7-Ghz PowerPC G5. All of our filters up to $h_{\tau}^{2,2}$, for which we gave fast implementations, give computation times significantly reduced when compared with interpolation methods, even the simple bilinear interpolation. The filters $h_{\tau}^{2,2}$ and $h_{\tau}^{1,3}$ provide a very good tradeoff between quality and speed, and better than the reference bicubic interpolation. If better quality is desired, the filters $h_{\tau}^{1,N}$ should be used with higher N. The filters $h_{\tau}^{2,N}$ are, in practice, less interesting for $N>2$, since their implementation is not so easy and fast. However, the computation time remains reasonable in comparison with 2-D interpolation methods.

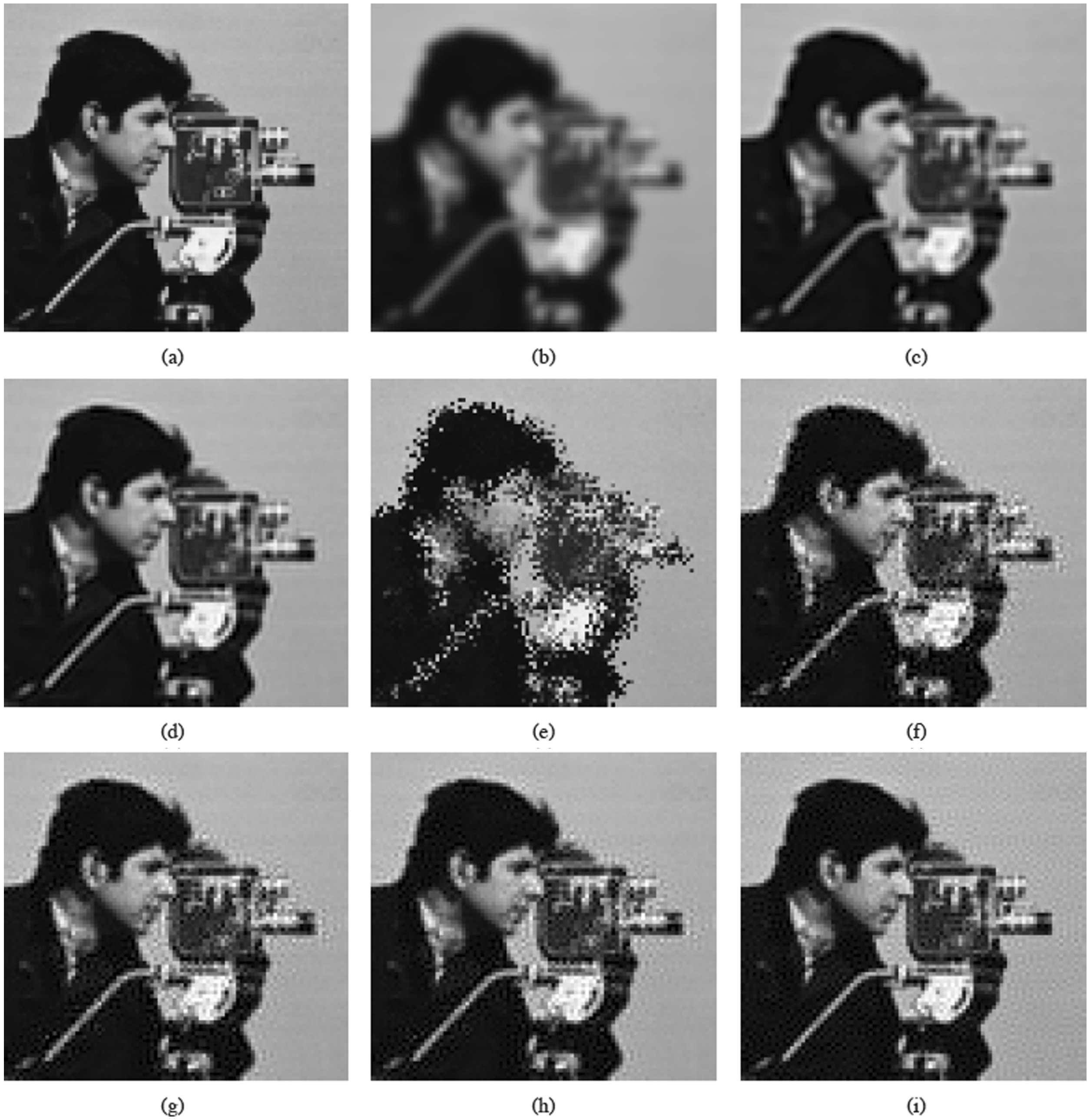

Fig. 7. Images obtained after nine rotations of angle $2\pi/9$ on a part of the *camera* image depicted in (a), using (b) bilinear, (c) bicubic, and (d) O-MOMS interpolation, and our method in three shears with filters (e) h_τ^0 (nearest neighbor), (f) $h_\tau^{1,1}$, (g) $h_\tau^{1,2}$, (h) $h_\tau^{1,3}$, (i) h_τ^∞ (sinc). The PSNRs are from (b) to (i): 21.19, 25.15, 29.77, 15.63, 23.07, 25.05, 26.02, and 28.67 dB, respectively. The images obtained with $h_\tau^{2,2}$, $h_\tau^{2,3}$, and $h_\tau^{2,4}$, not reproduced here, give PSNRs of 26.07, 27.24, and 27.93 dB, respectively.

Moreover, important savings of memory storage are possible with our approach; buffer memory can be limited to one column or row at a time. It is also important to notice that the conversion process can be performed in-place on the initial data, making the use of auxiliary memory unnecessary. Consequently, the processing of very large images is made easier. Separability also turns out to be a significant advantage for hardware implementation and parallelization.

VI. Conclusion

In this paper, we have presented an original approach for converting 2-D image data between two lattices with the same sampling density. Our resampling method is first and foremost reversible, a property not shared by the other methods of the literature. From the quality point-of-view, our approach provides decent results in comparison with the best interpolation methods, while being often faster. In conclusion, we summarize the key features of our work.

1) The decomposition of the conversion process can be done in at most three shear operations.
2) Symmetric reversibility can be enforced thanks to the combination of shear with all-pass fractional delay filters. Thus, no loss of information is introduced by the conversion process.

3) The design and implementation is reduced to 1-D translators. Short delay filters can be used, making our method substantially faster than classical interpolation methods.
4) New 1-D all-pass delay filters are proposed. No blur and good resampling quality are reported.

Last, but not least, we used for the experimental validation a new analytical differentiable phantom.

Appendix I
Synthetic Test Images

A. Zoneplate Image

The function is a 2-D frequency chirp, defined by the equation

$$f(\mathbf{x}) = 127.5 + 127.5 \cos\left(\frac{1440/\pi}{1 + 512/\sqrt{8\,(x_1^2 + x_2^2)}}\right) \tag{56}$$

where the intensity range is $[0, 255]$ and the image is defined for the domain $[-256, 255]^2$.

B. Shepp–Logan Image

We propose a blurred version of the modified Shepp–Logan phantom. The original Shepp–Logan phantom is a grayscale intensity image that consists of one large ellipse containing several smaller ellipses [50], [51]. The modified Shepp–Logan phantom is a variant of the latter, having higher contrast for better visual perception. Both phantoms are standard images used to test (medical) image processing algorithms. Example source code implementing them is provided in Matlab. The analytic function that we propose also includes a blurring effect that is determined directly in the spatial domain. Specifically, the perfect transitions at the boundaries of the ellipses are replaced by sigmoidal edges. We define the 2-D analytical function as

$$f(\mathbf{x}) = \sum_{i=1}^{10} f_i(\mathbf{x}), \tag{57}$$

where $f_i(\mathbf{x})$ represents the contribution of the ith ellipse. The ellipse's parameters are: gray level A_i, center $\mathbf{c}_i$, semi-major axis L_i, semi-minor axis l_i, and angle ϕ_i of the major axis with the horizontal axis. Its contribution is then computed as follows.

1) Determine $\mathbf{b}_i$, the point on the ellipse boundary lying on the line $(\mathbf{c}_i, \mathbf{x})$, the closest to $\mathbf{x}$. We have: $\mathbf{x} - \mathbf{b}_i = (\mathbf{x} - \mathbf{c}_i) * (1 - 1/\rho_i)$ where $\rho_i = ((v_1/L_i)^2 + (v_2/l_i)^2)^{1/2}$ and $\mathbf{v} = \mathbf{R}_{-\phi_i}(\mathbf{x} - \mathbf{c}_i)$. Note that computing the closest point to $\mathbf{x}$ on the boundary of the ellipse would be more satisfactory, but this is a much more difficult problem.
2) Use the distance from $\mathbf{b}_i$ to $\mathbf{x}$ to compute

$$f_i(\mathbf{x}) = \frac{A_i}{1 + \exp(\operatorname{sgn}(\rho_i - 1)\|\mathbf{x} - \mathbf{b}_i\|/\sigma)}. \tag{58}$$

Note that if $\mathbf{x} = \mathbf{c}_i$, then $\rho_i = 0$, and we set $\|\mathbf{x} - \mathbf{b}_i\| = (l_i + L_i)/2$.

The parameter σ controls the amount of blur. If $\sigma \to 0$, we recover the classical Shepp–Logan phantom. We propose $\sigma = 1/4$, which makes the transition at an edge spread over about three pixels. Our phantom has the advantage of being infinitely continuously differentiable (except at the centers of the ellipses), while the Shepp–Logan phantom is not even continuous. The blurred phantom is more realistic and better representative of natural images acquired with practical imaging devices.

References

[1] R. M. Mersereau, "The processing of hexagonally sampled two-dimensional signals," *Proc. IEEE*, vol. 67, no. 6, pp. 930–949, Jun. 1979.

[2] D. E. Dudgeon and M. E. Mersereau, *Multidimensional Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1984.

[3] E. Dubois, "The sampling and reconstruction of time-varying imagery with application in video systems," *Proc. IEEE*, vol. 73, no. 4, pp. 502–522, Apr. 1985.

[4] D. Van De Ville, T. Blu, M. Unser, W. Philips, I. Lemahieu, and R. Van De Walle, "Hex-spline: A novel family for hexagonal lattices," *IEEE Trans. Image Process.*, vol. 13, no. 6, pp. 758–772, Jun. 2004.

[5] L. Condat and D. Van De Ville, "Three-directional box-splines: Characterization and efficient evaluation," *IEEE Signal Process. Lett.*, vol. 13, no. 7, pp. 417–420, Jul. 2006.

[6] C. de Boor, K. Höllig, and S. Riemenschneider, *Box Splines*. Berlin, Germany: Springer-Verlag, 1993, vol. 98.

[7] L. Condat and D. Van De Ville, "Quasi-interpolating spline models for hexagonally-sampled data," *IEEE Trans. Image Process.*, vol. 16, no. 5, pp. 1195–1206, May 2007.

[8] R. A. Ulichney, *Digital Halftoning*. Cambridge, MA: MIT Press, 1987.

[9] D. P. Petersen and D. Middleton, "Sampling and reconstruction of wavenumber-limited functions in N-dimensional Euclidean spaces," *Inf. Control*, vol. 5, pp. 279–323, 1962.

[10] S. Jung, R. Thewes, T. Scheiter, K. F. Goser, and W. Weber, "Low-power and high-performance CMOS fingerprint sensing and encoding architecture," *IEEE J. Solid-State Circuit*, vol. 34, no. 7, pp. 978–984, Jul. 1999.

[11] M. Tremblay, S. Dallaire, and D. Poussart, "Low level segmentation using CMOS smart hexagonal image sensor," in *Proc. Conf. Computer Arch, for Machine Perception*, 1995, pp. 21–28.

[12] A. Rosenfeld, "Connectivity in digital pictures," *J. ACM*, pp. 146–160, 1970.

[13] R. C. Staunton, "The design of hexagonal sampling structures for image digitization and their use with local operators," *Image Vis. Comput.*, vol. 7, no. 3, pp. 162–166, 1989.

[14] S. Periaswamy, "Detection of microcalcifications in mammograms using hexagonal wavelets," M.S. thesis, Univ. South Carolina, Columbia, 1996.

[15] A. P. Fitz and R. J. Green, "Fingerprint classification using a hexagonal fast Fourier transform," *Pattern Recognit.*, vol. 29, no. 10, pp. 1587–1597, 1996.

[16] L. Middleton and J. Sivaswamy, "Edge detection in a hexagonal-image processing framework," *Image Vis. Comput.*, vol. 19, no. 14, pp. 1071–1081, Dec. 2001.

[17] Q. Wu, X. He, and T. Hintz, "Bilateral filtering based edge detection on hexagonal architecture," in *Proc. IEEE ICASSP*, 2005, vol. 2, pp. 713–716.

[18] L. Middleton and J. Sivaswamy, *Hexagonal Image Processing: A Practical Approach*. London, U.K.: Springer-Verlag, 2005.

[19] J. Serra, *Image Analysis and Mathematical Morphology*. London, U.K.: Academic, 1982.

[20] J. Serra and B. Lay, "Square to hexagonal lattice conversion," *Signal Process.*, vol. 9, no. 1, pp. 1–13, Jul. 1985.

[21] R. C. Staunton, "Hexagonal sampling in image processing," *Adv. Imag. Electron. Phys.*, vol. 107, pp. 231–307, 1999.

[22] R. C. Staunton, "One-pass parallel hexagonal thinning algorithm," *Proc. Inst. Elect. Eng.*, vol. 148, no. 1, pp. 45–53, Feb. 2001.

[23] L. N. Lester and J. Sandor, "Computer graphics on a hexagonal grid," *Comput. Graph.*, vol. 8, no. 1, pp. 401–409, 1984.

[24] G. Borgefors, "Distance transformations on hexagonal grids," *Pattern Recognit. Lett.*, vol. 9, pp. 97–105, 1989.

[25] Y.-K. Liu, "The generation of straight lines on hexagonal grids," *Comput. Graph. Forum*, vol. 12, no. 1, pp. 27–31, Mar. 1993.

[26] M. Unser, P. Thévenaz, and L. Yaroslavsky, "Convolution-based interpolation for high-quality rotation of images," *IEEE Trans. Image Process.*, vol. 4, no. 10, pp. 1371–1381, Oct. 1995.

[27] T. Toffoli, “Almost every unit matrix is a ULU,” *Linear Algebra Appl.*, vol. 259, pp. 31–38, 1997.

[28] G. Strang, “Every unit matrix is a LULU,” *Linear Algebra Appl.*, vol. 265, pp. 165–172, 1997.

[29] L. Condat, B. Forster-Heinlein, and D. Van De Ville, “H_2O: Reversible hexagonal-orthogonal grid conversion by 1-D filtering,” in *Proc IEEE Int. Conf. Image Process.*, Sep. 2007, vol. II, pp. 73–76.

[30] A. Paeth, “A fast algorithm for general raster rotation,” in *Proc. Graphics Interface*, 1986, pp. 77–81.

[31] H. Kiesewetter and A. Graf, “Rotation in digital grids and corresponding models,” Tech. Rep., Zentral Institut für Kybernetik und Informations Prozesse, Akademie der Wissenschaften der DDR, 1985.

[32] A. Tanaka, M. Kameyama, S. Kazama, and O. Watanabe, “A rotation method for raster image using skew transformation,” in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Jun. 1986, pp. 272–277.

[33] S. Srinivasan, “Modulo transforms—An alternative to lifting,” *IEEE Trans. Signal Process.*, vol. 54, no. 5, pp. 1864–1874, May 2006.

[34] R. W. Cox and R. Tong, “Two- and three-dimensional image rotation using the FFT,” *IEEE Trans. Image Process.*, vol. 8, no. 9, pp. 1297–1299, Sep. 1999.

[35] B. Chen and A. Kaufman, “Two-pass image and volume rotation,” presented at the IEEE Workshop on Volume Graphics, 2001.

[36] D. E. Friedman, “Two-dimensional resampling of line scan imagery by one-dimensional processing,” *Photogramm. Eng. Remote Sens.*, vol. 47, pp. 1459–1467, Oct. 1981.

[37] E. Catmul and A. R. Smith, “3-D transformation of images in scan-line order,” *Proc. SIGGRAPH*, vol. 14, pp. 279–285, 1980.

[38] P. Thévenaz and M. Unser, “Separable least-squares decomposition of affine transformations,” presented at the IEEE Int. Conf. Image Processing, Oct. 1997.

[39] B. Chen and A. Kaufman, “3D volume rotation using shear transformations,” *Graph. Models*, vol. 62, no. 4, pp. 308–322, 2000.

[40] A. Entezari, R. Dyer, and T. Möller, “Linear and cubic box splines for the body centered cubic lattice,” in *Proc. IEEE Visualization*, Austin, TX, Oct. 2004, pp. 11–18.

[41] T. I. Laakso, V. Välimäki, M. Karjalainen, and U. K. Laine, “Splitting the unit delay—Tools for fractional delay filter design,” *IEEE Signal Process. Mag.*, vol. 13, no. 1, pp. 30–60, Jan. 1996.

[42] C. E. Shannon, “Communication in the presence of noise,” *Proc. Inst. Radio Eng.*, vol. 37, no. 1, pp. 10–21, 1949.

[43] J. S. Welling, W. F. Eddy, and T. K. Young, “Rotation of 3D volumes by Fourier-interpolated shears,” *Graph. Models*, vol. 68, no. 4, pp. 365–370, Jul. 2006.

[44] J.-P. Thiran, “Recursive digital filters with maximally flat group delay,” *IEEE Trans. Circuit Theory*, vol. CT-18, no. 6, pp. 664–669, Nov. 1971.

[45] M. Unser, A. Aldroubi, and M. Eden, “B-spline signal processing: Part 1–Theory and Part lI–Efficient design and applications,” *IEEE Trans. Signal Process.*, vol. 41, no. 2, pp. 821–848, Feb. 1993.

[46] B. Appleton and H. Talbot, “Recursive filtering of images with symmetric extension,” *Signal Process.*, vol. 85, pp. 1546–1556, 2005.

[47] J. Kovačević and W. Sweldens, “Wavelet families of increasing order in arbitrary dimensions,” *IEEE Trans. Image Process.*, vol. 9, no. 3, pp. 480–496, Mar. 2000.

[48] R. G. Keys, “Cubic convolution interpolation for digital image processing,” *IEEE Trans. Acoust., Speech, Signal Proceess.*, vol. ASSP-29, no. 6, pp. 1153–1160, Dec. 1981.

[49] T. Blu, P. Thévenaz, and M. Unser, “MOMS: Maximal-order interpolation of minimal support,” *IEEE Trans. Image Process.*, vol. 10, no. 7, pp. 1069–1080, Jul. 2001.

[50] L. A. Shepp and B. F. Logan, “The Fourier reconstruction of a head section,” *IEEE Trans. Nucl. Sci.*, vol. 21, no. 3, pp. 21–34, Jun. 1974.

[51] A. K. Jain, *Fundamentals of Digital Image Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1989, p. 439.

Laurent Condat (S’05–M’07) received the M.S. degree in computer science from the École Nationale Supérieure d’Informatique et de Mathématiques Appliquées de Grenoble (ENSIMAG-INPG), Grenoble, France, the M.S. degree in applied mathematics from the Joseph Fourier University, Grenoble, in 2003, and the Ph.D. degree in applied mathematics from the Institut National Polytechnique de Grenoble (INPG) in 2006 for a work on approximation theory applied to signal processing reconstruction problems and image resizing.

He is currently a Postdoctorate at the Helmholtz Zentrum München—German Research Center for Environmental Health (formerly named GSF), Munich, Germany. His area of interest includes sampling and wavelet theories, image resizing, color image processing, and inverse problems.

Dr. Condat received a best student paper award at the IEEE International Conference on Image Processing in 2005 for an article on quasi-interpolation.

Dimitri Van De Ville (M’02) received the M.S. degree in engineering and computer sciences from Ghent University, Ghent, Belgium, in 1998, and the Ph.D. degree from the Medical Signal and Image Processing Group, Ghent University.

He obtained a grant as Research Assistant with the Fund for Scientific Research Flanders Belgium (FWO). In 2002, he joined Prof. M. Unser’s Biomedical Imaging Group at the École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland, where he continues his research today. In December 2005, he became responsible for the Signal Processing Antenna at the University Hospital of Geneva, Geneva, Switzerland, as part of the Centre d’Imagerie Biomédicale (CIBM). His research interests include wavelets, statistical analysis, multidimensional splines, and applications in biomedical imaging, such as functional magnetic resonance imaging, spectroscopy, electro-encephalography, and microscopy.

Dr. Van De Ville serves as an Associate Editor for the IEEE TRANSACTIONS ON IMAGE PROCESSING (since February 2006) and was previously an Associate Editor for the IEEE SIGNAL PROCESSING LETTERS (2004–2006). Since 2003, he has also been an Editor and Webmaster of *The Wavelet Digest*, a noncommercial electronic newsletter for the wavelet community with more than 22 000 subscribers. He organized the “Wavelets and Applications” semester of EPFL’s Bernoulli Center, together with M. Vetterli and M. Unser, and WavE2006, its associated international conference (July 2006).

Brigitte Forster-Heinlein (M’04) received the diploma and doctoral degree (summa cum laude) in mathematics from the Technische Universität München (TUM), Germany, in 1998 and 2001, respectively.

From 2002–2004, she was a Researcher in Prof. M. Unser’s Biomedical Imaging Group at the École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland. She is now a Junior Professor for mathematical modeling in miomedical engineering at TUM. In addition, she is the Team Leader of the Marie Curie Excellence Team MAMEBIA, funded by the European Commission, at the Helmholtz Zentrum München—German Research Center for Environmental Health. Her main research topics are Dirichlet series, complex wavelets, and applications for signal and image processing.

Dr. Forster is a member of the editoral board of the *International Journal for Mathematics and Mathematical Sciences*.