Wavelet and Shearlet-based Image Representations for Visual Servoing
Lesley-Ann Duflot, Rafael Reisenhofer, Brahim Tamadazte, Nicolas Andreff, Alexandre Krupa

To cite this version:

HAL Id: hal-01735241
https://hal.archives-ouvertes.fr/hal-01735241
Submitted on 15 Mar 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Wavelet and Shearlet-based Image Representations for Visual Servoing

Lesley-Ann Duflot$^{1,2}$, Rafael Reisenhofer$^3$, Brahim Tamadazte$^2$, Nicolas Andreff$^2$ and Alexandre Krupa$^1$

$^1$L. A. Duflot and A. Krupa are with Univ Rennes, Inria, CNRS, IRISA.
Campus universitaire de Beaulieu, Rennes 35042, France

$^2$L. A. Duflot, B. Tamadazte and N. Andreff are with FEMTO-ST, AS2M, Univ. Bourgogne Franche-Comté,
Univ. de Franche-Comté/CNRS/ENSMM, 25000 Besançon, France

$^3$R. Reisenhofer is with the University of Bremen, Computational Data Analysis,
Fachbereich 3, Postfach 330440, 28334 Bremen, Germany

Corresponding author:
lesley-ann.duflot@inria.fr

Abstract

A visual servoing scheme consists of a closed-loop control approach which uses visual information feedback to control the motion of a robotic system. Probably the most popular visual servoing method is image-based visual servoing (IBVS). This kind of method uses geometric visual features extracted from the image to design the control law. However, extracting, matching and tracking geometric visual features over time significantly limits the versatility of visual servoing controllers in various industrial and medical applications, in particular for "low-structured" medical images, e.g., ultrasounds and optical coherence tomography modalities. In order to overcome the limits of conventional IBVS, one can consider novel visual servoing paradigms known as “direct” or “featureless” approaches. This paper deals with the development of a new generation of direct visual servoing methods in which the signal control inputs are the coefficients of a multiscale image representation. In particular, we consider the use of multiscale image representations that are based on discrete wavelet and shearlet transforms. Up to now, one of the main obstacles in the investigation of multiscale image representations for visual servoing schemes was the issue of obtaining an analytical formulation of the interaction matrix that links the variation of wavelet and shearlet coefficients to the spatial velocity of the camera and the robot. In this paper, we derive four direct visual servoing controllers: two that are based on subsampled respectively non-subsampled wavelet coefficients and two that are based on the coefficients of subsampled respectively non-subsampled discrete shearlet transforms. All proposed controllers were tested in both simulation and experimental scenarios (using a 6 degrees-of-freedom (DOF) Cartesian robot in an eye-in-hand configuration). The objective of this paper is to provide an analysis of the respective strengths and weaknesses of wavelet- and shearlet-based visual servoing controllers.


Preliminary results of this work have been presented at the IEEE International Conference on Robotics and Automation
1 Introduction

1.1 Motivation

Vision-based control is a technique which uses visual features extracted from images, provided by one or multiple vision sensors, to control the motion of a robot in a closed-loop scheme (Hutchinson et al., 1996; Chaumette and Hutchinson, 2006). The past thirty years have seen rapid advances in this field with applications in industrial manipulation (Lippiello et al., 2007), medical robotics (Krupa et al., 2003; Abolmaesumi et al., 2002) or, more recently, drone navigation (Barajas et al., 2013; M"athe et al., 2016). Generally, the goal of a vision-based control law is to make a positioning task succeed by minimizing the difference between a set of desired visual features $s^\star$ and a set of current ones $s(t)$ to zero. The first visual features considered in visual servoing were geometric features such as points (Chaumette and Boukir, 1992), lines (Renaud et al., 2002) and image moments (Tahri et al., 2015). While controlling a robot using visual feedback is referred to as visual servoing, the continuous measurement of the visual features over time is known as visual tracking. However, most visual servoing approaches strongly depend on the ability to detect, extract and track visual features during the control process.

Recently, new methods emerged that allow avoiding the challenging visual tracking tasks by directly using global image information as direct signal control inputs. Accordingly, several types of global information were investigated such as image intensities (Silveira and Malis, 2012; Tamadazte et al., 2012), mutual information (Dame and Marchand, 2011), the sum of conditional variance (Richa et al., 2011), the Fourier transform (Marturi et al., 2014, 2016) or Gaussian mixtures (Crombez et al., 2015). Such techniques, which are typically referred to as direct visual servoing schemes, are often considered to be more accurate and robust than visual servoing approaches that are based on the extraction of geometric features. This is essentially due to the redundancy of visual information considered in the control loop.

The work described in this paper deals with the development of new direct visual servoing approaches where the input control signals consist of time-frequency multiscale coefficients. The latter are obtained by applying wavelet- and shearlet-based image decompositions. Generally, multiscale representations of an image are obtained by repeatedly smoothing and subsampling an image signal while also storing the detail information lost at each stage of this process. Such decompositions are not only useful for defining increasingly coarse approximations of images, but also to obtain sparse representations which means that most coefficients describing the detail information lost in each transformation step are typically close to zero.

A widely used multiscale image representation scheme is the so-called Laplacian pyramid, introduced in 1983 by Burt and Adelson (Burt and Adelson, 1983), which is based on repeated applications of a Gaussian blurring kernel. The Laplacian pyramid was later extended by Simoncelli et al. (Simoncelli and Freeman, 1995) to yield improved representations of oriented features such as curves and edges. Another way of defining multiscale image representations can be found in the realm of applied harmonic analysis, which originates from classical Fourier analysis and time-frequency analysis and has grown to be one of the major research areas in modern applied mathematics. Indeed, one of the main goals in applied harmonic analysis is to construct systems of basic building blocks that are provably optimal for describing features and structures typically occurring in certain classes of signals like “natural” images. Such systems of building blocks are often characterized by an inherent multiscale structure and thus yield efficient multiscale representations of the image signal.

Certainly, one of the "biggest success stories" of applied harmonic analysis can be found in the development of "wavelet" transforms. A wavelet transform can localize features simultaneously in the time and the
frequency domain. This is a significant advantage over the Fourier transform, which only yields information about which frequencies are present in a signal but cannot identify when they are occurring in the time domain. The first definition of a wavelet was given in 1910 by Alfred Haar (Haar, 1910). However, it was the ground-breaking work of pioneers like Ingrid Daubechies (Daubechies, 1988), Stéphane Mallat (Mallat, 1989) and Yves Meyer (Yves Meyer, 1992), in the late 1980s and early 1990s, that made discrete wavelet transforms applicable in many areas of engineering and applied mathematics and led to the development of a new generation of multiscale image representations. Wavelet-based multiscale image decomposition have since been invaluable tools in several digital image processing tasks such as compression (e.g., JPEG2000), filtering, feature detection and tracking. While wavelets are well suited for describing transient features such as singularities in 1-dimensional signals, they are not necessarily optimal for fitting the curvilinear features occurring in a 2-dimensional problem (Donoho, 2001). This motivated the investigation of multiscale transforms based on anisotropic (i.e. directionally sensitive) basis functions such as curvelets (Candes and Donoho, 2000), contourlets (Do and Vetterli, 2003) or more recently shearlets (Labate et al., 2005; Kutyniok and Labate, 2012b). These recent tools have since found various applications in image processing tasks such as image denoising (Easley et al., 2009), inpainting (King et al., 2013) or edge detection (Yi et al., 2008; Kutyniok and Petersen, 2015; Reisenhofer et al., 2016). Despite their prominent role in digital image processing, the use of multiscale coefficients as signal control inputs in a vision-based control scheme has only recently been considered. A 6 DOF visual servoing approach based on the low-pass approximation of an image obtained from a discrete wavelet transform was proposed in (Ourak et al., 2016a,b). In parallel, the shearlet transform was considered in the context of medical image-guided tasks, that is, a 6 DOF ultrasound-based visual servoing method was proposed for automatically positioning an ultrasound probe carried by a robotic arm (Duflot et al., 2016a,b). In this previous work, the link between the time-variation of the shearlet coefficients and the probe velocity was obtained by a numerical approximation method. These preliminary investigations already demonstrate the feasibility and the potential benefits of considering such visual information in a visual servoing scheme, in particular in unfavorable conditions (image noise, partial image occlusions, illumination changes). In fact, using multiscale image decompositions could serve as an interesting compromise between purely geometric feature-based approaches and image-intensity-based (e.g., photometry) ones. The coarse approximation of an image yielded by a multiscale representation scheme is basically a smoothed version of the original image and thus robust to noise. In addition, the detail coefficients obtained at the different decomposition levels are highlighting basic image features, such as edges or curves, and can thus be seen as implicit visual feature detectors.

1.2 Summary of Contributions

Up to now, one of the main obstacles in the investigation of multiscale image decompositions in visual servoing fields was the issue of obtaining an analytical formulation of the interaction model that links the variation of the wavelet or shearlet coefficients to the camera/robot spatial velocity. This interaction model is well-known in the visual servoing community as the “interaction matrix”. In this paper, we analytically derive interaction matrices for control laws based on both wavelet and shearlet coefficients. That is, the variation of the wavelet respectively shearlet coefficients in the visual feature vector is analytically linked to the camera velocity twist vector. In both the wavelet and the shearlet cases, we are considering subsampled and non-subsampled transforms. The difference between subsampled and non-subsampled transforms will be discussed in Section 2. In total, we propose four visual control laws which are:

- subsampled wavelet-based control law (s-wavelet)
• subsampled shearlet-based control law (s-shearlet)
• non-subsampled wavelet-based control law (ns-wavelet)
• non-subsampled shearlet-based control law (ns-shearlet)

Moreover, all proposed control laws were tested in both simulation and experimental scenarios (using a 6-DOF Cartesian robot in an eye-in-hand configuration). The objective was to provide a wide qualitative and quantitative analysis of their respective strengths and weaknesses. In particular, numerous experiments under favorable (nominal conditions) and unfavorable (partial occlusions, unstable illumination) working conditions were carried out. It can be highlighted that wavelet as well as shearlet-based visual servoing approaches provide good performances with regards to accuracy and robustness to external disturbances. Furthermore, experimental evidence suggests that the control laws based on subsampled transforms outperform those that are based on non-subsampled ones.

1.3 Plan
Section 2 gives a short introduction to wavelet- and shearlet-based multiscale image representations. Section 3 begins with a review of the well-known direct visual servoing approach in which image intensities are used as the only visual features and describes how it can be modified to obtain wavelet- and shearlet-based control laws. In particular, Section 3 discusses the mathematical formulation of the respective wavelet and shearlet-based interaction matrices. Section 4 presents the results obtained from several simulations and experimental validations under different working conditions, the comparison with the photometry visual servoing as well as a discussion of the performances of each proposed method.

1.4 Notation
We use bold non-italic letters to denote vectors (e.g. $s$) and bold non-italic uppercase letters to denote matrices (e.g. $A$). Scalar values and continuous-time functions will be denoted by italic letters (e.g. $x$, resp. $f$). Single entries of a vector or a matrix will be denoted using subscript indexes and non-bold italic letters (e.g. $h_n$, resp. $A_{n,m}$). We use $A_t$ to denote a matrix that is parameterized by $k$ and $A(t)$ to denote a matrix-valued continuous-time function. The same notation will be applied in the case of parameterized vectors and vector-valued continuous-time functions. To ensure a consistent discrimination of continuous-time and discrete objects, we also use the just described vector and matrix notation for elements of the spaces of square-summable one- and two-dimensional sequences $l^2(Z)$ and $l^2(Z^2)$. For a better understanding of the different symbols used in this paper, the reader may refer to the following Table 1.

<table>
<thead>
<tr>
<th>Notations</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathbb{N}$ and $\mathbb{N}_0$</td>
<td>natural numbers set without and with 0, respectively</td>
</tr>
<tr>
<td>$L^2(\mathbb{R}^n)$</td>
<td>space of real-valued square-integrable $n$-dimensional functions</td>
</tr>
<tr>
<td>$l^2(\mathbb{Z})$</td>
<td>space of square-summable sequences</td>
</tr>
<tr>
<td>$</td>
<td>.</td>
</tr>
<tr>
<td>$\lceil . \rceil$</td>
<td>ceiling function</td>
</tr>
<tr>
<td>$\langle , \rangle$</td>
<td>$L^2$-inner product</td>
</tr>
<tr>
<td>$\bar{x}$</td>
<td>complex conjugate of $x$</td>
</tr>
<tr>
<td>$\det A$</td>
<td>determinant of a matrix $A$</td>
</tr>
<tr>
<td>$D_2$ and $T_m$</td>
<td>dyadic dilation and translation operators on $L^2(\mathbb{R}^2)$, respectively</td>
</tr>
<tr>
<td>$\phi_{m}^{(1)}$</td>
<td>scaling function in $L^2(\mathbb{R}^2)$</td>
</tr>
</tbody>
</table>
Before discussing the derivation of the proposed direct visual servoing control laws, we will give a short in-
of visual servoing, especially in low structured images (e.g. medical images).

features in the control loop that correspond to redundant information improves the accuracy and robustness
the extraction, matching and tracking of specific features over time. In addition, using a wide set of visual
used to design the associated interaction matrices consist of global image information, thereby avoiding
this paper belong to the family of direct visual servoing approaches. This means that the visual information
inputs are the coefficients of wavelet- and shearlet-based image representations. The control laws derived in

Our goal is the design of original, accurate and robust visual servoing schemes in which the control signal
thermore, several open-source libraries implementing discrete wavelet and shearlet transforms are available.
which are well established mathematical tools widely used in signal and image processing applications. Fur-

2 Wavelet- and Shearlet-based Image Representations

In this paper, we focus on the investigation of multiscale image representations in the context of visual
ervoing. Two representations are considered, namely wavelet and shearlet-based image decompositions,
which are well established mathematical tools widely used in signal and image processing applications. Fur-
thermore, several open-source libraries implementing discrete wavelet and shearlet transforms are available.
Our goal is the design of original, accurate and robust visual servoing schemes in which the control signal
inputs are the coefficients of wavelet- and shearlet-based image representations. The control laws derived in
this paper belong to the family of direct visual servoing approaches. This means that the visual information
used to design the associated interaction matrices consist of global image information, thereby avoiding
the extraction, matching and tracking of specific features over time. In addition, using a wide set of visual
features in the control loop that correspond to redundant information improves the accuracy and robustness
of visual servoing, especially in low structured images (e.g. medical images).

Before discussing the derivation of the proposed direct visual servoing control laws, we will give a short in-

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\psi_{j,m}$</td>
<td>horizontal wavelet in $L^2(\mathbb{R}^2)$ at scale $j$ and shifted by $m$</td>
</tr>
<tr>
<td>$\psi_{j,m}$</td>
<td>vertical wavelet in $L^2(\mathbb{R}^2)$ at scale $j$ and shifted by $m$</td>
</tr>
<tr>
<td>$\psi_{j,m}$</td>
<td>diagonal wavelet in $L^2(\mathbb{R}^2)$ at scale $j$ and shifted by $m$</td>
</tr>
<tr>
<td>$\phi$</td>
<td>scaling function in $L^2(\mathbb{R})$</td>
</tr>
<tr>
<td>$\psi$</td>
<td>generating wavelet in $L^2(\mathbb{R})$</td>
</tr>
<tr>
<td>$W_qf$</td>
<td>wavelet transform of the function $f \in L^2(\mathbb{R}^2)$</td>
</tr>
<tr>
<td>$h$ and $g$</td>
<td>discrete low-pass and high-pass filters, respectively</td>
</tr>
<tr>
<td>$\psi_{j,k,m}$</td>
<td>vertical shearlet in $L^2(\mathbb{R}^2)$ at scale $j$, with shearing $k$ and shifted by $m$</td>
</tr>
<tr>
<td>$\psi_{j,k,m}$</td>
<td>horizontal shearlet in $L^2(\mathbb{R}^2)$ at scale $j$, with shearing $k$ and shifted by $m$</td>
</tr>
<tr>
<td>$D_M$</td>
<td>general geometric transformation operator on $L^2(\mathbb{R}^2)$</td>
</tr>
<tr>
<td>$D_A$</td>
<td>dilation operator on $L^2(\mathbb{R}^2)$ with scaling matrix $A$</td>
</tr>
<tr>
<td>$D_S$</td>
<td>shear operator on $L^2(\mathbb{R}^2)$ with shear matrix $S_k$</td>
</tr>
<tr>
<td>$S_wf$</td>
<td>shearlet transform of the function $f \in L^2(\mathbb{R}^2)$</td>
</tr>
<tr>
<td>$r$ and $r^*$</td>
<td>current and desired pose of the robot, respectively</td>
</tr>
<tr>
<td>$l$</td>
<td>image observed at time $t$</td>
</tr>
<tr>
<td>$l_t(x,y)$</td>
<td>image intensity at the point $(x,y)$ in the image plane at time $t$</td>
</tr>
<tr>
<td>$s_i$ (i = [ph, w, s])</td>
<td>photometric, wavelet and shearlet visual features, respectively</td>
</tr>
<tr>
<td>$e_i$ (i = [ph, w, s])</td>
<td>photometric, wavelet and shearlet visual error vector, respectively</td>
</tr>
<tr>
<td>$v$</td>
<td>velocity twist vector of the camera frame</td>
</tr>
<tr>
<td>$L_{s}$</td>
<td>interaction matrix in the photometric case</td>
</tr>
<tr>
<td>$(L_{s}^+)$ Moore-Penrose pseudo-inverse of the interaction matrix $L_{s}$</td>
<td></td>
</tr>
<tr>
<td>$L_{w}$ and $L_{w}^+$ wavelet-based interaction matrix and its approximation, respectively</td>
<td></td>
</tr>
<tr>
<td>$L_{s}$ and $L_{s}^+$ shearlet-based interaction matrix and its approximation, respectively</td>
<td></td>
</tr>
<tr>
<td>$L_{p}(x,y)$ interaction matrix related to the point $(x,y)$</td>
<td></td>
</tr>
<tr>
<td>$L_{pw}(x,y)$ interaction matrix related to the point $(x,y)$ in the wavelet case</td>
<td></td>
</tr>
<tr>
<td>$L_{psh}(x,y)$ interaction matrix related to the point $(x,y)$ in the shearlet case</td>
<td></td>
</tr>
<tr>
<td>$Z$</td>
<td>depth of the image plane</td>
</tr>
<tr>
<td>$\lambda$</td>
<td>positive gain parameter</td>
</tr>
<tr>
<td>$\mu$</td>
<td>Levenberg-Marquardt damping parameter</td>
</tr>
</tbody>
</table>

Table 1: List of symbols.
roduction to the basic concepts and notation associated with wavelet- and shearlet-based transforms. While discrete wavelet transforms have been intensively studied, and applied since the late 1980s, shearlets were first proposed in 2005 (Labate et al., 2005). In fact, the shearlet image decomposition can be considered as an extension of the wavelet case. Systems of wavelet and shearlet functions are both constructed by shifting and dilating a finite number of locally oscillating (hence the name "wavelet") generator functions. However, the definition of wavelet- and shearlet-based transforms significantly differ with regards to the type of scaling operator used for dilating a generator function. Wavelet transforms are based on isotropic scaling, that is, when dilating a generator function, both dimensions are being scaled equally, while shearlets are associated with an anisotropic (i.e. directionally dependent) scaling operator that dilates one direction more than the other. Anisotropic scaling causes high-frequency shearlets to be strongly directionally sensitive and, in turn, makes it necessary to introduce a third rotation-like operator that acts on shearlet generators by changing their preferred orientation. In the shearlet framework, this operator is chosen to be the shear operator (hence the name "shearlet"). In fact, it can be shown that for a certain type of images, so-called cartoon-like images, transforms based on anisotropic scaling operators yield optimally sparse image representation (Candès and Donoho, 2004; Guo and Labate, 2007; Kutyniok and Lim, 2011).

Another major difference between wavelets and shearlets can be found in the respective numbers of coefficients that define a wavelet- or shearlet-based representation of a digital image. In contrast to shearlet transforms, discrete wavelet transforms are naturally associated with a powerful subsampling scheme, which makes it possible to keep the number of wavelet coefficients equal to the original number of pixel values. This is due to the elegant structure of the multiresolution analysis (MRA) framework (Mallat, 1989), whose desirable properties cannot be easily generalized to the shearlet case (Han et al., 2011). Even in the case of non-subsampled transforms, shearlet transforms yield representations that have a significantly higher degree of redundancy than wavelet transforms. This means that there is a possible trade-off between the computational efficiency of the subsampled wavelet transform and the superior sparsity properties of shearlet-based transforms. These proprieties will be thoroughly discussed in Section 4, where we experimentally investigate the differences between both subsampled and non-subsampled wavelet- and shearlet-based visual servoing schemes.

2.1 Wavelet-Based Transforms

Wavelet transforms in the space of square-integrable real-valued 2-dimensional functions \( L^2(\mathbb{R}^2) \) can be constructed by shifting and dyadically scaling certain generating functions. These generating functions are defined as the tensor products of a 1-dimensional scaling function \( \phi \in L^2(\mathbb{R}) \) and a 1-dimensional wavelet \( \psi \in L^2(\mathbb{R}) \). While the scaling function \( \phi \) is used to define a “coarse approximation” of a given signal, the function \( \psi \), often called the mother wavelet, can be used to encode the “detail information” of a signal at different stages of a multilevel processing scheme.

Let us define two operators on \( L^2(\mathbb{R}^2) \), namely the dyadic dilation operator

\[
D_2 f(x, y) = 2f(2x, 2y),
\]

and the translation operator

\[
T_m f(x, y) = f(x - m_1, y - m_2),
\]

where \((x, y)\) represent the metric coordinates of an image point, \((m_1, m_2)\) a translation vector and \(f\) a function in \(L^2(\mathbb{R}^2)\). The dyadic scaling factor 2 is associated with the concept of an MRA, which will briefly be described later in this section. In particular, it allows for an efficient implementation of the subsampled discrete wavelet transform via the so-called fast wavelet transform (FWT) (Beylkin et al., 1991).
A 2-dimensional wavelet system can be constructed as follows:
\[
\begin{align*}
\left\{ \phi_{m}^{(1)} = T_{m} \phi^{(1)} : m \in \mathbb{Z}^{2} \right\} \cup \\
\left\{ \psi_{j,m}^{(l)} = D_{2}^{l} T_{m} \psi^{(l)} : j \in \mathbb{N}_{0}, m \in \mathbb{Z}^{2}, l \in \{1, 2, 3\} \right\},
\end{align*}
\]

where \( \phi^{(1)} \) denotes the low-pass wavelet generating function, \( \psi^{(1)}, \psi^{(2)}, \psi^{(3)} \) denote the vertical, horizontal and diagonal detail wavelet generating functions respectively, \( j \) is a scaling parameter and \( m \) a translation parameter. Note that \( j \) determines how often the dyadic scaling operator \( D_{2} \) is applied to a generating wavelet. Broadly speaking, choosing \( j \) to be large will yield an extremely squeezed wavelet that is tuned to pick up the high-frequency components of an image. The effect of repeatedly applying \( D_{2} \) is visualized in the case of digital wavelet filters in Fig. 3a.

The 2-dimensional wavelet generators are given by
\[
\begin{align*}
\phi^{(1)}(x,y) &= \phi(x)\phi(y), \\
\psi^{(1)}(x,y) &= \phi(x)\psi(y), \\
\psi^{(2)}(x,y) &= \psi(x)\phi(y), \\
\psi^{(3)}(x,y) &= \psi(x)\psi(y),
\end{align*}
\]

where \( \phi \in L^{2}(\mathbb{R}) \) is a 1-dimensional scaling function and \( \psi \in L^{2}(\mathbb{R}) \) a 1-dimensional mother wavelet. In particular, for generating wavelets \( \psi^{(1)}, \psi^{(2)} \) and \( \psi^{(3)} \) defined as above, the family
\[
\left\{ \psi_{j,m}^{(l)} = D_{2}^{l} T_{m} \psi^{(l)} : j \in \mathbb{Z}, m \in \mathbb{Z}^{2}, l \in \{1, 2, 3\} \right\}
\]
forms an orthonormal basis for \( L^{2}(\mathbb{R}^{2}) \). Two popular examples of 1-dimensional mother wavelets that can be used to define orthonormal bases for \( L^{2}(\mathbb{R}^{2}) \) are the Haar wavelet (Haar, 1910) and the family of Daubechies wavelets (Daubechies, 1988), depicted in Fig. 1 alongside their corresponding scaling functions.

![Wavelet Functions](image)

Figure 1: Different mother wavelet functions and their corresponding scaling functions.

For a square-integrable function \( f \in L^{2}(\mathbb{R}^{2}) \), its wavelet transform \( \mathcal{W}_{\psi}f \) with respect to a “generating function” \( \psi \in L^{2}(\mathbb{R}^{2}) \), is given by the \( L^{2} \)-inner products
\[
(\mathcal{W}_{\psi}f)(j,m) = \langle f, \psi_{j,m} \rangle = \iint_{\mathbb{R}^{2}} f(x,y) \ D_{2}^{l} T_{m} \psi(x,y) \ dx dy,
\]
where \( j \in \mathbb{Z} \) defines the scale and \( m \in \mathbb{Z}^{2} \) the location of the wavelet \( \psi_{j,m} \). Note that squeezing and stretching the generator \( \psi \) by applying the dyadic scaling operator \( D_{2}^{l} \) changes the frequency and the degree of
localization at the same time. One can note that higher frequencies correspond to a higher degree of localization and vice-versa. This behavior characterizes the fundamental difference between wavelet-based transforms and the short-time Fourier transform (Allen, 1977), in which the window specifying the degree of localization in the time-domain remains fixed.

Up to this point, we have only considered wavelet transforms in the continuous setting. While the derivation of wavelet- and shearlet-based visual servoing schemes in Section 3 will also be carried out in the continuous realm, any real-world implementation will eventually be based on discrete and finite input signals. It is thus important to ensure a faithful transition between the theory established in the continuum and its implementation in the discrete realm. A powerful framework, which connects the continuous theory of wavelets with discrete filter-based multiscale decompositions of images is the so-called MRA, which was introduced by Mallat and Meyer (Mallat, 1989). In an MRA framework, the scaling function \( \phi \in L^2(\mathbb{R}) \) and the mother wavelet \( \psi \in L^2(\mathbb{R}) \) are chosen to satisfy the scaling relations

\[
\phi(x) = \sqrt{2} \sum_{n \in \mathbb{Z}} h_n \phi(2x - n) \tag{7}
\]

\[
\psi(x) = \sqrt{2} \sum_{n \in \mathbb{Z}} g_n \phi(2x - n), \tag{8}
\]

where \( h \in \ell^2(\mathbb{Z}) \) represents a discrete low-pass filter and \( g \in \ell^2(\mathbb{Z}) \) a discrete high-pass filter.

By viewing a given discrete image \( I \in \ell^2(\mathbb{Z}^2) \) as a coarse approximation of a square-integrable function \( I \) depending on the scaling function \( \phi(1) \) at a fixed scale \( j \in \mathbb{Z} \), that is,

\[
I(x,y) \approx \sum_{n \in \mathbb{Z}^2} I_{n_1,n_2} 2^{j} \phi(1)(2^j x - n_1, 2^j y - n_2), \tag{9}
\]

the MRA framework makes it possible to compute the wavelet coefficients of the image \( I \), which is defined in the continuum, by successively convolving the discrete image \( I \) with the filters \( h \) and \( g \).

Fig. 2 depicts the first and second stage of a wavelet-based multiscale decomposition of a digital image. While the number of coefficients remains basically constant, the multiscale representation contains both a coarse approximation of the original image as well as several levels of details (horizontal, vertical and diagonal).

It should be highlighted that wavelet transforms are not optimal to model anisotropic features that play an important role in 2-dimensional signals such as images. This is due to the isotropic nature of the dilation operator \( D_2 \) defined in (1). As \( D_2 \) is treating both dimensions equally, high-frequency (i.e. highly squeezed) 2-dimensional wavelets are not well suited for fitting anisotropic (i.e. directionally dependent) singularities like edges. As we already noted, this drawback can be remedied by considering anisotropic dilation operators. One important class of functions based on anisotropic dilations has been developed in the theory of shearlets, whose construction and properties will be discussed in the following Section 2.2.

### 2.2 Shearlet-Based Transforms

A system of shearlets, which were first introduced in (Labate et al., 2005), can be obtained by anisotropically dilating, shifting and shearing a finite number of generating functions. Formally, these operations are carried out in the space of square-integrable functions \( L^2(\mathbb{R}^2) \) by applying the translation operator already defined in (2) and the general geometric transformation operator \( D_M \) given by

\[
D_M f(x) = |\det M|^{1/2} f(Mx), \text{ where } M \in \mathbb{R}^{2 \times 2}, \tag{10}
\]
cone invariant under the shear operator for any transform (Candes and Donoho, 2000) is a significant advantage for discretization, as the integer lattice is used for wavelets, and anisotropic scaling, which is applied in the construction of shearlet-based systems, is invariant under the shear operator for any \( k \in \mathbb{Z} \).

The factors 2 and \( \sqrt{2} \) define a so-called *parabolic* scaling operator \( D_A \). There exist generalizations in which \( \sqrt{2} \) can be replaced with an exponent \( \alpha \in [0, 1] \) (Grohs et al., 2016) and which are not restricted to the dyadic scaling factor 2 (Genzel and Kutyniok, 2014). However, these generalizations are beyond the scope of this work and will not be considered further in this paper. The difference between isotropic scaling, which is used for wavelets, and anisotropic scaling, which is applied in the construction of shearlet-based systems, is illustrated in Fig. 3. Using the shear operator \( D_{S_k} \) instead of the rotation operator applied by the curvelet transform (Candes and Donoho, 2000) is a significant advantage for discretization, as the integer lattice is invariant under the shear operator for any \( k \in \mathbb{Z} \).

For a 2-dimensional scaling function \( \phi^{(1)} \in L^2(\mathbb{R}^2) \) and generating shearlets \( \psi^{(1)}, \psi^{(2)} \in L^2(\mathbb{R}^2) \), a so-called *cone-adapted* shearlet system can be defined analogously to the wavelet system (3) as the following union:

\[
\begin{align*}
\left\{ \phi_m^{(1)} = T_m \phi^{(1)} : m \in \mathbb{Z}^2 \right\} & \cup \\
\left\{ \psi_{j,k,m}^{(1)} = D_A^j D_{S_k} T_m \psi^{(1)} : j \in \mathbb{N}_0, |k| < \left\lfloor 2^j \right\rfloor, m \in \mathbb{Z}^2 \right\} & \cup \\
\left\{ \psi_{j,k,m}^{(2)} = D_A^j D_{S_k} T_m \psi^{(2)} : j \in \mathbb{N}_0, |k| < \left\lfloor 2^j \right\rfloor, m \in \mathbb{Z}^2 \right\},
\end{align*}
\]

Figure 2: Wavelet decompositions of a digital image at level 1 and 2. Each pixel in 2b and 2c corresponds to the inner product of the image 2a with a discrete wavelet filter. The coefficients associated with the low-pass scaling function \( \phi^{(1)} \) define a coarse approximation of the original image. The coefficients associated with the high-pass wavelet functions \( \psi^{(1)}, \psi^{(2)}, \text{and } \psi^{(3)} \) highlight horizontal, vertical and diagonal edges, respectively. Wavelet transforms often yield sparse representations of images, which is indicated by the large number of zeros in 2b and 2c.
Figure 3: Comparison of discrete 2D wavelet and shearlet filters at different scales. Due to the anisotropic scaling, the high-frequency shearlet filters in the last row of 3b are elongated and better adapted to fit edges in images than the high-frequency wavelet filters in the last row of 3a. The frequency spectrum of each filter is depicted in the right columns of 3a and 3b.

with \([\cdot]\) denoting the ceiling function. Note that the condition \(|k| < \left\lfloor 2^j \right\rfloor \) prohibits shears of the generating shearlet that would change the preferred orientation by more than 45°. This forces the essential support of a shearlet in the frequency domain to remain within the cones in which the corresponding generating shearlet is located (hence the name “cone-adapted”, cf. Fig. 4). In turn, this means that for a cone-adapted shearlet system to cover the whole frequency plane, it is important to choose the generating shearlets \(\psi^{(1)}\) and \(\psi^{(2)}\) such that they are supported in different frequency cones. In particular, this can be ensured by defining the functions \(\phi^{(1)}, \psi^{(1)}\) and \(\psi^{(2)}\) analogously to equation (4) as tensor products of 1-dimensional scaling and wavelet functions, which leads to the construction of so-called separable shearlet systems. The tiling of the horizontal frequency cones induced by the system (13) is schematically depicted in Fig. 4.

Figure 4: The tiling of the horizontal frequency cones induced by the shearlet system (13). \(\xi_1\) and \(\xi_2\)

Like in the wavelet case, the shearlet-based transform \(S_y f\) of a square-integrable 2-dimensional function

\[
j = 0 \quad \text{(a) Discrete 2D wavelet filters.} \\
j = 1 \quad \text{(b) Discrete 2D shearlet filters.}
\]
\( f \in L^2(\mathbb{R}^2) \) with respect to a generating function \( \psi \in L^2(\mathbb{R}^2) \) that is essentially supported in the horizontal frequency cones is given at a scale \( j \in \mathbb{Z} \), with a shear parameter \( |k| < 2^j \) and a translation parameter \( m \in \mathbb{Z}^2 \) by the inner products

\[
(S_{\psi}f)(j, k, m) = \langle f, \psi_{j,k,m} \rangle = \int_{\mathbb{R}^2} f(x, y) D_A^{j} D_{S_k} T_m \psi(x, y) \, dx \, dy.
\]  

(14)

For shearlet generators that are in the vertical frequency cone, \( A \) and \( S_k \) have to be replaced in (14) with \( \tilde{A} \) and \( S_k^\top \) respectively. The reader is invited to refer to (Kutyniok and Labate, 2012a) for more details about the theory and applications of shearlet-based transforms.

During the past decade, several libraries implementing different types of shearlet transforms for finite and discrete data have been developed

- Fast Finite Shearlet Transform (FFST) (Häuser and Steidl, 2012), which implements non-subsampled transforms based on band-limited shearlets,
- ShearLab (Kutyniok et al., 2012), which implements a subsampled transform based on separable compactly supported shearlets,
- ShearLab 3D (Kutyniok et al., 2016), which contains both two- and three-dimensional non-subsampled shearlet transforms that are based on non-separable and compactly supported shearlets.

Fig. 5 shows the coefficients of a non-subsampled shearlet-based transform of a digital image after two stages of decomposition. Comparing Fig. 2 and Fig. 5 also illustrates the difference between subsampled and non-subsampled decompositions. The main advantage of subsampled shearlet- and wavelet-based transforms is that the total number of coefficients is almost equal to the original number of pixel values, while non-subsampled transforms often introduce a significant amount of redundancy.

![Figure 5](image)

Figure 5: Non-subsampled shearlet decomposition at level 2 of the 256 \( \times \) 256 digital image shown in Fig. 2a. The parameter \( j \) denotes the scale of the corresponding shearlet filter while the shear parameter \( k \) controls its orientation. All coefficient matrices have the same size as the input image. The depicted decomposition is thus highly redundant with a redundancy factor of 13.
3 Direct Visual Servoing Schemes

3.1 Photometric-based Visual Servoing

Before discussing wavelet- and shearlet-based visual servoing control laws, we will briefly revisit the well-known photometric-based visual servoing approach in order to recall some basic concepts and notation. The objective of a visual servoing scheme is to control the motion of a robot such that a set of \( N \in \mathbb{N} \) visual features \( s \in \mathbb{R}^N \) depending on the robot’s pose \( \mathbf{r}(t) \in SE(3) \) at time \( t \) matches a set of desired features \( s^* \in \mathbb{R}^N \) obtained at the desired pose \( \mathbf{r}^* \). While the feature vector \( s \) should technically be viewed as a function of the robot pose, to simplify our notation, we will treat the set of visual features as a function of time by setting \( s(t) := s(\mathbf{r}(t)) \).

For the remainder of this paper, we will use \( I(x,y,t) \in \mathbb{R}_{\geq 0} \) to denote the image intensities at the coordinates \((x,y)\) observed at time \( t \). Furthermore, we write \( I_t(x,y) := I(x,y,t) \) to denote the 2-dimensional image observed at time \( t \). This notation will be helpful in unambiguously stating mathematical expressions that contain operations which are only defined for 2-dimensional functions such as 2D wavelet and shearlet transforms. The control laws derived in this paper will eventually be based on the spatial gradient \( \nabla I_t \) as well as wavelet and shearlet transforms of the 2-dimensional image \( I_t \), that is, \( \mathcal{W}_q I_t \) and \( S_q I_t \). Note that formally, these objects are only defined in the continuum. Consequently, the corresponding visual servoing schemes described in this section are also derived in the continuous realm. In practice, however, we cannot observe the complete continuous-time image \( I_t \) but only a discretized approximation \( \mathbf{I}_t \) that is typically obtained by spatial averaging and sampling. This means that in any actual implementation of the proposed control laws, the spatial gradient as well as the wavelet- and shearlet-based transforms of \( I_t \) need to be approximated in the discrete realm. This can be achieved by convolving \( \mathbf{I}_t \) with gradient filters and by computing the discrete wavelet and shearlet transforms of \( \mathbf{I}_t \).

In photometric visual servoing, introduced in (Collewet and Marchand, 2011), the visual feature vector at time \( t \) can be considered as a finite set of individual image intensities sampled at successive 2D points \((x_n,y_n)_{n \leq N} \subset \mathbb{R}^2\):

\[
s_{ph}(t) = \left( I(x_1,y_1,t), I(x_2,y_2,t), \ldots, I(x_N,y_N,t) \right)^\top.
\]

In order to reach the desired robot’s pose \( s^* \), a control law is applied to minimize the visual error given by

\[
e_{ph}(t) = s_{ph}(t) - s_{ph}^*
\]

towards zero over time by moving the robotic system. In order to find a camera velocity twist vector \( \mathbf{v} = (v_x, v_y, v_z, \omega_x, \omega_y, \omega_z)^\top \) which decreases the error, it is necessary to link the time-variation of the visual features \( s_{ph} \) (defined in the image frame) to the camera movement. In (Collewet and Marchand, 2011), the authors formalize this relationship by linearly describing the time-derivative of the image intensities in terms of the camera spatial velocities via a so-called interaction matrix \( \mathbf{L}_{s_{ph}} \). Thereby, it becomes possible to relate the camera motion (expressed in the camera frame) to the image intensities through

\[
\frac{ds_{ph}(t)}{dt} = \mathbf{L}_{s_{ph}}(t) \mathbf{v}(t).
\]

By assuming temporal luminance constancy and applying the optical flow constraint equation (OFCE) introduced in (Horn and Schunck, 1981), we can write

\[
\frac{\partial I(x,y,t)}{\partial t} = -\left( \nabla I_t(x,y) \right)^\top \left( \frac{dx}{dt}, \frac{dy}{dt} \right)^\top.
\]
Using the velocity twist vector, the relation (18) can be written as
\[
\frac{\partial I(x,y,t)}{\partial t} = -\left(\nabla I(x,y)\right)^T L_p(x,y)v(t),
\]
(19)
where \( L_p(x,y) \) denotes the interaction matrix of the 2D image point \((x,y) \subset \mathbb{R}^2\) corresponding to the perspective projection of a 3D point of the scene into the image plane, as proposed in (Chaumette and Hutchinson, 2006). The matrix \( L_p \) is defined as
\[
L_p(x,y) = \begin{bmatrix}
-\frac{1}{Z} & 0 & \frac{x}{Z} & xy & -(1+x^2) & y \\
0 & -\frac{1}{Z} & \frac{y}{Z} & 1+y^2 & -xy & -x
\end{bmatrix}, \tag{20}
\]
where \( Z \in \mathbb{R} \) is the depth of the observed 3D point expressed in the camera Cartesian frame. The depth \( Z \) is generally approximated as constant and equal for all the image points. Other variants were reported in the literature that consist of estimating \( Z \) at the current or desired positions or by considering an average value.

Merging (17) and (19), we can explicitly compute the interaction matrix \( L_{sph} \) at a point in time \( t \) as follows:
\[
L_{sph}(t) = \begin{bmatrix}
\left(\nabla I(x_1,y_1)\right)^T L_p(x_1,y_1) \\
\vdots \\
\left(\nabla I(x_N,y_N)\right)^T L_p(x_N,y_N)
\end{bmatrix}. \tag{21}
\]
Finally, the camera velocity twist vector at a time \( t \) is obtained by
\[
v(t) = -\lambda \left( L_{sph}(t) \right)^+ e_{ph}(t), \tag{22}
\]
where \( \lambda > 0 \) is a control gain and \( (L_{sph}(t))^+ \) denotes the Moore-Penrose pseudo-inverse of the interaction matrix \( L_{sph} \).

To increase the domain of convergence, the traditional Gauss-Newton optimization approach (22) can be replaced by the Levenberg-Marquardt method. Thus, the control law (22) becomes
\[
v(t) = -\lambda \left( H(t) + \mu \text{diag}(H(t)) \right)^{-1} \left( L_{sph}(t) \right)^T e_{ph}(t), \tag{23}
\]
where \( H(t) = (L_{sph}(t))^T L_{sph}(t) \) and \( \text{diag}(H(t)) \) denotes the diagonal matrix given by the diagonal entries of \( H(t) \). Note that a high value for \( \mu \) (e.g., \( \mu = 1 \)) yields a gradient descent method, whereas choosing \( \mu \) close to zero (typically \( \mu = 10^{-3} \)) is equivalent to the Gauss-Newton optimization approach.

### 3.2 Wavelet-based Visual Servoing

Instead of the pixel intensities used in (15), we will now consider the design of a visual servoing scheme in which the visual features are defined as the coefficients of a wavelet-based multiscale representation of an image.

Let us consider a finite set of \( L \in \mathbb{N} \) generating wavelets \( \{\psi_{l, j_n, m_n}\}_{n \leq N} \subset L^2(\mathbb{R}^2) \), which typically consists of a total of four separable generators constructed from one-dimensional wavelet and scaling functions as defined in (4). The wavelet-based feature vector for a set of \( N \in \mathbb{N} \) triples \( \{l_n, j_n, m_n\} \_{n \leq N} \subset \{1, \ldots, L\} \times \mathbb{N} \times \mathbb{Z}^2 \),
where \( l_n \) specifies a wavelet generator, \( j_n \) defines the scale and \( m_n \) the translation parameter is given at a time \( t \) as
\[
s_w(t) = \left( (\mathcal{W}_{\psi(1)} I_1)(j_1, m_1), (\mathcal{W}_{\psi(2)} I_2)(j_2, m_2), \ldots, (\mathcal{W}_{\psi(N)} I_N)(j_N, m_N) \right)^T.
\] (24)

The goal is now to derive an interaction matrix \( \mathbf{L}_{s_w} \in \mathbb{R}^{N \times 6} \) that relates the camera motion defined by the 6-dimensional velocity vector \( \mathbf{v} \) to the time-derivation of the wavelet coefficients vector, such that the following linearization holds
\[
\frac{ds_w(t)}{dt} = \mathbf{L}_{s_w}(t) \mathbf{v}(t).
\] (25)

Let us assume that all wavelet generators as well as the image intensities \( I \) are continuously differentiable. With the 2-dimensional generalization of the Leibniz integral rule, also known as the Reynolds transport theorem (see for example (Flanders, 1973)), we can express the time-derivative of a single entry of the feature vector \( s_w \) at a fixed point \((l, j, m)\) by
\[
\frac{d}{dt} \left( \mathcal{W}_{\psi(l)} I(l, j, m) \right) = \frac{d}{dt} \left( \mathbf{I} \mathbf{L}_p(x, y) \mathbf{v}(t) \mathbf{y}_{l,j,m}(x, y) \right) = \int_{\mathbb{R}^2} \frac{\partial \mathbf{I}(x, y, t)}{\partial t} \mathbf{y}_{l,j,m}(x, y) \, dx \, dy.
\] (26)

Then, by applying relation (19), we get
\[
\frac{d}{dt} \left( \mathbf{I}_l \mathbf{y}_{l,j,m} \right) = - \int_{\mathbb{R}^2} \left( \nabla \mathbf{I}_l(x, y) \right)^T \mathbf{L}_p(x, y) \mathbf{v}(t) \mathbf{y}_{l,j,m}(x, y) \, dx \, dy.
\] (27)

We can simplify our notation by setting
\[
\mathbf{I}_l = (\nabla \mathbf{I}_l(x, y))^T \mathbf{L}_p(x, y) \mathbf{e}_i,
\] (28)

where \( i \in \{1, \ldots, 6\} \) and \( \mathbf{e}_i \in \mathbb{R}^6 \) denotes the \( i \)-th canonical unit vector (note that \( \mathbf{L}_p(x, y) \mathbf{e}_i \) is the \( i \)-th column of the matrix \( \mathbf{L}_p(x, y) \)). In other words, \( \mathbf{I}_l \) encodes the correlation of the image gradient with the movement contributed by the \( i \)-th degree-of-freedom. Using (28), the interaction matrix with respect to the wavelet-based feature vector \( s_w \) can be written as:
\[
\mathbf{L}_{s_w}(t) = \left[ \begin{array}{ccc}
(\mathcal{W}_{\psi(1)} I_1^{(1)})(j_1, m_1) & \cdots & (\mathcal{W}_{\psi(1)} I_1^{(6)})(j_1, m_1) \\
\vdots & \ddots & \vdots \\
(\mathcal{W}_{\psi(N)} I_N^{(1)})(j_N, m_N) & \cdots & (\mathcal{W}_{\psi(N)} I_N^{(6)})(j_N, m_N)
\end{array} \right] \in \mathbb{R}^{N \times 6}.
\] (29)

Each column of \( \mathbf{L}_{s_w} \) represents the wavelet transform of an image \( I_l^{(i)} \) with respect to the wavelet system defined by the wavelet generators \( \{\psi^{(i)}\}_{i \leq L} \) and the sequence \( \{l_n, j_n, m_n\}_{n \leq N} \). Constructing the interaction matrix \( \mathbf{L}_{s_w} \) thus corresponds to computing the image gradient of \( I_l \) and performing six discrete wavelet transforms. While the FWT is indeed a linear time algorithm, this might still be too much in time-critical applications. One possibility to limit the computational effort is to reduce the number of considered scales \( j \). Another approach, which reduces the number of discrete wavelet transforms to two, is given by only considering an approximation of \( \mathbf{L}_{s_w} \).

Let us assume that all generator functions \( \psi^{(i)} \) are compactly supported in the spatial domain with their support being centered around the origin. It was already noted in (Bernard, 2001) that in such a setting, the
velocities \( \left( \frac{dx}{dt}, \frac{dy}{dt} \right) \) in the optical flow constraint equation (18) remain approximately constant on the support of the wavelet \( \psi^{(l)}_{j,m} \). This assumption is equivalent with replacing \( L_p(x,y) \) inside the integral in (27) with a matrix \( L_{pw}(j,m) \) that only depends on the center of the support set of the wavelet \( \psi^{(l)}_{j,m} \) and is thus fixed for any pair of parameters \((j,m) \in \mathbb{N} \times \mathbb{Z}^2\). Therefore, \( L_{pw}(j,m) \) can simply be computed by evaluating \( L_p \) at the center of the wavelet, that is
\[
L_{pw}(j,m) = L_p(2^{-j}m_1,2^{-j}m_2).
\]
By using (30), an approximation \( \tilde{L}_{s_u}(t) \approx L_{s_u}(t) \) of the wavelet-based interaction matrix at a time \( t \) that only requires the computation of two discrete wavelet transforms is thus given by
\[
\tilde{L}_{s_u}(t) = - \left[ \begin{array}{c}
\left( (\mathcal{W}_{\psi}^{(l)} \frac{\partial}{\partial x})(j_1,m_1), (\mathcal{W}_{\psi}^{(l)} \frac{\partial}{\partial y})(j_1,m_1) \right) L_{pw}(j_1,m_1) \\
\vdots \\
\left( (\mathcal{W}_{\psi}^{(l)} \frac{\partial}{\partial x})(j_N,m_N), (\mathcal{W}_{\psi}^{(l)} \frac{\partial}{\partial y})(j_N,m_N) \right) L_{pw}(j_N,m_N)
\end{array} \right] \in \mathbb{R}^{N \times 6}.
\] (31)
Let us denote the visual error with respect to the wavelet feature vector by
\[
e_w(t) = s_w(t) - s_w^*.
\] (32)
In order to minimize \( e_w \), the following control law is derived
\[
v(t) = -\lambda \left( H(t) + \mu \text{diag}(H(t)) \right)^{-1} \left( L_{s_u}(t) \right)^T e_w(t),
\] (33)
where \( \lambda > 0 \) is a controller gain parameter, \( \mu > 0 \) a damping factor and \( H(t) = \left( L_{s_u}(t) \right)^T L_{s_u}(t) \).

The interaction matrix \( \tilde{L}_{s_u} \) is based on the approximation of the partial derivatives of the 2-dimensional image \( I_t \) yielded by a gradient filter. However, by applying integration by parts, \( \tilde{L}_{s_u} \) could also be computed by considering the partial derivatives of the wavelet \( \psi^{(l)}_{j,m} \). While this approach is of limited practical value, it has some interesting theoretical aspects and is thus briefly described in Appendix 6.1.

In order to derive a shearlet-based visual servoing scheme, we can follow the same steps as in the wavelet-based approach.

### 3.3 Shearlet-based Visual Servoing

Shearlet-based image representations can be used in visual servoing tasks analogously to the wavelet-based approach described in the previous section. For a finite set of \( L \in \mathbb{N} \) shearlet generators \( \{\psi^{(l)}\}_{l \leq L} \subset L^2(\mathbb{R}^2) \), the shearlet-based feature vector for a set of \( N \in \mathbb{N} \) quadruples \( \{l_n, j_n, k_n, m_n\}_{n \leq N} \subset \{1, \ldots, L\} \times \mathbb{N} \times \mathbb{Z} \times \mathbb{Z}^2 \), where \( l_n \) specifies a shearlet generator, \( j_n \) defines the scale, \( k_n \) the shear operator and \( m_n \) the translation parameter, is given as a function of \( t \) by
\[
s_{sh}(t) = \left( (S_{\psi^{(l)}I_l}(j_1,k_1,m_1)), \ldots, (S_{\psi^{(l)}I_l}(j_N,k_N,m_N)) \right)^T,
\] (34)
where
\[
(S_{\psi^{(l)}I_l}(j,k,m)) = \iint_{\mathbb{R}^2} I_l(x,y) D^j \nabla^k T_m \psi^{(l)}(x,y) \, dx \, dy,
\] (35)
if the generator $\psi^{(l)}$ belongs to the horizontal frequency cone, while

$$
(S^{(l)}_{\psi}I_l)(j, k, m) = \int_{\mathbb{R}^2} I_l(x,y) D^l_A D_{S^l} T_m \psi^{(l)}(x,y) \, dx \, dy,
$$

(36)

if the shearlet generator belongs to the vertical frequency cone.

After repeating the computations carried out in (26) to (28), the interaction matrix associated with the shearlet coefficient vector $s_{sh}$ at a time $t$ can be written as

$$
L_{sh}(t) = \begin{bmatrix}
(S^{(1)}_{\psi}I_l^1)(j_1, k_1, m_1) & \cdots & (S^{(6)}_{\psi}I_l^6)(j_1, k_1, m_1) \\
\vdots & \ddots & \vdots \\
(S^{(1)}_{\psi}I_l^1)(j_N, k_N, m_N) & \cdots & (S^{(6)}_{\psi}I_l^6)(j_N, k_N, m_N)
\end{bmatrix} \in \mathbb{R}^{N \times 6}.
$$

(37)

The main formal difference between shearlet- and wavelet-based visual servoing control laws can be found in the definition of the approximated interaction matrix $L_{sh}$ (which is the core of a visual servoing scheme). This is due to the fact that the center of a shearlet is not only determined by the scale parameter $j$, but also depends on the shear parameter $k$. Under the assumption that the shearlet generators are compactly supported in the time-domain with their supports being centered around the origin, we can write

$$
L_{psh}(j, k, m) = L_p\left(2^{-j}(m_1 - km_2), 2^{-j/2}m_2\right),
$$

(38)

if the corresponding generator belongs to the horizontal frequency cone and

$$
L_{psh}(j, k, m) = L_p\left(2^{-j/2}m_1, 2^{-j}(m_2 - km_1)\right),
$$

(39)

if the generator belongs to the vertical frequency cone. The approximated shearlet-based interaction matrix $\bar{L}_{sh}(t) \approx L_{sh}(t)$ is now given by

$$
\bar{L}_{sh}(t) = \begin{bmatrix}
(S^{(1)}_{\psi}I_l^1)(j_1, k_1, m_1), (S^{(6)}_{\psi}I_l^6)(j_1, m_1) L_{psh}(j_1, k_1, m_1) \\
\vdots \\
(S^{(1)}_{\psi}I_l^1)(j_N, k_N, m_N), (S^{(6)}_{\psi}I_l^6)(j_N, m_N) L_{psh}(j_N, k_N, m_N)
\end{bmatrix} \in \mathbb{R}^{N \times 6}.
$$

(40)

The visual error with respect to the shearlet-based feature vector

$$
e_{sh}(t) = s_{sh}(t) - s^*_{sh},
$$

(41)

can be minimized towards zero by adjusting the velocity twist vector using the following control law

$$
v(t) = -\lambda \left( H(t) + \mu \text{diag}(H(t)) \right)^{-1} \left( L_{sh}(t) \right)^T e_{sh}(t),
$$

(42)

where $\lambda > 0$ is a gain parameter, $\mu > 0$ a damping factor and $H(t) = \left( L_{sh}(t) \right)^T L_{sh}(t)$. 

16
4 Simulation and Experimental Validations

4.1 Implementation Details

In order to compare both wavelet and shearlet-based visual servoing control laws, different scenarios were considered. Note that, for each method, there exist two implementation variants, i.e., subsampled and non-subsampled ones. In fact, in total, we developed four different control laws, i.e., two were based on the wavelet coefficients as signal inputs in the control loop while the other two use the shearlet ones. Also, each control law was validated with simulation and experimentally. Simulations allowed also to compared them (under the same conditions) to the photometry method (using the authors’s code provided in the Visual Servoing Platfrom (ViSP) library \(^{1}\)). The aim was to compare qualitatively and quantitatively the different laws in both favorable and unfavorable working conditions. To do that, the general framework was implemented in C++ under the ViSP platform. However, the computation of the the wavelet coefficients (i.e., subsampled wavelet: \(s\text{-wavelet}\), and non-subsampled wavelet: \(ns\text{-wavelet}\)) as well as the shearlet coefficients (i.e., the subsampled shearlet: \(s\text{-shearlet}\) and the non-subsampled shearlet: \(ns\text{-shearlet}\)) was performed under MATLAB. The MATLAB wavelet toolbox \(^{2}\) and the Shearlab 3D \(^{3}\) were called inside the ViSP framework using a developed function that can compile and retrieve the results of matlab functions directly into the C++ environment.

\[
\begin{array}{|c|c|c|c|c|}
\hline
\text{interaction matrix} & L_{s_{\text{w}}} & L_{s_{\text{sh}}} & L_{n_{\text{w}}} & L_{n_{\text{sh}}} \\
\hline
\text{t}_{s} & 208.18 \text{ ms} & 498.57 \text{ ms} & 392.93 \text{ ms} & 220.65 \text{ ms} \\
\text{t}_{\text{loop}} & 100 \text{ ms} & 140 \text{ ms} & 700 \text{ ms} & 800 \text{ ms} \\
\text{n}_{s} & 20 059 & 24 576 & 262 144 & 327 680 \\
\text{n}_{\text{level}} & 2 & 2 & 1 & 1 \\
\hline
\end{array}
\]

Table 2: Some details concerning the different developed methods. \(t_{s}\): required time to compute each interaction matrix, \(t_{\text{loop}}\): duration of one iteration of the control loop, \(n_{s}\): size of the visual feature vector, \(n_{\text{level}}\): how many levels of wavelet/shearlet coefficients used in the interaction matrix.

Table 2 details the required time of the main steps of each developed control law (e.g., wavelet/shearlet coefficients computation, interaction matrix, etc.). The respective interaction matrices were computed at each iteration (new grabbed image) and the \(Z\) depth was estimated manually at the desired position and maintained constant during the visual servoing process. The number of decomposition levels \(n_{\text{level}}\), which were used to obtain the corresponding feature vector (directly linked to the size \(n_{s}\) of the visual feature vector \(s\)) is different in the case of subsampled and non-subsampled transforms. In fact, a compromise has to be found, the higher the level, the larger the size of \(s\) and then the computation time increases. It is for this reason that only one level of the decomposition was considered for the non-subsampled methods which contains a large number of coefficients (10 times more than the subsampled methods). Furthermore, the visual feature vectors representations are only subsets of the corresponding transforms, where the high-frequency coefficients are omitted.

As the main objective of this paper is to provide the methodology to follow in order to obtain the wavelet and shearlet visual servoing control laws and to demonstrate its feasibility from experimental validations,

\(^{1}\)https://visp.inria.fr/
\(^{3}\)http://www.shearlab.org/software
we do not address here the issue of computation time optimization. However, this issue will be tackled in future investigations which are more engineering developments than basic research. Furthermore, for a better convergence behaviour of the different control laws, we have implemented adaptive gains for $\lambda$ and $\mu$ (the parameters of the Levenberg-Marquardt optimization algorithm). They are given by

$$
\lambda = 10^{\log_{10}(N_e)-7.5}
$$

$$
\mu = 10^{2\log_{10}(N_e)-17}
$$

where the gain $\mu$ decreases in function of the reduction of the square sum $N_e$ of the difference between the current image $I$ and the desired image $I^\ast$. This allows a rapid convergence at the beginning of the positioning task when the initial error is large and vice-versa when the robot approaches the desired position, the convergence becomes slow and smooth. The value 7.5 and 17 has been chosen empirically.

4.2 Wavelet and Shearlet Coefficients

![Image of a pepper and its wavelet transform](image)

Figure 6: The feature vector $s_w$ was obtained by performing a discrete wavelet transform at level 3. The upper left corner in Fig. 6b contains a coarse approximation of the original image of size $32 \times 32$. The remaining pixels correspond to detail coefficients at the second and third stage of decomposition. Note that the high-frequency detail coefficients from the first stage of decomposition were omitted.

We consider the image shown in Fig. 6a to illustrate what the visual feature vectors obtained from different image representation methods such as the sub-and non-subsampled wavelet and shearlet transforms look like.

Indeed, the obtained visual feature vector $s_w$ corresponding to the case of $s$-wavelet visual servoing control law is shown in Fig. 6b. This vector includes the vertical, horizontal, and diagonal detail wavelet coefficients as well as the wavelet coarse ones. Also, the visual feature vector $s_w$ corresponding to the $ns$-wavelet image decomposition is depicted in Fig. 7. Similarly to the wavelet case, the visual feature vectors $s_{sh}$ resulted from the shearlet image decomposition are depicted in Fig. 8 and Fig. 9. More precisely, the $s_{sh}$ corresponding to the $s$-shearlet version of the subsampled shearlet decomposition is depicted in Fig. 8 when the non-subsampled shearlet $ns$-shearlet coefficients are represented in Fig. 9.
Figure 7: $s_w$ in case of ns-wavelet. Concatenating the four displayed coefficient matrices yields the feature vector $s_w$. The matrices were obtained by performing a non-subsampled wavelet transform at level 4. The first coefficient matrix defines a coarse approximation of the original image. The three remaining matrices show detail coefficients at the fourth level of decomposition. Similar to the $s$-wavelet case, the high-frequency coefficients yielded by the first, second and third stage of decomposition were omitted. Note that all matrices are of the same size as the original image.

Figure 8: $s_{sh}$ in case of s-shearlet. Concatenating the six displayed coefficient matrices yields the feature vector $s_{sh}$. The matrices were obtained by performing a discrete subsampled shearlet transform at level 4. Each upper left corner contains a coarse approximation of the original image of size $16 \times 16$ along one of the six considered directions. The remaining pixels in each matrix correspond to detail coefficients at the third and fourth stage of decomposition. Note that the high-frequency detail coefficients from the first and second stage of decomposition were omitted.

### 4.3 Cost-functions

In order to judge the effectiveness of the developed wavelet and shearlet visual servoing approaches, especially in term of convergence domain allowing reaching the desired visual features $s^*$, we have computed the cost-function corresponding to each of the four methods by varying the different robot DOF using the
The matrices were obtained by performing a non-subsampled shearlet transform at level 4. The first coefficient matrix defines a coarse approximation of the original image. Each of the other matrices was obtained by convolving the input image with a discrete shearlet filter associated with a different orientation. Note that the high-frequency shearlet coefficients from the first, second and third stage of decomposition were omitted and that each matrix has the same size as the original image.

The following relationship:

\[
C(s) = \left( \frac{(s - s^*)^\top (s - s^*)}{N_{\text{pix}}} \right)^{\frac{1}{2}}
\]

where \(N_{\text{pix}}\) is the number of pixels of the whole image.

Figure 10: Cost-function of (a) the subsampled wavelet visual features and (b) the subsampled shearlet visual features, along the \(x\) and \(y\) axes.
As can be seen in Fig. 10 and 11, the global minimum is clearly identified for each method (subsampled and non-subsampled wavelet/shearlet). This can be considered as a positive point concerning the capabilities of the developed control laws to converge towards the desired position using a state-of-the-art optimization method (e.g., the Levenberg-Marquardt algorithm).

### 4.4 Simulation Validation

In order to demonstrate that the proposed visual servoing control laws work, first we opted for a simulation framework. The aim is to test qualitatively the proposed methods in different scenarios without any effect due to the camera or camera-robot calibrations, measurement errors, etc. To perform this study, each developed control law was validated in favorable (nominal) conditions (Fig. 12(a)) as well as unfavorable ones (partial occlusions (Fig. 12(b)) and light disturbance (Fig. 12(c))). The initial pose error $e_0 (e_0 \in SE(3))$ for each test ($Test_i : i \in [1,2,3]$) is reported in Table 3. Also, the images representing the initial intensity difference between the initial and desired images are depicted in Fig. 13.

In the next, the image of difference, noticed $I_{diff}$, between the current and desired images will also be used to illustrate ”visually” the achievement of each positioning task in both the simulation and the experimental
validations. Indeed, when the robot reaches, ”perfectly”, the desired position, $I_{diff}$ must be completely gray (i.e. the desired and final images are perfectly superposed). $I_{diff}$ is defined as follows

$$I_{diff} = \frac{(I - I^*) + 255}{2}$$

We add 255 in the numerator in order to obtain a value of $I_{diff} = 128$ i.e., a perfect gray image (more suitable to show an image difference) instead of a black image.

<table>
<thead>
<tr>
<th>test</th>
<th>$\Delta T_x$ (mm)</th>
<th>$\Delta T_y$ (mm)</th>
<th>$\Delta T_z$ (mm)</th>
<th>$\Delta R_x$(deg)</th>
<th>$\Delta R_y$(deg)</th>
<th>$\Delta R_z$(deg)</th>
<th>figure</th>
</tr>
</thead>
<tbody>
<tr>
<td>test 1</td>
<td>10</td>
<td>-10</td>
<td>100</td>
<td>5</td>
<td>-5</td>
<td>10</td>
<td>Fig. 13(a)</td>
</tr>
<tr>
<td>test 2</td>
<td>5</td>
<td>-5</td>
<td>100</td>
<td>-20</td>
<td>10</td>
<td>-5</td>
<td>Fig. 13(b)</td>
</tr>
<tr>
<td>test 3</td>
<td>-2</td>
<td>2</td>
<td>-50</td>
<td>5</td>
<td>10</td>
<td>-30</td>
<td>Fig. 13(c)</td>
</tr>
</tbody>
</table>

Table 3: Initial pose error $e_0$ between the initial camera pose and the desired one.

4.4.1 Under Nominal Conditions [scenario 1]

Let us consider the scenario 1. As mentioned above, this test aims to study the behavior of the four developed control laws in nominal working conditions. In practice, for each of the 3 tests ($test_i$) a virtual camera observing a planar scene is first placed on a desired pose where the desired image is recorded. Its pose is then changed with respect to the desired one such to obtain the initial pose error $e = e_0 (e \in SE(3))$ given in Table 3. Then, the visual servoing is launched and the final pose error $e_f = e$ is measured at the end of the convergence.

Table 4 reports the results obtained for the four proposed methods as well as the photometric approach. At the convergence, we obtain a final translation error $\Delta T_i$ ($i = [x, y, z]$) around $10^{-4}$mm and a rotation error $\Delta R_i$ ($i = [x, y, z]$) around $10^{-4}$deg for each method. As can be highlighted, all controllers converge and present the same performances in term of accuracy. Note that the pixel size of the simulated images is $\eta = 1.2$mm.

4.4.2 Under Partial Occlusions [scenario 2]

The scenario 2 deals with the validation of the four developed control laws and their comparison to the photometric one in unfavourable working conditions, i.e., partial occlusions. This consists of the study of
the behavior of the control laws when a part of the desired image (Fig. 12(b)) is hidden during the positioning task. The same initial pose as scenario 1 is considered in this test. As can be seen in Table 5, for the test 1, the four controllers as well as the photometric method converge towards the desired pose despite a static error especially for the out-plane rotations $R_x$ and $R_y$ and the translation stage $T_z$. However, the s-wavelet approach behaves better than the three others, especially along the Z axis. Generally, the methods based on the wavelet/shearlet image decomposition have a slight advantage over the photometry one. Finally, in test 3, all the control laws fail (X) because of the large initial error $e_0$ and the partial occlusions.

### 4.4.3 Under Illumination Changes [scenario 3]

The last simulation scenario aims to evaluate the ability of the tested control laws to work under lighting disturbances. The desired image $I^*$ (Fig. 12(b)) is acquired under a saturated lighting source when the current images are grabbed in normal conditions. As shown in Table 6, all controllers converge for test 1. However, because of the lighting disturbances and the larger error in the out-plane rotations $R_x$ and $R_y$, the wavelet/shearlet methods converge contrary to the photometric approach. The observed static errors in the $Z$-translation, $R_x$ and $R_y$ rotations can be explained by the fact that even if the proposed approaches are robust to external disturbances, the control laws were derived from global image information. Indeed, if the common information between the desired and current images is weak (e.g., due to the lighting disturbances and/or occlusions), there is a risk of having static errors. Another remark is that where the photometry approach diverges completely in test 3 and test 3, the subsampled s-wavelet and s-shearlet control laws remain convergent (see test 3 in Table 6).

<table>
<thead>
<tr>
<th>Method / Test</th>
<th>$\Delta T_x$</th>
<th>$\Delta T_y$</th>
<th>$\Delta T_z$</th>
<th>$\Delta R_x$</th>
<th>$\Delta R_y$</th>
<th>$\Delta R_z$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$e_0$ test 1</td>
<td>10 $10^{-4}$</td>
<td>10 $10^{-4}$</td>
<td>100</td>
<td>5 $10^{-4}$</td>
<td>-5 $10^{-4}$</td>
<td>10</td>
</tr>
<tr>
<td>$e_f$ photometry</td>
<td>10 $10^{-4}$</td>
<td>10 $10^{-4}$</td>
<td>10 $10^{-4}$</td>
<td>10 $10^{-4}$</td>
<td>10 $10^{-4}$</td>
<td>10 $10^{-4}$</td>
</tr>
<tr>
<td>$e_f$ s-wavelet</td>
<td>10 $10^{-4}$</td>
<td>10 $10^{-4}$</td>
<td>10 $10^{-4}$</td>
<td>10 $10^{-4}$</td>
<td>10 $10^{-4}$</td>
<td>10 $10^{-4}$</td>
</tr>
<tr>
<td>$e_f$ s-shearlet</td>
<td>10 $10^{-4}$</td>
<td>10 $10^{-4}$</td>
<td>10 $10^{-4}$</td>
<td>10 $10^{-4}$</td>
<td>10 $10^{-4}$</td>
<td>10 $10^{-4}$</td>
</tr>
<tr>
<td>$e_f$ ns-wavelet</td>
<td>10 $10^{-4}$</td>
<td>10 $10^{-4}$</td>
<td>10 $10^{-3}$</td>
<td>10 $10^{-3}$</td>
<td>10 $10^{-3}$</td>
<td>10 $10^{-3}$</td>
</tr>
<tr>
<td>$e_f$ ns-shearlet</td>
<td>10 $10^{-4}$</td>
<td>10 $10^{-4}$</td>
<td>10 $10^{-3}$</td>
<td>10 $10^{-3}$</td>
<td>10 $10^{-3}$</td>
<td>10 $10^{-3}$</td>
</tr>
<tr>
<td>$e_0$ test 2</td>
<td>5 $10^{-5}$</td>
<td>100</td>
<td>-20 $10^{-10}$</td>
<td>10 $10^{-5}$</td>
<td>-5 $10^{-5}$</td>
<td>10</td>
</tr>
<tr>
<td>$e_f$ photometry</td>
<td>10 $10^{-4}$</td>
<td>10 $10^{-4}$</td>
<td>10 $10^{-4}$</td>
<td>10 $10^{-4}$</td>
<td>10 $10^{-4}$</td>
<td>10 $10^{-4}$</td>
</tr>
<tr>
<td>$e_f$ s-wavelet</td>
<td>10 $10^{-4}$</td>
<td>10 $10^{-4}$</td>
<td>10 $10^{-4}$</td>
<td>10 $10^{-4}$</td>
<td>10 $10^{-4}$</td>
<td>10 $10^{-4}$</td>
</tr>
<tr>
<td>$e_f$ s-shearlet</td>
<td>10 $10^{-4}$</td>
<td>10 $10^{-4}$</td>
<td>10 $10^{-4}$</td>
<td>10 $10^{-4}$</td>
<td>10 $10^{-4}$</td>
<td>10 $10^{-4}$</td>
</tr>
<tr>
<td>$e_f$ ns-wavelet</td>
<td>10 $10^{-4}$</td>
<td>10 $10^{-4}$</td>
<td>10 $10^{-4}$</td>
<td>10 $10^{-4}$</td>
<td>10 $10^{-4}$</td>
<td>10 $10^{-4}$</td>
</tr>
<tr>
<td>$e_f$ ns-shearlet</td>
<td>10 $10^{-4}$</td>
<td>10 $10^{-4}$</td>
<td>10 $10^{-4}$</td>
<td>10 $10^{-4}$</td>
<td>10 $10^{-4}$</td>
<td>10 $10^{-4}$</td>
</tr>
<tr>
<td>$e_0$ test 3</td>
<td>-2 $10^{-2}$</td>
<td>-50 $10^{-5}$</td>
<td>10 $10^{-5}$</td>
<td>10 $10^{-5}$</td>
<td>-30 $10^{-3}$</td>
<td>10 $10^{-3}$</td>
</tr>
<tr>
<td>$e_f$ photometry</td>
<td>10 $10^{-4}$</td>
<td>10 $10^{-4}$</td>
<td>10 $10^{-4}$</td>
<td>10 $10^{-4}$</td>
<td>10 $10^{-4}$</td>
<td>10 $10^{-4}$</td>
</tr>
<tr>
<td>$e_f$ s-wavelet</td>
<td>10 $10^{-4}$</td>
<td>10 $10^{-4}$</td>
<td>10 $10^{-4}$</td>
<td>10 $10^{-4}$</td>
<td>10 $10^{-4}$</td>
<td>10 $10^{-4}$</td>
</tr>
<tr>
<td>$e_f$ s-shearlet</td>
<td>10 $10^{-4}$</td>
<td>10 $10^{-4}$</td>
<td>10 $10^{-4}$</td>
<td>10 $10^{-4}$</td>
<td>10 $10^{-4}$</td>
<td>10 $10^{-4}$</td>
</tr>
<tr>
<td>$e_f$ ns-wavelet</td>
<td>10 $10^{-4}$</td>
<td>10 $10^{-4}$</td>
<td>10 $10^{-4}$</td>
<td>10 $10^{-4}$</td>
<td>10 $10^{-4}$</td>
<td>10 $10^{-4}$</td>
</tr>
<tr>
<td>$e_f$ ns-shearlet</td>
<td>10 $10^{-4}$</td>
<td>10 $10^{-4}$</td>
<td>10 $10^{-4}$</td>
<td>10 $10^{-4}$</td>
<td>10 $10^{-4}$</td>
<td>10 $10^{-4}$</td>
</tr>
</tbody>
</table>

Table 4: [scenario1] comparison of the different control laws in nominal conditions. $\Delta T_i$ (mm), $\Delta R_i$ (deg) $e_0$ and $e_f$ represent the initial and final positioning errors, respectively.
Table 5: [scenario 2] comparison of the different control laws under partial occlusion. $\Delta T_i$ (mm), $\Delta R_i$ (deg) $e_0$ and $e_f$ represent the initial and final positioning errors, respectively. X = fail.

To sum up, the objective of these simulations is the study of the ability of the four developed controllers to work in different working conditions as well as their comparison with the photometric method. All controllers work effectively in normal conditions, i.e., without external disturbances and demonstrated high accuracy. When we introduced external disturbances (by varying the illumination conditions or occluding a part of the scene), the controllers converged despite a large static error on the out-plane rotation and translation along Z axis. Also, the photometric method fails twice i.e., for the test 2 and test 3 when the developed approaches remain working despite the external disturbances. Furthermore, as it has been noticed, the subsampled version of the control laws (s-wavelet and s-shearlet) stood out in some unfavorable tests compared to the non-sampled ones ns-wavelet and ns-shearlet) and also comparing to the photometry method. This is due to the fact that subsampled methods integrate a low-pass filter during the image decomposition allowing to remove intuitively the image high-frequencies (corresponding to the image noise) from the visual features set s. In addition, the fact that the subsampled methods use less visual features ($\text{dim}(s_s) \ll \text{dim}(s_{ns})$) to compute the related interaction matrices allows reducing computation time, i.e., $t_{loop} = 120$ ms for the subsampled methods, while $t_{loop} = 700$ ms for the non-subsampled methods. Only the subsampled methods will be considered in the following.

### 4.4.4 Robustness to Intrinsic Camera Parameters Errors [scenario 4]

Simulation allows also to easily study the robustness of each method to intrinsic camera parameter errors. Thus, the two subsampled methods have been compared. Two parameters were changed: the principal point coordinates and the focal length. The real principal point coordinates are $(u_0, v_0) = (160, 120)$ pixels while
Table 6: 

<table>
<thead>
<tr>
<th>$e_f$ and $e_0$</th>
<th>$\Delta T_x$</th>
<th>$\Delta T_y$</th>
<th>$\Delta T_z$</th>
<th>$\Delta R_x$</th>
<th>$\Delta R_y$</th>
<th>$\Delta R_z$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$e_0$</td>
<td>test 1</td>
<td>10</td>
<td>-10</td>
<td>100</td>
<td>5</td>
<td>-5</td>
</tr>
<tr>
<td>$e_f$</td>
<td>photometry</td>
<td>0.84</td>
<td>0.57</td>
<td>-31.49</td>
<td>4.27</td>
<td>-5.61</td>
</tr>
<tr>
<td>$e_f$</td>
<td>s-wavelet</td>
<td>0.10</td>
<td>0.69</td>
<td>-23.48</td>
<td>4.43</td>
<td>-5.65</td>
</tr>
<tr>
<td>$e_f$</td>
<td>s-shearlet</td>
<td>0.52</td>
<td>0.49</td>
<td>-13.57</td>
<td>4.26</td>
<td>-5.66</td>
</tr>
<tr>
<td>$e_f$</td>
<td>ns-wavelet</td>
<td>0.83</td>
<td>0.60</td>
<td>-23.66</td>
<td>4.26</td>
<td>-5.65</td>
</tr>
<tr>
<td>$e_f$</td>
<td>ns-shearlet</td>
<td>0.78</td>
<td>0.59</td>
<td>-22.76</td>
<td>4.27</td>
<td>-5.62</td>
</tr>
<tr>
<td>$e_0$</td>
<td>test 2</td>
<td>5</td>
<td>-5</td>
<td>100</td>
<td>-20</td>
<td>10</td>
</tr>
<tr>
<td>$e_f$</td>
<td>photometry</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>$e_f$</td>
<td>s-wavelet</td>
<td>0.34</td>
<td>0.66</td>
<td>-59.68</td>
<td>-20.27</td>
<td>9.25</td>
</tr>
<tr>
<td>$e_f$</td>
<td>s-shearlet</td>
<td>0.80</td>
<td>0.89</td>
<td>-52.28</td>
<td>-20.31</td>
<td>9.28</td>
</tr>
<tr>
<td>$e_f$</td>
<td>ns-wavelet</td>
<td>1.34</td>
<td>0.456</td>
<td>-60.99</td>
<td>-20.32</td>
<td>9.33</td>
</tr>
<tr>
<td>$e_f$</td>
<td>ns-shearlet</td>
<td>1.20</td>
<td>0.52</td>
<td>-59.19</td>
<td>-20.28</td>
<td>9.35</td>
</tr>
</tbody>
</table>

Table 6: [scenario 3] comparison of the different control laws under illumination changes. $\Delta T_i$ (mm), $\Delta R_i$ (deg) $e_0$ and $e_f$ represent the initial and final positioning errors, respectively. X = fail.

The real focal length is $p_x = p_y = 870$ pixels. Table 7 shows the robustness of both methods to focal length error. When the error goes in favor of an increase of the focal length, both methods converge but they take more iterations. When the error goes in favor of a decrease of the focal length, the subsampled wavelet method falls into a local minima while the subsampled shearlet method converges in few iterations.

Table 7: Robustness study of the controller against focal length errors for subsampled methods in test2

<table>
<thead>
<tr>
<th>Focal length</th>
<th>Percent of error</th>
<th>s-wavelet</th>
<th>s-shearlet</th>
</tr>
</thead>
<tbody>
<tr>
<td>435px</td>
<td>(-50%)</td>
<td>X</td>
<td>(190 iterations)</td>
</tr>
<tr>
<td>609px</td>
<td>(-30%)</td>
<td>X</td>
<td>(440 iterations)</td>
</tr>
<tr>
<td>696px</td>
<td>(-20%)</td>
<td>✓</td>
<td>(700 iterations)</td>
</tr>
<tr>
<td>870px</td>
<td>(0%)</td>
<td>✓</td>
<td>(950 iterations)</td>
</tr>
<tr>
<td>1305px</td>
<td>(+50%)</td>
<td>✓</td>
<td>(2970 iterations)</td>
</tr>
</tbody>
</table>

Since both methods converge when the focal length given in the algorithm is 20% lower than the real one (i.e. $p_x = p_y = 696$ pixels), this error is kept for the next test where different false principal point coordinates are given to the algorithm. The results presented in Table 8 show that there are no striking differences between both methods which can be therefore considered as robust to camera parameters errors.

4.5 Experimental Validation using a 6 DOF Cartesian Robot

In order to experimentally validate the proposed visual servoing control laws, we used a 6 DOF gantry robotic system (Fig. 14). A Charge Coupled Device (CCD) camera was mounted in an eye-in-hand config-
Table 8: Robustness study of the controller against principal point coordinates errors for subsampled methods in $test_1$

<table>
<thead>
<tr>
<th>Principal point coordinates</th>
<th>Euclidian distance to the real point</th>
<th>s-wavelet</th>
<th>s-shearlet</th>
</tr>
</thead>
<tbody>
<tr>
<td>(160,120)</td>
<td>0</td>
<td>✅ (160 iterations)</td>
<td>✅ (70 iterations)</td>
</tr>
<tr>
<td>(320,240)</td>
<td>200</td>
<td>✅ (360 iterations)</td>
<td>✅ (180 iterations)</td>
</tr>
<tr>
<td>(160,-100)</td>
<td>220</td>
<td>✅ (540 iterations)</td>
<td>✅ (170 iterations)</td>
</tr>
<tr>
<td>(-200,120)</td>
<td>360</td>
<td>✅ (310 iterations)</td>
<td>✅ (330 iterations)</td>
</tr>
<tr>
<td>(500,-200)</td>
<td>467</td>
<td>✗</td>
<td>✗</td>
</tr>
</tbody>
</table>

Figure 14: Photography of the experimental setup for planar scene.

The camera provides $450 \times 450$ pixels images at 25 frames per second. The grabbed images were resized into $256 \times 256$ pixels images to reduce the computation time of the associated interaction matrices. All the control process and the communication with the robot are performed on a 2.4-GHz computer working under a Linux distribution. All the wavelet and shearlet image decompositions are performed using the MATLAB wavelet toolbox and the open-source SHEARLAB 1.1, respectively. Finally, the related interaction matrices were computed with an estimated constant depth $Z^* = 0.8m$ (approximative distance between the camera and the observed scene).

Furthermore, as mentioned above, in the experimental validation, only two methods were considered, i.e., s-wavelet and s-shearlet-based control laws which outperform the non-subsampled methods in simulation.
Similarly to the simulation validation framework, we implemented different favourable and unfavourable scenarios to assess the performances of the proposed control laws. Note that, in each scenario, we used two tests (with two initial positions) as reported in Table 9. In addition to the scenarios studied in the simulation framework, we added validation tests in 2D and 3D scenes.

<table>
<thead>
<tr>
<th>Test</th>
<th>$\Delta T_x$ (mm)</th>
<th>$\Delta T_y$ (mm)</th>
<th>$\Delta T_z$ (mm)</th>
<th>$\Delta R_x(\cdot)$</th>
<th>$\Delta R_y(\cdot)$</th>
<th>$\Delta R_z(\cdot)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>test 1</td>
<td>5</td>
<td>50</td>
<td>100</td>
<td>5</td>
<td>-5</td>
<td>-4</td>
</tr>
<tr>
<td>test 2</td>
<td>-20</td>
<td>-30</td>
<td>-20</td>
<td>10</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 9: Initial error $e_0$ between the initial image $I$ and the desired one $I'$ for planar scene.

### 4.5.1 Nominal Conditions and Planar Scene [scenario 1]

The first experimental validation scenario consists of the test of both $s$-wavelet and $s$-shearlet controllers in nominal working conditions as well as in a planar scene (depth $Z$ is approximatively constant over the entire observed scene). Fig. 15 depicts the experimental test 1: Fig. 15(a) the desired image $I'$, Fig. 15(b) the initial difference image $I_{diff}$ between the desired and initial images, while Fig. 15(c) and (d) illustrate the final difference image at convergence for both $s$-wavelet and $s$-shearlet methods, respectively. As can be highlighted both methods converge accurately (difference images are completely gray) towards the desired position.

The final positioning error $e$ ($e \in SE(3)$), in each DOF, was recorded and plotted in Fig. 16. It was computed from the measures provided by the high resolution robot encoders. The first row of Fig. 16 depicts the error evolution in the case of $s$-wavelet method, while the second row presents the one of the $s$-shearlet method. As can be seen, all robot DOF converge to their respective desired position. However, the error decay is not exponential as in a conventional image-based visual servoing (IBVS) approach. This is due to the optimization method used in this work, namely, the Levenberg-Marquardt algorithm. The numerical values of the final positioning error in both methods are

- $s$-wavelet
  - translation: $mean(\Delta T_i) = 0.30 \text{ mm}$
  - rotation: $mean(\Delta R_i) = 0.029 \text{ deg}$
• \textit{s-shearlet}
  
  – translation: \textit{mean}(\Delta T_i) = 0.08 \text{ mm}
  – rotation: \textit{mean}(\Delta R_i) = 0.014 \text{ deg}

The \textit{s-wavelet}-based control law reaches the desired position in 700 iterations when the \textit{s-shearlet}-based approach converges in less iterations, i.e., 400. In addition, the \textit{s-shearlet} method demonstrates a better accuracy in both translation and rotation stages.

The achievement of \textit{test 2} is shown in Fig. 17. As can be underlined, both controller reach the desired position with accuracy demonstrated by the completely gray difference image \( I_{\text{diff}} \) as seen in Fig. 17(c) and Fig. 17(d).

Again, the error \( e \) decay in each DOF for the second \textit{test 2} is plotted in Fig. 18. Based on the evolution of the error, both the \textit{s-wavelet} and \textit{s-shearlet} have the same behavior as in the first \textit{test 1}. Finally, the numerical values of the final positioning error in both methods are

\begin{itemize}
  \item \textit{s-wavelet}
    \begin{itemize}
      \item translation: \textit{mean}(\Delta T_i) = 0.15 \text{ mm}
      \item rotation: \textit{mean}(\Delta R_i) = 0.16 \text{ deg}
    \end{itemize}
  \item \textit{s-shearlet}
    \begin{itemize}
      \item translation: \textit{mean}(\Delta T_i) = 0.13 \text{ mm}
      \item rotation: \textit{mean}(\Delta R_i) = 0.02 \text{ deg}
    \end{itemize}
\end{itemize}
4.5.2 Partial Occlusions and Planar Scene [scenario 2]

In this scenario, we tested the proposed control laws under partial occlusions, always using a planar object as the viewed scene when one used to hide a part of the image is three-dimensional. Thus, Fig. 19(a) shows the desired image $I^*$ in which we have added an external object (removed during the positioning task performing). The initial image difference $I_{diff}$ is depicted in Fig. 19(b). Also, Fig. 19(c) and Fig. 19(d) illustrate the same difference image when the s-wavelet and s-shearlet methods reach the desired position, respectively.

The error decay for both methods are depicted in Fig. 20. In the same figure, the first row represents the error evolution in case of the s-wavelet approach while the second row presents the s-shearlet one. As can
Figure 19: [scenario 2, test 1] (a) desired image $I^*$ (b) initial difference image $I_{diff}$ (c) final difference image for s-wavelet method, and (d) that of s-shearlet method.

Figure 20: [scenario 2, test 1] first row depicts the positioning error error using the s-wavelet-based control law and the second row the one of the s-shearlet-based approach.

It be highlighted both the control laws converge. However, the final error, as expected, increases with respect to the results of scenario 1. The numerical values on the accuracy of the proposed methods are

- **s-wavelet**
  - translation: $\text{mean}(\Delta T_i) = 2.91 \text{ mm}$
  - rotation: $\text{mean}(\Delta R_i) = 0.19 \text{ deg}$

- **s-shearlet**
  - translation: $\text{mean}(\Delta T_i) = 7.91 \text{ mm}$
Another remark can be raised, the s-wavelet method presents a better accuracy in both translation and rotation stages with respect to the s-shearlet approach. Also, the external 3D object added during the positioning task affects more the translation accuracy than the rotation one.

The validation scenario is repeated a second time (test 2) with a different initial error and external added object. Thus, Fig. 21(a) depicts the desired image with partial occlusions and Fig. 21(c) and Fig. 21(d) show the final difference images at convergence for the s-wavelet and s-shearlet, respectively. It can be noticed that the desired image and final image are accurately superimposed using the s-wavelet control law when the s-shearlet method fails.

![Figure 21](image)

Figure 21: [scenario 2, test 2] (a) desired image $I^*$ (b) initial difference image $I_{diff}$ (c) final difference image for s-wavelet method, and (d) that of s-shearlet method.

Indeed, the s-shearlet method converges towards a local minimum by the added external object. Fig. 22 shows the error decay in robot’s stages. Finally, the numerical values of the final positioning error in both methods are

- **s-wavelet**
  - translation: $\text{mean}(\Delta T_i) = 3.57$ mm
  - rotation: $\text{mean}(\Delta R_i) = 0.28$ deg

- **s-shearlet**
  - translation: $\times$
  - rotation: $\times$

### 4.5.3 Illumination Changes and Planar Scene [scenario 3]

This scenario associates a planar scene with illumination changes which are introduced during the positioning task. Fig. 23(a) illustrates the desired position grabbed under a saturated illumination source. The initial difference image is depicted in Fig. 23(b) as well as the final difference images which are shown in Fig. 23(c) and Fig. 23(d) for both s-wavelet and s-shearlet approaches, respectively. As can be seen, both methods reach the desired position despite the lighting disturbances. Note that the final difference images are not completely gray due to the illumination variation effect.

The behaviors of the control laws almost remain the same despite the illumination changes effect. For this test, the numerical values of the final positioning error in both methods are given below
Figure 22: [scenario 2, test 2] first row depicts the error $\mathbf{e}$ using the $s$-wavelet-based control law and the second row the one of the $s$-shearlet-based approach.

Figure 23: [scenario 3, test 2] (a) desired image $\mathbf{I}^\ast$ (b) initial difference image $\mathbf{I}_{\text{diff}}$ (c) final difference image for $s$-wavelet method, and (d) that of $s$-shearlet method.

- **s-wavelet**
  - translation: $\text{mean}(\Delta T_i) = 23.5$ mm
  - rotation: $\text{mean}(\Delta R_i) = 1.3$ deg

- **s-shearlet**
  - translation: $\text{mean}(\Delta T_i) = 21.3$ mm
  - rotation: $\text{mean}(\Delta R_i) = 1.3$ deg
4.5.4 Nominal Conditions and 3D Scene [scenario 4]

The scenarios described above were carried out using a planar scene. In this last scenario, we replaced the photography scene by a 3D scene and we increase the incidence angle between the camera and the 3D object in the desired position (Fig. 24). Despite the fact that our controller is designed for the general case, our monocular camera imposes us to consider an approximation of a constant depth $Z$ for all the points of the image in the equation (20). Therefore the goal of this scenario is to experimentally evaluate the robustness of our controller to modeling error on the depth of the scene. Two initial error between the initial image $I$ and the desired one $I^*$ were considered (Table 10).

![Figure 24: Photography of the experimental setup with 3D scene](image)

<table>
<thead>
<tr>
<th>Test</th>
<th>$\Delta T_x$ (mm)</th>
<th>$\Delta T_y$ (mm)</th>
<th>$\Delta T_z$ (mm)</th>
<th>$\Delta R_x$ ($^\circ$)</th>
<th>$\Delta R_y$ ($^\circ$)</th>
<th>$\Delta R_z$ ($^\circ$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>test 3</td>
<td>40</td>
<td>30</td>
<td>50</td>
<td>-5</td>
<td>2</td>
<td>-3</td>
</tr>
<tr>
<td>test 4</td>
<td>-100</td>
<td>-40</td>
<td>-200</td>
<td>3</td>
<td>-3</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 10: Initial error $e_0$ between the initial image $I$ and the desired one $I^*$ for 3D scene.

![Figure 25: [scenario 4, test3] (a) desired image $I^*$ (b) initial difference image $I_{\text{diff}}$ (c) final difference image for s-wavelet method, and (d) that of s-shearlet method.](image)

The numerical values of the final error measured for test 3 are

- **s-wavelet**

33
Figure 26: [scenario 4, test3] first row depicts the error $e$ using the $s$-wavelet-based control law and the second row the one of the $s$-shearlet-based approach.

- translation: $\text{mean} (\Delta T_i) = 0.43 \text{ mm}$
- rotation: $\text{mean} (\Delta R_i) = 0.03 \text{ deg}$

- $s$-shearlet
  - translation: $\text{mean} (\Delta T_i) = 0.15 \text{ mm}$
  - rotation: $\text{mean} (\Delta R_i) = 0.02 \text{ deg}$

As can be noticed, both control laws present a high accuracy (less than 1 mm in translation and less than 0.05 deg in rotation) for test3. In this test, both control laws work with 3D scene similarly as with 2D ones (the final errors for both cases are in the same order in this case).

In test4, the initial error in translation is enhanced compared to test3, the $s$-wavelet-based method converges with high accuracy while the $s$-shearlet-based method fails (Fig.27). The numerical values of the final error measured for test 4 are

- $s$-wavelet
  - translation: $\text{mean} (\Delta T_i) = 0.53 \text{ mm}$
  - rotation: $\text{mean} (\Delta R_i) = 0.11 \text{ deg}$

- $s$-shearlet
  - translation: $\text{mean} (\Delta T_i) = \chi$
– rotation: \( \text{mean}(\Delta R_i) = X \)

This last test shows that the \textit{s-wavelet} method outperforms the \textit{s-shearlet} one experimentally. In these two 3D tests, rotations can seem unassuming but it is important to keep in mind that a more than 10 degrees rotation in any direction makes the 3D object out of the field of view, except if a translation is also done. Unfortunately, the combination of both a high rotation and a high translation makes our controller fails. Figure 26 shows also that motions are jerky with a 3D scene. Nevertheless, the convergence can be reached for a 3D scene for a small initial pose error since in this case the scene behaves like a 2D scene. In opposite, if the initial pose error is important then considering an approximation of the 3D scene depth as a constant in the interaction matrix is not suitable and results in a convergence failure.

5 Conclusion

The main objective of this paper was the design of wavelet- and shearlet-based visual servoing schemes. We recalled the basic mathematical formalism concerning wavelet- and shearlet-based image decompositions and eventually derived four interaction matrices by considering the subsampled and non-subsampled wavelet- and shearlet-based multiscale representations of images. The derived control laws are direct visual servoing schemes and avoid the traditional steps necessary in classical geometrical feature-based visual servoing approaches: visual features detection, extraction and tracking over time. In comparison, the wavelet (respectively, shearlet) coefficients are directly injected into the control laws loop.

Several simulation and experimental validation scenarios were implemented in order to quantitatively and qualitatively assess the performance of the different proposed control laws, especially in terms of robustness and accuracy. The simulation validations have demonstrated a good behavior of all methods allowing an accuracy of \( 10^{-4}\text{mm} \) in translation and \( 10^{-4}\text{deg} \) in rotation (in nominal working conditions). However, the subsampled methods demonstrated a better robustness to external disturbances (occlusions, unstable lighting source) as compared to their non-subsampled counterparts. Based on the experimental validation scenarios, it was difficult to clearly differentiate the \textit{s-wavelet} and \textit{s-shearlet} techniques when dealing with planar scene. Both work effectively in normal conditions with an accuracy of about 0.4mm in translation and about 0.03deg in rotation. Under external disturbances, it appeared that the \textit{s-wavelet} approach is more robust than the \textit{s-shearlet} method under partial occlusions, while the \textit{s-shearlet} control law outperforms the \textit{s-wavelet} one under illumination variations. Even if the \textit{s-shearlet} method is more robust to calibration errors, the experimental validation shows a better robustness to occlusion for the \textit{s-wavelet} method.
Among the current drawbacks of the proposed control schemes is the time required for computing the wavelet- and shearlet-based interaction matrices. Indeed, the computation of wavelet and shearlet coefficients is currently carried out by open-sources libraries implemented in MATLAB. We expect that the computation time can be decreased by a factor of approximately 10 when using a C++ implementation or utilizing GPU hardware to parallelize the computations. Our last experimental scenario has also shown that our method has some difficulties to deal with 3D objects. This is certainly due to the fact that we considered a constant value of the depth for each image point in the computation of the interaction matrix used in the control law. This approximation was done since we used a monocular camera that did not provide depth information. As a result, the convergence of the current image to the desired one in case of a 3D scene can only be achieved for a small initial pose error, this is also one of the limitations of the direct visual servoing approaches, e.g. photometry-based method. Nevertheless, the formulation of the interaction matrix we derived is generic and a better adaptation of this matrix in the control law based on the knowledge of the depth of each image point may probably increase the performances, especially in case of 3D scenes. To address this limitation, future work will consist of using a RGB-D camera that also provides a depth map of the scene allowing therefore to online adapt the interaction matrix used in the control law with a good estimation of Z for each point of the RGB image.

Further work will also be undertaken to improve the developed control laws by investigating complex-valued generalizations of wavelets and shearlets. The magnitude response of complex-valued wavelet and shearlet transforms is known to exhibit a certain degree of shift invariance, which should lead to smoother cost functions and help increase the robustness of the proposed methods, especially with respect to the detail coefficients. We will also consider the applicability of Compressed Sensing techniques to exploit the sparsity properties of wavelet- and shearlet-based image representations directly during the data acquisition process.

6 Appendices

6.1 Computing $\tilde{L}_{sw}$ with Derivative Wavelets

Using integration by parts, we can move the partial derivatives in (31) from the two-dimensional image $I_t$ to the respective wavelet and write

$$\left\langle \frac{\partial I_t}{\partial x}, \psi_{j,m}^{(i)} \right\rangle = \left\langle I_t, \frac{\partial \psi_{j,m}^{(i)}}{\partial x} \right\rangle \quad \text{and} \quad \left\langle \frac{\partial I_t}{\partial y}, \psi_{j,m}^{(i)} \right\rangle = \left\langle I_t, \frac{\partial \psi_{j,m}^{(i)}}{\partial y} \right\rangle. \quad (45)$$

This means that instead of computing the wavelet transform of an approximation of the image gradient $\nabla I_t$, one can compute the wavelet transform of $I_t$ with respect to the partial derivatives of the wavelet generators $\psi^{(i)}$. In particular, for separable two-dimensional wavelet generators, the partial derivatives can easily be computed by considering the derivatives of the corresponding one-dimensional wavelet and scaling functions (4).

This approach was already investigated in (Bernard, 2001) with the goal of defining an efficient multiscale transform-based framework for computing optical flows. In particular, Bernard showed that for a sufficiently smooth scaling function $\phi \in L^2(\mathbb{R})$ and a wavelet $\psi \in L^2(\mathbb{R})$ defined by finite impulse response filters $h, g \subset \ell^2(\mathbb{Z})$ (cf. equation (8)), an MRA can be constructed that is based on a wavelet $\Psi \in L^2(\mathbb{R})$ and a
scaling function $\Phi \in L^2(\mathbb{R})$ that satisfy the following relationships:

$$
\Psi(x) = \frac{d\omega(x)}{dx},
$$
$$
\Phi(x+1) - \Phi(x) = \frac{d\phi(x)}{dx}.
$$
(46)

In (Bernard, 2001) this was achieved by computing the derivatives of $\psi$ and $\phi$ in the frequency domain by applying the relationship

$$
\hat{f}'(\omega) = i\omega \hat{f}(\omega),
$$
(47)

where $f'$ denotes the first derivative of a differentiable function $f$ and $\hat{f}$ the fourier transform of $f$.

The Fourier transforms of $\psi$ and $\phi$ are given by

$$
\hat{\phi}(\omega) = \prod_{n=1}^{\infty} m_0(2^{-n}\omega),
$$
(48)

$$
\hat{\psi}(\omega) = m_1(2^{-1}\omega)\hat{\phi}(2^{-1}\omega),
$$
(49)

where $m_0$ and $m_1$ are trigonometric polynomials defined by the filter coefficients of $h$ and $g$, that is,

$$
m_0(\omega) = 2^{-\frac{1}{2}} \sum_{n \in \mathbb{Z}} h_ne^{-in\omega},
$$
(50)

$$
m_1(\omega) = 2^{-\frac{1}{2}} \sum_{n \in \mathbb{Z}} g_ne^{-in\omega}.
$$
(51)

Note that (50) and (51) are a consequence of the scaling relations (8) and the convolution theorem, which states that convolution in the time domain is equivalent to point-wise multiplication in the frequency domain.

Eventually, it was shown in (Bernard, 2001) that the wavelet $\Psi$ and the scaling function $\Phi$ can be defined by the trigonometric polynomials

$$
\tilde{m}_0(\omega) = \frac{2m_0(\omega)}{e^{i\omega} + 1},
$$
(52)

$$
\tilde{m}_1(\omega) = \frac{e^{i\omega} - 1}{2} m_1(\omega),
$$
(53)

in the sense that

$$
\hat{\Phi}(\omega) = \prod_{n=1}^{\infty} \tilde{m}_0(2^{-n}\omega),
$$
(54)

$$
\hat{\Psi}(\omega) = \tilde{m}_1(2^{-1}\omega)\hat{\phi}(2^{-1}\omega).
$$
(55)

The following proposition, which is not part of the original work presented in (Bernard, 2001), gives a simple rule for constructing filters $\tilde{h}, \tilde{g} \in L^1(\mathbb{Z})$ associated with the trigonometric polynomials $\tilde{m}_0$ and $\tilde{m}_1$.

**Proposition 1** Let $\psi$ be a wavelet and $\phi$ be a scaling function defined by a low-pass filter $h$ and a high-pass filter $g$ through eqs. (48) to (51), then the derivative wavelet $\Psi$ and the scaling function $\Phi$ satisfying (46) are defined by the finite impulse response filters

$$
\tilde{h}_n = 2h_n - \tilde{h}_{n+1},
$$
(56)

$$
\tilde{g}_n = \frac{g_{n+1} - g_n}{2},
$$
(57)
and the trigonometric polynomials

\[ m_0(\omega) = 2^{-\frac{1}{2}} \sum_{n \in \mathbb{Z}} h_n e^{-in\omega}, \]  
(58)

\[ m_1(\omega) = 2^{-\frac{1}{2}} \sum_{n \in \mathbb{Z}} \bar{g}_n e^{-in\omega}, \]  
(59)

via eqs. (54) and (55).

**Proof:** We use the definitions of \( m_0 \) and \( m_1 \) as well as (52) and compute

\[ \sum_{n \in \mathbb{Z}} h_n e^{-in\omega} = 2^{-\frac{1}{2}} (e^{i\omega} + 1) \tilde{m}_0(\omega) \]  
(60)

\[ = \frac{1}{2} \left( \sum_{n \in \mathbb{Z}} \bar{h}_n e^{-i(n-1)\omega} + \sum_{n \in \mathbb{Z}} \bar{h}_n e^{-in\omega} \right) \]  
(61)

\[ = \sum_{n \in \mathbb{Z}} \bar{h}_{n+1} + \bar{h}_n e^{-in\omega} \]  
(62)

For \( \tilde{g} \), the definitions of the trigonometric polynomials \( m_1 \) and \( \tilde{m}_1 \) together with (53) lead to

\[ \sum_{n \in \mathbb{Z}} \bar{g}_n e^{-in\omega} = 2^{-\frac{1}{2}} (e^{i\omega} - 1) m_1(\omega) \]  
(63)

\[ = \frac{1}{2} \left( \sum_{n \in \mathbb{Z}} \bar{g}_n e^{-i(n-1)\omega} - \sum_{n \in \mathbb{Z}} \bar{g}_n e^{-in\omega} \right) \]  
(64)

\[ = \sum_{n \in \mathbb{Z}} \bar{g}_{n+1} - \bar{g}_n e^{-in\omega}. \]  
(65)

Proposition 1 provides an analytically interesting perspective on the entries of the matrix \( \tilde{L}_{kw} \) as coefficients of an MRA-based representation of the image \( I_t \). However, it does unfortunately not yield a straightforward method to compute \( \tilde{L}_{kw} \) by simply applying the filters \( h \) and \( \tilde{g} \). While \( \frac{dy(x)}{dx} \) is in fact equal to the derivative wavelet \( \Psi \), the derivative of the scaling function needs to be computed via \( \frac{df(x)}{dx} = \Phi(x+1) - \Phi(x) \) (cf. equation (46)). Furthermore, the usual assumption when representing a digital image with an MRA-based transform is that the observed pixel values correspond to coefficients defining a coarse approximation with respect to the scaling function \( \phi \) (cf. equation (9)). However, when computing the coefficients of the derivative wavelet-based MRA, we would already assume that the pixel values correspond to a coarse approximation with respect to \( \Phi \).

### 6.2 Index to Multimedia Extensions

**Acknowledgment**

This work is partially supported by DESWEEP project funded by the Région de Bretagne and from the project NEMRO (ANR-14-CE17-0013-001) funded by the ANR, France. It is also performed in the framework of the Labex ACTION (ANR-11-LABEX-01-01).
<table>
<thead>
<tr>
<th>Extension</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>video</td>
<td>video showing the global view of the experimental set-up used in this work</td>
</tr>
<tr>
<td>2</td>
<td>video</td>
<td>video showing the simulation validation The four methods are tested in three scenari: scenario 1: nominal condition scenario 2: under partial occlusions scenario 3: under illumination changes</td>
</tr>
<tr>
<td>3</td>
<td>video</td>
<td>video showing the experimental validation with a planar scene The two subsampled methods are tested in four scenari: scenario 1: nominal conditions and planar scene scenario 2: partial occlusions and planar scene scenario 3: illumination changes and planar scene</td>
</tr>
<tr>
<td>4</td>
<td>video</td>
<td>video showing the experimental validation with 3D object in nominal condition</td>
</tr>
</tbody>
</table>

References


