Interpolatory Nonlinear and Non-Separable Multi-scale Representation: Application to Image Compression

Basarab Matei
LAGA Laboratory, Paris XIII University, France
Tel:0033-1-49-40-35-71
FAX:0033-4-48-26-35-68
E-mail: matei@math.univ-paris13.fr

Sylvain Meignen*
LJK Laboratory, University of Grenoble, France
Tel:0033-4-76-51-43-95
FAX:0033-4-76-63-12-63
E-mail: sylvain.meignen@imag.fr

Anastasia Zakharova
LJK Laboratory, University of Grenoble, France
Tel:0033-4-76-51-43-95
FAX:0033-4-76-63-12-63
Email: anastasia.zakharova@gmail.com
Abstract- In this paper, we introduce the notion of nonlinear and non-separable multi-scale representation. We show how it can be derived from nonlinear and non-separable subdivision schemes associated to a non-diagonal dilation matrix. We focus on nonlinear multi-scale decomposition where the dilation matrix is either the quincunx or the hexagonal matrix. We then detail the encoding and decoding algorithm of the representation and, in particular, how the EZW (Embedded Zero-tree Wavelet) algorithm adapts in that context. Numerical experiments on image compression conclude the paper.

Keywords- Nonlinear multi-scale representation, image compression.

EDICS Category: TEC-MRS
I. Introduction

Due to the hierarchical structure of visual information, the multi-scale representations are widely applied in image processing [2][1][3]. Images are bidimensional complex objects made of homogeneous regions separated by smooth singularities. It is then natural to process them using techniques which are intrinsically bidimensional and are adapted to the information contained in an image. The multi-scale representations based on the tensor product of wavelet bases are not well adapted to process efficiently an image near the singularities. In order to have a better treatment of singularities, A. Harten in [5][6] introduced a general framework to multi-scale data representations. The idea is to associate to any function $v$, a set of sequences $\mathcal{M}v := (v^0, d^0, d^1, d^2, \ldots)$, where the sequence $v^0 := (v^0_k)_{k \in \mathbb{Z}^d}$ is the coarsest approximation of $v$ and the sequences $d^j := (d^j_k)_{k \in \mathbb{Z}^d}$, with $j \geq 0$, are additional detail coefficients which represent the fluctuations of $v$ between two successive levels of resolution.

There exist many ways to build such nonlinear representations. In most papers, the multi-scale structure is associated to dyadic levels of resolution [5][2]. In [4], the quincunx matrix is used to define the levels, and then a multi-scale representation is proposed. In the present paper, we build multi-scale representations either based on quincunx or hexagonal dilation matrices. The general philosophy of the approach, is to predict at a fine level using only coarser levels and to memorize prediction errors as detail coefficients. The approach we propose is therefore well adapted to progressive transmission of data.

We first recall some definitions on nonlinear subdivision operators in a non-separable framework and we will explain how this entails the convergence of multi-scale representations associated to such operators. We will then give applications to interpolatory multi-scale representations corresponding to quincunx or hexagonal dilation matrices. In particular, we will study affine interpolatory scheme and also interpolation using polynomial of total degree two. The whole procedure from the encoding of the representation, to the compression step and, finally, to the decoding step will be detailed. In particular, we will show how the EZW (embedded zero-tree wavelets) adapts in our context. Some illustrations on image compression conclude the paper.

II. Notations and Multi-Resolution Analysis Definition

A. Notations

Let $\ell(\mathbb{Z}^d)$ be the space of all sequences indexed by $\mathbb{Z}^d$. The subspace of bounded sequences is denoted by $\ell^\infty(\mathbb{Z}^d)$ and $\|u\|_{\ell^\infty(\mathbb{Z}^d)}$ is the supremum of $\{|u_k| : k \in \mathbb{Z}^d\}$. We denote $\ell^0(\mathbb{Z}^d)$ the subspace of all sequences with finite support (i.e. the number of non-zero components of a sequence is finite). As usual,
let $\ell^p(\mathbb{Z}^d)$ be the Banach space of sequences $u$ on $\mathbb{Z}^d$ such that $\|u\|_{\ell^p(\mathbb{Z}^d)} < \infty$, where

$$\|u\|_{\ell^p(\mathbb{Z}^d)} := \left( \sum_{k \in \mathbb{Z}^d} |u_k|^p \right)^{\frac{1}{p}} \text{ for } 1 \leq p < \infty.$$ 

For any $w \in \ell^p(\mathbb{Z}^d)^d$, that is a vector of size $d$ with each component in $\ell^p(\mathbb{Z}^d)$, we will denote $\|w\|_{\ell^p(\mathbb{Z}^d)^d}$, the supremum of the $\ell^p$-norm of the components of $w$. For a vector $u \in \mathbb{Z}^d$, $\|u\|_\infty$ denotes its sup norm in $\mathbb{Z}^d$. As in the discrete case, we denote by $L^p(\mathbb{R}^d)$ the space of all measurable functions $f$ such that $\|f\|_{L^p(\mathbb{R}^d)} < \infty$, where

$$\|f\|_{L^p(\mathbb{R}^d)} := \left( \int_{\mathbb{R}^d} |f(x)|^p dx \right)^{\frac{1}{p}} \text{ for } 1 \leq p < \infty$$

and $\|f\|_{L^\infty(\mathbb{R}^d)}$ is the essential supremum of $|f|$ on $\mathbb{R}^d$. A matrix $M$ is called a dilation matrix if it has integer entries and if $\lim_{n \to \infty} M^{-n} = 0$. In the following, the invertible dilation matrix is always denoted by $M$ and $m$ stands for $|\det(M)|$. Finally, for two positive quantities $A$ and $B$ depending on a set of parameters, the relation $A \preceq B$ implies the existence of a positive constant $C$, independent of the parameters, such that $A \leq CB$. Also $A \sim B$ means $A \preceq B$ and $B \preceq A$.

**B. Multi-Resolution Analysis**

Let $V$ be a $d$-dimensional Hilbert space. We recall the concept of multi-resolution analysis associated to the space $V$.

**Definition 1.** A multi-resolution analysis of $V$ is a sequence $(V_j)_{j \in \mathbb{Z}^d}$ of closed subspaces embedded in $V$ satisfying the following properties:

1) The subspaces are embedded $V_j \subset V_{j+1}$
2) $f \in V_j$ if and only if $f(M.) \in V_{j+1}$
3) $\bigcup_{j \in \mathbb{Z}^d} V_j = V$.
4) $\bigcap_{j \in \mathbb{Z}^d} V_j = \{0\}$
5) We assume the existence of a compactly supported function $\varphi \in V_0$ such that the family of their translates $\{\varphi(\cdot - k)\}_{k \in \mathbb{Z}^d}$ forms a Riesz basis for $V_0$.

From $V_0 \subset V_1$, it follows that the function $\varphi$, called scaling function, satisfies:

$$\varphi(x) = \sum_{n \in \mathbb{Z}^d} g_n \varphi(M x - n), \quad \text{with } \sum_n g_n = m. \quad (1)$$
To build the approximation of a given function \( v \) at level \( j \), we adopt the biorthogonal point of view that is we assume the existence of a **dual** function \( \tilde{\varphi} \) with compact support satisfying:

\[
\tilde{\varphi}(x) = \sum_{k \in \mathbb{Z}^d} h_k \tilde{\varphi}(M x - k), \quad \text{with} \quad \sum_{k} h_k = m,
\]

(2)

and such that the following duality property holds \(< \tilde{\varphi}(x - n), \varphi(x - k) >= \delta_{n,k} \), where \( \delta_{n,k} \) denotes the Kronecker symbol and \(< ., . > \) the Euclidean inner product. The approximation at level \( j \) can be obtained by projection of \( v \) on \( V_j \) as follows:

\[
v_j = \sum_{n \in \mathbb{Z}^d} v_j^n \varphi(M^j x - n)
\]

(3)

where \( v_j^n = m_j < v, \tilde{\varphi}(M^j x - k) > \).

### III. Multi-scale Representation Using Dilation Matrix \( M \)

Let \( (\Gamma^j)_{j=0,\ldots,J} \) be the set of embedded grids where \( \Gamma^j = \{M^{-j} k, k \in \mathbb{Z}^d\} \), \( J \) corresponding to the finest level of resolution. We now define the projection and prediction operators on these grids.

**A. Projection and Prediction Operators**

Let \( (\Gamma^j)_{j=0,\ldots,J} \) be the multiresolution structure defined above and then \( v^J = (v_j^k)_{k \in \mathbb{Z}^d} \) the data at the finest level \( J \). The value \( v_j^k \) is associated to the location \( M^{-j} k \) on the grid \( \Gamma^j \). In order to build the multi-scale representation of \( v^J \), we assume the existence of two discrete interscale operators:

1) the **projection** operator \( P_{j-1}^j \) acting from fine to coarse level. If \( \hat{v}^j \) is an approximation of \( v^j \) on \( \Gamma^j \), we define \( v^j = P_{j-1}^j \hat{v}^j \). This operator is always assumed to be linear. In our context, \( \tilde{\varphi} \) defined in (2) fixes the projection operator as follows:

\[
v_k^{j-1} := (m_j)^{-1} \sum_{n \in \Gamma^j} h_{n-M k} \hat{v}_n^j.
\]

(4)

2) the **prediction** operator \( P_{j-1}^j \) acting from coarse to fine level. This operator computes \( \hat{v}^j = P_{j-1}^j v^{j-1} \) and may be nonlinear.

We further assume that these two operators satisfy the **consistency** property: the projection of \( v^j \) on \( \Gamma^{j-1} \) coincides with \( v^{j-1} \), i.e.

\[
P_{j-1}^j P_{j-1}^j = I.
\]

(5)

Note that the prediction error \( e^j = v^j - \hat{v}^j \), is not arbitrary data on \( \Gamma^j \). Indeed, from (5), one has

\[
P_{j-1}^j e^j = P_{j-1}^j v^j - P_{j-1}^j \hat{v}^j = v^{j-1} - v^{j-1} = 0.
\]
Hence, \( e^j \in \text{Ker}(P^j_{j-1}) \). It follows that we may write this error in a non-redundant way using a basis of \( \text{Ker}(P^j_{j-1}) \). The coefficients of the errors in this kernel gives the detail coefficients \( d^{j-1} \). Consequently, the data \( v^j \) is completely equivalent to \( (v^{j-1}, d^{j-1}) \). Iterating this process from the initial data \( v^J \), we obtain its \textit{nonlinear multi-scale representation}

\[
\mathcal{M}v^J = (v^0, d^0, \ldots, d^{J-1}).
\]

\[
B. \text{Interpolatory Multi-scale Representations}
\]

We now introduce interpolatory multi-scale representations. In this case, the data at level \( j \) is defined by:

\[
v^j_k = v(M^{-j}k), \quad k \in \mathbb{Z}^d.
\]

From this, we directly obtain the form of the projection operator \( P^j_{j-1} \):

\[
v^{j-1}_k = v(M^{-j+1}k) = v^j_{Mk}.
\]

In a nutshell, at some resolution level \( j \), the multi-scale representation of the data \( v^j \) is built as follows: by using the prediction operator \( P^j_{j-1} \), we compute for \( v^j_{Mk+\epsilon}, \epsilon \neq 0 \) the predicted value \( (P^j_{j-1}v^{j-1})_{Mk+\epsilon} \) by using the values of \( v^{j-1} \). We stress here that the computation only involves only the values \( v^{j-1} \).

We then compute the prediction error as:

\[
e^j_{Mk+\epsilon} = v^j_{Mk+\epsilon} - (P^j_{j-1}v^{j-1})_{Mk+\epsilon}.
\]

The value \( v^j_{Mk+\epsilon} \), is completely determined by \( (P^j_{j-1}v^{j-1})_{Mk+\epsilon}, e^j_{Mk+\epsilon}, \epsilon \neq 0, k \in \mathbb{Z}^d \). By iterating, the multi-scale representation can then directly be written as \( \mathcal{M}v^J = (v^0, d^0, \ldots, d^{J-1}) \).

\[
IV. \text{THEORETICAL RESULTS ON NONLINEAR MULTI-SCALE REPRESENTATIONS}
\]

\[
A. \text{Definition of the Prediction Operator}
\]

The definition of the multi-scale representation is related to that of \( P^j_{j-1} \), the \textit{prediction operator}. In the present paper, we allow the definition and the use of \textit{quasi-linear} prediction operator defined by:

\[
\text{Definition 1. A quasi-linear prediction operator is a function which associates to each } w \in \ell^\infty(\mathbb{Z}^d) \text{ a linear operator } S(w) \text{ defined by}
\]

\[
(S(w)u)_k := \sum_{\|l\|_\infty \leq K} a_{k-Ml}(w)u_l,
\]

\[
\text{for any } u \in \ell^\infty(\mathbb{Z}^d) \text{ and } |a_{k-Ml}(w)| < C \quad \forall w.
\]
The constants $K$ and $C$ are independent of the data $w$.

Note that the recursive action of the quasi-linear prediction operator $Su := S(u)u$ on the initial data $u = u^0$, defines a nonlinear subdivision scheme:

$$w^j := Su^{j-1} = S(u^{j-1})u^{j-1} = \ldots = S(u)u^0 = S^j u^0.$$  

The general form of the prediction operator for the given data $v^{j-1}$ is then given by:

$$\hat{v}_k^j = (P_j^{j-1}v^{j-1})_k = (S(v^{j-1})v^{j-1})_k = \sum_{l \in \mathbb{Z}^d} a_{k-Ml}(v^{j-1})v_l^{j-1}.$$  

The consistency property implies that $a_{k,l}(v^{j-1})$ satisfies

$$\sum_{k \in \mathbb{Z}^d} a_{k-Mp}(v^{j-1})\delta_{k-Mi} = m\delta_{p-i}. \quad (6)$$

In an interpolatory framework, $h = m\delta_{k,0}$ and (6) amounts to $a_{Mk}(v^{j-1}) = \delta_{k,0}$.

**B. Definitions of Schemes for the Differences and of Joint Spectral Radius**

We say that the quasi-linear prediction operator $S$ reproduces the constants when:

$$\sum_{p \in \mathbb{Z}^d} a_{k-Mp}(u) = 1 \quad \forall k \in \mathbb{Z}^d \quad \text{and} \quad \forall u \in \ell^\infty(\mathbb{Z}^d).$$

Then, the following result holds:

**Proposition 1.** Let $S$ be a quasi-linear prediction operator reproducing the constants and assume $u^j = Su^{j-1}$ defined in (6). Then there exists a local and bounded operator $S_1$ such that $\nabla u^j := S_1(u^{j-1})\nabla u^{j-1}$, where $\nabla u_k = (u_k - u_{k-e_1}, \ldots, u_k - u_{k-e_d})$ and $(e_1, \ldots, e_d)$ is the canonical basis.

**Proof:** Consider

$$u_{k+e_i}^j - u_k^j = \sum_{p \in V(k+e_i) \cup V(k)} (a_{k+e_i-Mp}(u^{j-1}) - a_{k-Mp}(u^{j-1}))u_p^{j-1}$$

$$= \sum_{p \in V(k+e_i) \cup V(k)} a_{k-Mp}(u^{j-1})u_p^{j-1},$$

where $V(k) = \{ p \in \mathbb{Z}^d, \|k - Mp\|_\infty < K \}$. Since $S$ reproduces the constants, we have

$$\sum_{p \in V(k+e_i) \cup V(k)} a_{k-Mp}(u^{j-1}) = 0.$$  

We then deduce that, since

$$\{ \nabla_l \delta_{n,\beta}, n \in V(k+e_i) \cup V(k), \beta \in \mathbb{Z}^d, l = 1, \ldots, d \},$$
with \( \nabla_l u_k = u_k - u_{k-e_i} \), spans \( \{\alpha_{k-Mp}(u^{j-1})\}_{p \in V(k+e_i) \cup V(k)} \) [7]:

\[
u^j_{k+e_i} - u^j_k = \sum_{\beta \in \mathbb{Z}^d} \sum_{p \in V(k+e_i) \cup V(k)} \sum_{l=1}^d c_{p-\beta,l} \nabla_l u^j_{p-1},
\]

where \((c_k)\) is a finite sequence. Computing the differences for other directions \( e_i \), we obtain the desired result \( \blacksquare \).

C. Convergence Theorem for the Multi-scale Representation

In this section, we state a convergence result for the multi-scale representation when the matrix \( M \) is isotropic which corresponds to the following definition:

**Definition 2.** We say that a matrix \( M \) is isotropic if it is similar to the diagonal matrix \( \text{diag}(\sigma_1, \ldots, \sigma_d) \), i.e. there exists an invertible matrix \( \Lambda \) such that

\[
M = \Lambda^{-1} \text{diag}(\sigma_1, \ldots, \sigma_d) \Lambda,
\]

\((\sigma_i)_{i=1,\ldots,d}\) being the eigenvalues of matrix \( M \).

For an isotropic matrix it holds that \( |\sigma_1| = \ldots = |\sigma_d| = \sigma = m^{1/2} \). Moreover, for any given norm in \( \mathbb{R}^d \) there exist constants \( C_1, C_2 \) such that for any integer \( n \) and for any \( v \in \mathbb{R}^d \)

\[
C_1\sigma^n \|v\| \leq \|M^n v\| \leq C_2\sigma^n \|v\|.
\]

The convergence theorems we now state involve Besov spaces. Let us briefly recall the definition of such spaces. First consider the modulus of smoothness of order \( N \in \mathbb{Z} \) in \( L^p(\mathbb{Z}^d) \) of a function \( v \in L^p(\mathbb{R}^d) \), given by

\[
\omega_N(v, t)_{L^p} = \sup_{h \in \mathbb{Z}^d, \|h\| \leq t} \|\nabla^N_h v\|_{L^p(\mathbb{R}^d)},
\]

where \( \|\cdot\|_2 \) is the Euclidian norm and where \( \nabla^N_h \) is the finite difference operator of order \( N \) in the direction \( h \):

\[
\nabla^N_h v(x) := \sum_{\beta=0}^N (-1)^\beta C_N^\beta v(x - \beta h)
\]

We now recall the definition of the Besov space \( B^s_{p,q}(\mathbb{R}^d) \). Let \( N \) be any integer, \( N > s \). For \( p, q > 0 \) and \( s > 0 \), the space \( B^s_{p,q}(\mathbb{R}^d) \) contains functions \( v \in L^p(\mathbb{R}^d) \) such that \( (2^{js} \omega_N(v, 2^{-j})_{L^p})_{j \geq 0} \in \ell^q(\mathbb{Z}^d) \).

The norm in \( B^s_{p,q}(\mathbb{R}^d) \) is then given by \( \|v\|_{B^s_{p,q}(\mathbb{R}^d)} := \|v\|_{L^p(\mathbb{R}^d)} + \|(2^{js} \omega_N(v, 2^{-j})_{L^p})_{j \geq 0}\|_{\ell^q(\mathbb{Z}^d)} \). Given \( v \) and an isotropic dilation matrix \( M \), we define the set \( v_j, j \geq 0 \) following (3) and then the nonlinear multi-scale representation. Then, we have the following direct-type theorem:
Theorem 1. If the prediction operator reproduces the constants, then for all \(0 < s < 1\), one has the inequality

\[
\|v^0\|_{\ell^\infty(\mathbb{Z}^d)} + \|\sigma^s d^{j-1}\|_{\ell^\infty(\mathbb{Z}^d)} j \geq 0 \|\ell^r(\mathbb{Z}^d) \lesssim \|v\|_{B_{\infty,q}(\mathbb{R}^d)}.
\]

PROOF: First, we show that the consistency property implies that \(\|e^j\|_{\ell^\infty(\mathbb{Z}^d)} \lesssim \|\nabla v^j\|_{\ell^\infty(\mathbb{Z}^d)^d}\). We recall that

\[
e^j_k := v^j_k - \tilde{v}^j_k = v^j_k - \sum_{\|k-M\|_\infty \leq K} a_{k-M} (v^j_k - 1) v_i^j - 1.
\]

Using (4), we write the prediction error in the form

\[
e^j_k = v^j_k - m^{-1} \sum_{\|k-M\|_\infty \leq K} a_{k-M} (v^j_k - 1) \sum_{p: \|p-M\|_\infty \leq P} \tilde{h}_{p-M} v^j_p
\]

\[
= v^j_k - m^{-1} \sum_{p: \|p-k\|_\infty \leq K+P} v^j_p \sum_{l: \|l-M\|_\infty \leq K} a_{k-M} (v^j_k - 1) \tilde{h}_{p-M} l
\]

\[
= \sum_{p \in F(k)} b_{k,p} (v^j_k - 1) v^j_p,
\]

where \(F(k) = \{p: \|p-k\|_\infty \leq P + K\}\) is a finite set for any given \(k\). Let us define, for each \(k \in \mathbb{Z}^d\), a vector \(b_k(w) = (b_{k,n}(w))_{n \in F(k)}\). By hypothesis, \(e^j = 0\) if \(v^j_k = k\), since the prediction operator reproduces the constants. Since, \(\{(\nabla \delta_{n-i})_{n \in F(k)}, l \in \mathbb{Z}^d, i = 1, \ldots, d\}\) spans the orthogonal to the family \((1, (l_1, \ldots, l_d))_{l \in F(k)}\), by denoting \(c_l(w)\) the coordinates of \(b_k(w)\) in this basis and after simple computations we obtain that

\[
e^j_k = \sum_{n \in F(k)} \sum_{i=1}^d c_{n,i}(v^j_k - 1) \nabla_i v^j_n.
\]

This proves that \(\|e^j\|_{\ell^\infty(\mathbb{Z}^d)} \lesssim \|\nabla v^j\|_{\ell^\infty(\mathbb{Z}^d)}\) since the sequence \(c\) is finitely supported. We get from this that

\[
\|v^0\|_{\ell^\infty(\mathbb{Z}^d)} + \|\sigma^s d^{j-1}\|_{\ell^\infty(\mathbb{Z}^d)} j \geq 0 \|\ell^r(\mathbb{Z}^d) \lesssim \|v^0\|_{\ell^\infty(\mathbb{Z}^d)} + \|\sigma^s d^{j-1}\|_{\ell^\infty(\mathbb{Z}^d)^d} \|_{\ell^r(\mathbb{Z}^d)} \lesssim \|v\|_{B_{\infty,q}(\mathbb{R}^d)}.\tag{7}
\]

Then remark that \(\|v^0\|_{\ell^\infty(\mathbb{Z}^d)} = \|v(\cdot)\|_{\ell^\infty(\mathbb{Z}^d)} \lesssim \|v\|_{L^\infty(\mathbb{R}^d)}\) and also, since the matrix is isotropic, we can show that \(\|\nabla v\|_{\ell^\infty(\mathbb{Z}^d)^d} \lesssim \omega_1(v, C\sigma^{-j}) \lesssim \omega_1(v, \sigma^{-j})\), the last inequality being a property of the modulus of smoothness. One can also show that \(2^j \omega_1(v, 2^{-j})_{L^p} \sim \sigma^j \omega_1(v, \sigma^{-j})_{L^p}\). Finally, replacing in equation (7), we get that:

\[
\|v^0\|_{\ell^\infty(\mathbb{Z}^d)} + \|\sigma^s d^{j-1}\|_{\ell^\infty(\mathbb{Z}^d)} j \geq 0 \|\ell^r(\mathbb{Z}^d) \lesssim \|v\|_{B_{\infty,q}(\mathbb{R}^d)}.
\]

We now study the reconstruction process of a function \(v_r\), where \(r\) stands for "reconstruction", from the coefficients \(v^0_r\) and \(d^j r\). In the case of nonlinear representations, it is still possible to study the
reconstruction algorithm of the function \( v_r \) from its coefficients by iteratively using the reconstruction step

\[
v^{j,r} = P_j^{j-1} v^{j-1,r} + e^j = P_j^{j-1} v^{j-1,r} + Ed^{j-1,r}.
\]

The function \( v_r \) is defined as the limit (when it exists) of:

\[
v_{j,r}(x) = \sum_{k \in \mathbb{Z}^d} v^{j,r}_k \varphi(M^j x - k)
\]

where \( \varphi \) was introduced in (1). To study the convergence of the reconstruction process, we first need to introduce the notion of joint spectral radius for \( S_1 \).

**Definition 3.** Let \( S_1(v) : (\ell^\infty(\mathbb{Z}^d))^d \rightarrow (\ell^\infty(\mathbb{Z}^d))^d \) be the difference operator associated to the prediction operator \( S \). Its joint spectral radius is given by

\[
\rho_\infty(S_1) := \inf_{j \geq 1} \| (S_1)^j \|^\frac{1}{j} \ell^\infty(\mathbb{Z}^d)^d.
\]

In other words, \( \rho_\infty(S_1) \) is the infimum of all \( \rho > 0 \) for which there exists \( C > 0 \) such that for all \( v \in \ell^\infty(\mathbb{Z}^d) \), one has

\[
\| \nabla S^j v \|_{\ell^\infty(\mathbb{Z}^d)^d} \leq C \rho^j \| \nabla v \|_{\ell^\infty(\mathbb{Z}^d)^d},
\]

for all \( j \geq 0 \).

The following theorem then gives a sufficient condition to the existence of the limit function in Besov spaces:

**Theorem 2.** Let \( S \) be a quasi-linear prediction operator reproducing the constants. If \( \rho_\infty(S_1) < \sigma^{-s} \) for some \( s > 0 \) and if \( (v^{0,r}, d^{0,r}, d^{1,r}, \ldots) \) are such that

\[
\| v^{0,r} \|_{\ell^\infty(\mathbb{Z}^d)} + \| (\sigma^{|j|} d^{j,r}) \|_{j \geq 0, \ell^\infty(\mathbb{Z}^d)} < \infty,
\]

and assume that \( \sum_{k \in \mathbb{Z}^d} \varphi(x - k) = 1 \), where \( \varphi \) satisfies (1), then the function \( v \) belongs to \( B^{s}_{\infty,q}(\mathbb{R}^d) \) and

\[
\| v \|_{B^{s}_{\infty,q}} \lesssim \| v^{0,r} \|_{\ell^\infty(\mathbb{Z}^d)} + \| (\sigma^{|j|} d^{j,r}) \|_{j \geq 0, \ell^\infty(\mathbb{Z}^d)}.
\]

The proof of the inverse theorem is available in [10]. Note that the inverse theorem requires that the joint spectral radius of the difference operator be smaller than 1. We will design in the following sections examples of prediction operators that satisfy this property.
D. Stability of Nonlinear Multi-scale representations

The stability of nonlinear multi-scale decomposition is strongly related to that of the associated prediction operator. We recall the important notion of the stability of quasi-linear prediction operator [2]:

**Definition 4.** A quasi-linear prediction operator is stable if for every \( v, w \) in \( \ell^\infty(\mathbb{Z}^d) \):

\[
\| S(v) - S(w) \|_{\ell^\infty(\mathbb{Z}^d)} \leq C \| v - w \|_{\ell^\infty(\mathbb{Z}^d)}
\]

where \( C \) depends in a non-decreasing way on \( \| v \|_{\ell^\infty(\mathbb{Z}^d)} \).

The stability of the quasi-linear prediction operator is essential for the stability of the multi-scale representation \( \mathcal{M}v^J \). We will emphasize when this property holds for the prediction operators we will consider, but the study of the stability of the associated multi-scale representation is beyond the goal of the present article.

V. Bidimensional Interpolatory Quasi-Linear Prediction Operators

The nonlinear representations we will study are essentially based on edge non-oscillatory (ENO) approach in an interpolatory framework. ENO methods consist in computing a cost function denoted by \( C^j(k) \) that determines the best prediction operator among a group of potential ones. These prediction operators are associated to interpolation polynomials on different stencils, the cost function \( C^j(k) \) being a function computed on the stencils. An arbitrarily small change at the round off level could lead to change the stencil chosen for the prediction. For this reason, the ENO scheme is not stable [10]. Such a drawback can be avoided using weighted-ENO (WENO) interpolation which provides a smooth transition between prediction operators. The WENO formulation is based on a convex combination of potential prediction operators given by the ENO method, that is:

\[
\hat{v}^j_k := \sum_{r=0}^{m-1} \alpha_r \hat{v}^{j,r}_k
\]

with \( \alpha_r \geq 0 \) and \( \sum_{r=0}^{m-1} \alpha_r = 1 \). A possible form for the weights is given by \( \alpha_r := \frac{a_r}{\sum_{i=0}^{m-1} a_i} \) for \( r = 0, \ldots, m-1 \) where \( a_r := \frac{1}{C^j(r)} \), where \( C^j(r) \) is the cost function of the corresponding rule. We will show that the WENO prediction is stable.

In the following, we build nonlinear multi-scale decompositions using the quincunx or the hexagonal dilation matrix which are particular cases of isotropic matrices. A similar approach using the quincunx
matrix was proposed in [4] in a lifting scheme framework, we will emphasize the differences with our method when necessary. In one dimension and in a dyadic scales framework, the lifting scheme is made of an update step at even locations and a prediction step at odd locations but, when one considers interpolatory scheme to predict, as we will do, no update is required. Since we consider an interpolatory scheme, \( h = m \delta_0 \) and the consistency property leads to \( a_{Mk} = \delta_{k,0} \). Consequently, we only define \( a_{Mk+\varepsilon} \) when \( \varepsilon \) is a non-zero coset vector of \( M \).

A. Nonlinear Affine Prediction Using the Quincunx Matrix

The quincunx matrix is defined by

\[
M = \begin{pmatrix}
-1 & 1 \\
1 & 1
\end{pmatrix},
\]

whose coset vectors are \( \varepsilon_0 = (0, 0)^T \) and \( \varepsilon_1 = (0, 1)^T \). Note that \( a_{0,0} = 1 \) since the scheme is interpolatory (this corresponds to \( v^{j}_{Mk} = v^{j-1}_{k} \)) and since the nonlinear prediction operators reproduce the constants we have \( \sum_i a_{Mi+\varepsilon} = 1 \) for all coset vectors \( \varepsilon \). We consider prediction operators based on interpolation by polynomials of degree 1 (i.e. \( a + bx + cy \)) on \( \Gamma^{j-1} \) leading the following four potential stencils

\[
\begin{align*}
V^{j,1}_{k} &= M^{-j+1}\{k, k + e_1, k + e_2\} \\
V^{j,2}_{k} &= M^{-j+1}\{k, k + e_1, k + e_1 + e_2\} \\
V^{j,3}_{k} &= M^{-j+1}\{k + e_1, k + e_2, k + e_1 + e_2\} \\
V^{j,4}_{k} &= M^{-j+1}\{k, k + e_2, k + e_1 + e_2\},
\end{align*}
\]

which in turn entails the following prediction rules:

\[
\begin{align*}
\hat{v}^{j+1}_{Mk+\varepsilon_1} &= \frac{1}{2}(v^{j-1}_{k} + v^{j-1}_{k+e_1+e_2}), \\
\hat{v}^{j+1}_{Mk+\varepsilon_2} &= \frac{1}{2}(v^{j-1}_{k+e_1} + v^{j-1}_{k+e_2}).
\end{align*}
\]

Now, as \( M^2 = 2Id \), after double iteration these subdivision schemes could be associated with a limit function of the kind:

\[
\Psi(x) = \sum_p \sum_k (a_{M})_k a_{p-k} \Psi(2x - p)
\]

where

\[
(a_{M})_k = \begin{cases} 
  a_p & \text{if } k = Mp \\
  0 & \text{otherwise},
\end{cases}
\]
and, therefore, we can draw a comparison between the different cases in terms of limit functions. On Figure 1.C, we display the limit function corresponding to the separable case (i.e. \( \Psi(x) = \prod_{i=1}^{2} \max(0, 1 - |x_i|) \)) where the predictions (9) and (10) are applied successively, while we display on Figure 1. A (resp. B), the limit function corresponding to prediction (9) (resp. (10)). To relate this to the general biorthogonal framework of section II, we shall say that the limit function corresponding to the separable case is the function \( \varphi \) and the other predictions define other limit functions that are still orthogonal to \( \tilde{\varphi} \).

We now pass on to ENO prediction, which consists in choosing between the two prediction rules (9) and (10). In [4], the choice of stencils corresponds to:

\[
\min \left( \frac{1}{2} (v_{j-1}^{Mk+\epsilon_1} + v_{j-1}^{Mk+\epsilon_2}), |v_{j}^{Mk+\epsilon_1} - \frac{1}{2} (v_{k}^{j-1} + v_{k+\epsilon_1+\epsilon_2})| \right).
\]

The above cost function assumes that the true values at level \( j \) are known before prediction which is not relevant in a progressive data transmission context. To avoid this, we define the cost function as follows:

\[
C^j(k) = \min(|v_{j-1}^{k+\epsilon_1} - v_{k+\epsilon_1+\epsilon_2}^{j-1}|, |v_{j-1}^{k+\epsilon_1} - v_{k+\epsilon_1+\epsilon_2}^{j-1}|).
\]

When the minimum of \( C^j(k) \) corresponds to the first (resp. second) argument the prediction (10) (resp. (9)) is used. The motivation for the choice of such a cost function is the following: when an edge intersect...
the cell $Q_{k}^{j-1}$ with the vertices $M^{-j+1}\{k, k + e_1, k + e_2, k + e_1 + e_2\}$ of the grid $\Gamma^{j-1}$, several cases may happen:

1) either the edge intersect $[M^{-j+1}k, M^{-j+1}(k + e_1 + e_2)]$ and $[M^{-j+1}(k + e_1), M^{-j+1}(k + e_2)]$ in which case no direction is favored.

2) or the edge intersect $[M^{-j+1}k, M^{-j+1}(k + e_1 + e_2)]$ or $[M^{-j+1}(k + e_1), M^{-j+1}(k + e_2)]$, in this case the prediction operator favors the direction which is not intersected by the edge.

When $Q_{k}^{j-1}$ is not intersected by an edge, the gain between choosing one direction or the other is very small [4]. Therefore, we will apply the stencil selection procedure only when:

$$\text{argmin}_{k'=k, k+e_1, k-e_1} \left( |v_{k'}^{j-1} - v_{k+e_1}^{j-1}| + |v_{k'}^{j-1} - v_{k+e_2}^{j-1}| + |v_{k+e_1}^{j-1} - v_{k+e_2}^{j-1}| ight)$$

$$= 1$$

or

$$\text{argmin}_{k'=k, k+e_2, k-e_2} \left( |v_{k'}^{j-1} - v_{k+e_1}^{j-1}| + |v_{k'}^{j-1} - v_{k+e_2}^{j-1}| + |v_{k+e_1}^{j-1} - v_{k+e_2}^{j-1}| ight)$$

$$= 1$$

which corresponds to the case where the average first order differences are locally maximum in either vertical or horizontal direction that is when a cell is possibly intersected by an edge. We will also test another possible choice to detect the cells where the nonlinear prediction is potentially interesting which is the following:

$$\text{argmin}_{k'=k, k+e_1, k-e_1} |v_{k'}^{j-1} - v_{k+e_1}^{j-1}| = 1 \text{ or } \text{argmin}_{k'=k, k+e_2, k-e_2} |v_{k'}^{j-1} - v_{k+e_2}^{j-1}| = 1$$

which corresponds to the case where the first order differences are locally maximum is the direction of prediction. When the cell does not satisfy this property, we will apply the linear prediction method which corresponds to apply predictions (9) and (10) successively.

Using the WENO approach, we consider the approximation in the form:

$$\tilde{v}_{Mk+e_1}^{j} = \frac{a_1}{2(a_1 + a_2)}(v_{k+e_1}^{j-1} + v_{k+e_2}^{j-1}) + \frac{a_2}{2(a_1 + a_2)}(v_{k}^{j-1} + v_{k+e_1+e_2}^{j-1})$$

with

$$a_1 = \frac{1}{\epsilon + |v_{k+e_1}^{j-1} - v_{k+e_2}^{j-1}|}, a_2 = \frac{1}{\epsilon + |v_{k}^{j-1} - v_{k+e_1+e_2}^{j-1}|}$$

(11)

for some $\epsilon > 0$. The ENO and WENO predictions are such that $\rho_{\infty}(S_1) < 1$ since they satisfy the following property:
**Proposition 2.**

- When \( k = M k' \), we can show that:

\[
\| v_{M k + \varepsilon_1}^j - v_{M k}^j \|_{l^\infty(\mathbb{Z}^2)} \leq \frac{1}{2} \| \nabla v_{M k}^j \|_{l^\infty(\mathbb{Z}^2)}^2
\]

\[
\| v_{M k + \varepsilon_1}^j - v_{M k}^j \|_{l^\infty(\mathbb{Z}^2)} \leq \| \nabla v_{M k}^j \|_{l^\infty(\mathbb{Z}^2)}^2
\]

- When \( k = M k' + \varepsilon_1 \), we can show that:

\[
\| v_{M k + \varepsilon_1}^j - v_{M k}^j \|_{l^\infty(\mathbb{Z}^2)} \leq \frac{1}{2} \| \nabla v_{M k}^j \|_{l^\infty(\mathbb{Z}^2)}^2
\]

\[
\| v_{M k + \varepsilon_1}^j - v_{M k}^j \|_{l^\infty(\mathbb{Z}^2)} \leq \| \nabla v_{M k}^j \|_{l^\infty(\mathbb{Z}^2)}^2
\]

The proof is available in Appendix B. The same result can be obtained for other directions on the grid \( \Gamma^j \) (i.e., \( v_{M k + \varepsilon_1}^j - v_{M (k+e_1+e_2)}^j \) or \( v_{M k + \varepsilon_1}^j - v_{M (k+e_2)}^j \) for \( i = 1, 2 \)), which proves that \( \rho_\infty(S_1) < 1 \) and then enables us to apply Theorem 2.

**B. Nonlinear Affine Prediction Using the Hexagonal Dilation Matrix**

We now focus on the construction of nonlinear multi-scale decompositions using as dilation matrix the hexagonal matrix

\[
M = \begin{pmatrix} 2 & 1 \\ 0 & -2 \end{pmatrix}
\]

with coset vectors \( \varepsilon_0 = (0, 0)^T, \varepsilon_1 = (1, 0)^T, \varepsilon_2 = (1, -1)^T, \varepsilon_3 = (2, -1)^T \). We compute the prediction of \( v^j \) for the different coset points using an affine interpolant of the neighboring points associated to values computed on the grid \( \Gamma^{j-1} \). To do this, we use the following four different stencils on the grid \( \Gamma^{j-1} \):

\[
V_{k}^{j,1} = M^{-j+1} \{ k, k + e_1, k + e_2 \},
\]

\[
V_{k}^{j,2} = M^{-j+1} \{ k, k + e_2, k + e_1 + e_2 \},
\]

\[
W_{k}^{j,1} = M^{-j+1} \{ k + e_1, k + e_2, k + e_1 + e_2 \},
\]

\[
W_{k}^{j,2} = M^{-j+1} \{ k, k + e_1, k + e_1 + e_2 \}.
\]

For each point of \( \Gamma^j \), we determine the stencil it belongs to, and then define the prediction as its barycentric coordinates. The prediction rules at \( M k \) and \( M k + \varepsilon_1 \) are independent of the choice of the stencil, and we always have:

\[
v_{M k}^j = v_{k}^{j-1} \quad \text{and} \quad v_{M k + \varepsilon_1}^j = \frac{1}{2} v_{k}^{j-1} + \frac{1}{2} v_{k+e_1}^{j-1},
\]
For the points of the coset defined by \( \varepsilon_2 \), we use stencils \( V^1_k \) or \( V^2_k \) leading respectively to:

\[
\begin{align*}
v_{Mk + \varepsilon_2}^{j,1} &= \frac{1}{4} v_{k+e_1}^{j-1} + \frac{1}{2} v_{k+e_2}^{j-1} + \frac{1}{4} v_k^{j-1} \\
v_{Mk + \varepsilon_2}^{j,2} &= \frac{1}{2} v_k^{j-1} + \frac{1}{4} v_{k+e_1}^{j-1} + \frac{1}{4} v_{k+e_2}^{j-1} + \frac{1}{4} v_{k+e_1+e_2}^{j-1}.
\end{align*}
\] (13)

For the points of the coset defined by \( \varepsilon_3 \), we use \( W^1_k \) or \( W^2_k \) leading respectively to:

\[
\begin{align*}
v_{Mk + \varepsilon_3}^{j,1} &= \frac{1}{4} v_{k+e_1}^{j-1} + \frac{1}{4} v_{k+e_1+e_2}^{j-1} + \frac{1}{2} v_{k+e_2}^{j-1} \\
v_{Mk + \varepsilon_3}^{j,2} &= \frac{1}{4} v_k^{j-1} + \frac{1}{4} v_{k+e_1}^{j-1} + \frac{1}{2} v_{k+e_1+e_2}^{j-1}.
\end{align*}
\] (14)

when the stencils \( W^1_k \) and \( W^2_k \) are used respectively.

This leads to four different linear prediction rules depending on the choice for the prediction operator for coset vector \( \varepsilon_2 \) and \( \varepsilon_3 \). The corresponding limit functions are shown on Figure 2. Coming back to the general biorthogonal framework given in section II, the function \( \phi \) is associated to the rules (13.1) and (14.1) and these rules define the so-called linear prediction. The described quasi-linear prediction operator satisfies \( \rho_{\infty}(S_1) < 1 \) since we have:

**Proposition 3.** The prediction defined by (12), (13), (14) satisfies:

\[
\| \nabla v_{Mk+e_1}^j \|_{(l^\infty(Z^2))^2} \leq \frac{3}{4} \| \nabla v_{Mk+e_2}^{j-1} \|_{(l^\infty(Z^2))^2}
\]
The proof is given in Appendix C.

The choice of stencil in the ENO method is determined by minimizing a certain cost function, which is different for $\varepsilon_2$ and $\varepsilon_3$ since these coset points belong to different stencils. We will consider the following cost function to predict at $\varepsilon_2$:

$$C_{\varepsilon_2}^j(k) = \min(|v_{k+e_1}^{j-1} - v_{k+e_2}^{j-1}| + |v_{k+e_1}^{j-1} - v_k^{j-1}|, |v_k^{j-1} - v_{k+e_1}^{j-1}| + |v_{k+e_1}^{j-1} - v_{k+e_2}^{j-1}|).$$

When the minimum corresponds to the first (resp. second) argument, the stencil $V_k^1$ (resp. $V_k^2$) is used. Similarly, when one considers the prediction at $\varepsilon_3$, we will compute:

$$C_{\varepsilon_3}^j(k) = \min(|v_{k+e_1}^{j-1} - v_{k+e_2}^{j-1}| + |v_{k+e_1}^{j-1} - v_{k+e_2}^{j-1}|, |v_k^{j-1} - v_{k+e_1}^{j-1}| + |v_{k+e_1}^{j-1} - v_{k+e_2}^{j-1}|).$$

When the minimum corresponds to the first (resp. second) argument, the stencil $W_k^1$ (resp. $W_k^2$) is used. As in the quincunx case, we apply the above cost function only for the cells possibly containing a cell (using $(C_1)$ as criterion, otherwise, we use the linear prediction.

As far as WENO prediction is concerned, we may write that:

$$\tilde{v}_M^{j+e_2} = \left(\frac{\alpha_1}{4} + \frac{\alpha_2}{2}\right)v_k^{j-1} + \frac{\alpha_1}{4}v_{k+e_1}^{j-1} + \left(\frac{\alpha_1}{4} + \frac{\alpha_2}{2}\right)v_{k+e_2}^{j-1} + \frac{\alpha_2}{4}v_{j-1}^{j-1},$$

$$\tilde{v}_M^{j+e_3} = \left(\frac{\alpha_2}{4} + \frac{\alpha_2}{2}\right)v_k^{j-1} + \frac{\alpha_2}{4}v_{k+e_1}^{j-1} + \left(\frac{\alpha_2}{4} + \frac{\alpha_2}{2}\right)v_{k+e_2}^{j-1} + \frac{\alpha_2}{4}v_{j-1}^{j-1},$$

with

$$a_1 = \frac{1}{\epsilon + |v_{k+e_1}^{j-1} - v_{k+e_2}^{j-1}|}, \quad a_2 = \frac{1}{\epsilon + |v_k^{j-1} - v_{k+e_1}^{j-1}|},$$

$$a_3 = \frac{1}{\epsilon + |v_{k+e_1}^{j-1} - v_{j-1}^{j-1}|}, \quad a_4 = \frac{1}{\epsilon + |v_k^{j-1} - v_{k+e_1}^{j-1}|},$$

for some $\epsilon > 0$. The WENO predictions satisfy the following stability property:

Theorem 3. If $a_c$ are defined by (11) for the quincunx matrix and by (15) for the hexagonal matrix, the corresponding WENO subdivision schemes are stable.

The proof is given in Appendix A.

C. Prediction Operators Using Higher Degree Polynomials

We now introduce prediction rules based on higher degree polynomials. The prediction rules built using an affine interpolation for a given cell are independent of neighboring cells. To consider higher degree polynomials enables to build more sophisticated prediction operators but results in spurious oscillations
close to edges due to Gibbs phenomenon. A possibility to reduce the oscillations is given by the PPH scheme \[8\]. It is described by the one dimensional prediction operator defined on a dyadic grid as follows:

\[
v_{2k+1}^{j} = \begin{cases} 
\frac{v_{k+1}^{j-1} + v_{k}^{j-2}}{2} - \frac{1}{4} \frac{\nabla^2 v_{k}^{j-1}}{\nabla^2 v_{k}^{j-1} + \nabla^2 v_{k+1}^{j-1}} v_{k+1}^{j-1} & \text{if } \nabla^2 v_{k}^{j-1} \nabla^2 v_{k+1}^{j-1} > 0, \\
\frac{v_{k}^{j-1} + v_{k+1}^{j-1}}{2} & \text{otherwise},
\end{cases}
\]

where \(\nabla^2 v_{k}^{j-1}\) is the second order divided difference based on \(x_{k}^{j-1}, x_{k+1}^{j-1}\) from this prediction operator, one can derive a bidimensional prediction operator using a tensor product approach. The convergence and the stability of the one-dimensional subdivision scheme were shown in \[8\] and in \[9\]. The convergence of the bidimensional scheme derived by tensor product remains an open issue. The PPH approach is different in essence from the prediction we introduced since it is neither quasi-linear nor non-separable.

In what follows, we consider polynomials of total degree 2 for prediction (i.e. \(a + bx + cy + dx^2 + ey^2 + f xy\)). These polynomials interpolate \(v\) on different stencils which we define below. The approach we propose next is valid both for the quincunx and the hexagonal dilation matrices. The prediction rules we consider are based on the following stencils defined on \(\Gamma^{j-1}\):

\[
V^1 = M^{-j+1} \{k, k + e_1, k + e_2, k + e_1 + e_2, k + 2e_1, k + 2e_2\} \\
V^2 = M^{-j+1} \{k, k + e_1, k + e_2, k + e_1 + e_2, k - e_1, k + 2e_2\} \\
V^3 = M^{-j+1} \{k, k + e_1, k + e_2, k + e_1 + e_2, k + 2e_1, k - e_2\} \\
V^4 = M^{-j+1} \{k, k + e_1, k + e_2, k + e_1 + e_2, k - e_1, k - e_2\}.
\]

As in the affine case, to detect edge-cells, we check property (\(C_1\)). Then, we use nonlinear affine prediction on edge-cells. Once we have dealt with edge-cells, we apply the following strategy on remaining cells:

- If a cell has an edge-cell as neighboring cell in the vertical or horizontal direction then the prediction is made using the stencil selection we describe below;
- otherwise, we apply the stencil \(V^1\) for the prediction.

Now, let us explain how we do stencil selection. For each stencil \(V^j\), defined in \(16\), we consider the triangles made of neighboring points inside that stencil. This corresponds to 6 triangles. For instance for stencil \(V^1\), the triangles are as follows: \(M^{-j+1} \{k, k + e_1, k + e_2\}, M^{-j+1} \{k, k + e_1, k + e_1 + e_2\}, M^{-j+1} \{k, k + e_2, k + e_1 + e_2\}, M^{-j+1} \{k + e_1, k + e_2, k + e_1 + e_2\}, M^{-j+1} \{k + e_1, k + 2e_1, k + e_1 + e_2\}, M^{-j+1} \{k + e_2, k + e_1 + e_2, k + 2e_2\}\). The triangles are defined the same way for other stencils.

We compute a cost function on each triangle as the sum of the first order differences along its edges (for instance, the cost function for the first triangle of \(V^1\) is \(|v_{k+e_1}^{j-1} - v_{k}^{j-1}| + |v_{k+e_2}^{j-1} - v_{k+e_1}^{j-1}| + |v_{k+e_2}^{j-1} - v_{k}^{j-1}|\).

February 12, 2010 DRAFT
The cost function $C^{j,i}(k)$ associated to $V^i$ is then the sum of the cost functions computed on all the triangles that make up $V^i$. For the prediction, we then use the stencil with minimal cost.

In the case where $M$ is the quincunx matrix, the stencils $V^i$ are used to predict $v^j_{Mk+\varepsilon_1}$, $\varepsilon_1 = (0,1)$ while when $M$ is the hexagonal matrix, the stencils are used to predict $v^j_{Mk+\varepsilon_1}$, with $\varepsilon_1 = (1,0), \varepsilon_2 = (1,-1), \varepsilon_3 = (2,-1)$. To prove the convergence of such schemes would involve the computation of all the differences in the canonical directions. However, at each point we have this time 4 potential predictions, so it would be prohibitive to detail the computations here. We are currently looking for a concise proof of the fact that these nonlinear prediction operators satisfy the condition on the joint spectral radius (i.e. $\rho_{\infty}(S_1) < 1$). Once the prediction operator is chosen, we implement the multi-scale representation $M^j v^j$.

This is what we deal with next.

VI. MULTI-SCALE REPRESENTATION WHEN $M^2 = \lambda I d$

Note first that since $M^2 = \lambda I d$, the finest resolution level $J$ is considered even. Let $\Gamma^j = \{x_{k_1}^j, y_{k_2}^j\}_{k_1,k_2=0}^{N-j-1}, x_{k_1}^j = k_1 h_{j,1}, y_{k_2}^j = k_2 h_{j,2}, h_{j,1} = 2^{-j} h_0, N_j = 2^j N_0$ where $N_0$ is an integer and $h_0 = 1/N_0$. Since $M^2 = \lambda I d$, we obtain, for $k_1, k_2 = 0, ..., N_j/\lambda - 1, x_{k_1}^j = x_{k_1}^{j-2}$ and $y_{k_2}^j = y_{k_2}^{j-2}$. The connections between the levels $J$ and $J-1$ or $J-1$ and $J-2$ are more elaborate.

A. Encoding and Decoding Algorithm

Let us detail the quincunx case, the hexagonal case can be dealt with in the same way. The following encoding and decoding algorithms were originally proposed in [11], but we recall it for the sake of consistency.

We recall that the quincunx matrix satisfies $M^2 = 2I d$. For the first step, we have for $k_2 = 0, ..., N_j - 1$

\[
(x_{2k_1}^j, y_{2k_2}^j) = (x_{k_1}^{j-1}, y_{k_2}^{j-1}), \quad k_1 = 0, ..., N_j/2 - 1 \text{ if } k_2 \text{ even},
\]

\[
(x_{2k_1-1}^j, y_{2k_2}^j) = (x_{k_1}^{j-1}, y_{k_2}^{j-1}), \quad k_1 = 1, ..., N_j/2 \text{ if } k_2 \text{ odd},
\]

and for the second

\[
(x_{k_1}^{j-1}, y_{2k_2}^j) = (x_{k_1}^{j-2}, y_{k_2}^{j-2}), \quad k_1, k_2 = 0, ..., N_j/2 - 1.
\]

The following steps are performed similarly. Let us put $N_j := N_j/2^{(J-|j|/2)}$. We consider $v_{k_1,k_2}^j = v(x_{k_1}^j, y_{k_2}^j)$, where if $j$ is even $0 \leq k_1, k_2 \leq N_j - 1$ and if $j$ is odd $0 \leq k_2 \leq 2 N_j$ and $1 \leq k_1 \leq N_j$.

The projection operators are for $j$ even:

\[
v_{k_1,k_2}^{j-1} = (P_{j-1}^j v_j)^{k_1,k_2} = v_{2k_1,k_2}^j \text{ if } k_2 \text{ is even},
\]

\[
v_{k_1,k_2}^{j-1} = (P_{j-1}^j v_j)^{k_1,k_2} = v_{2k_1-1,k_2}^j \text{ if } k_2 \text{ is odd},
\]
and for \( j \) odd
\[
v_{k_1,k_2}^{j-1} = (P_{j-1}^j v^j)_{k_1,k_2} = v_{k_1,2k_2}^j.
\]

In particular, we obtain for \( j \) even that the kernel of the projection operator is
\[
\text{Ker}(P_{j-1}^j) = \{ v^j \in V_j, v_{2k_1,k_2}^j = 0, k_2 \text{ even }, v_{2k_1-1,k_2}^j = 0, k_2 \text{ odd } \}, \text{ for } j \text{ even}
\]
\[
\text{Ker}(P_{j-1}^j) = \{ v^j \in V_j, v_{k_1,2k_2}^j = 0 \} \text{ for } j \text{ odd}.
\]

Thus, if we denote by \( e_k \) the prediction error, we will need to keep when \( j \) is even \( e_{2k_1-1,k_2}^j \) for \( k_2 \) even, \( e_{2k_1,k_2}^j \) for \( k_2 \) odd and \( e_{k_1,2k_2-1}^j \) when \( j \) is odd. A reconstruction procedure for this discretization is given by operator \( S(v^{j-1}) := S(v^{j-1})v^{j-1} \) that interpolates the data \( v^{j-1} \) at the grid points of \( \Gamma^{j-1} \).

We consider three different types of prediction rules: linear prediction, ENO and WENO prediction. Let us remark that in the algorithm given below we denote \( \tilde{v}^j = v^j \setminus v^{j-1} \), that is, for all \( j = 0, \ldots, J \), \( v^j = \bigcup_{k=1}^J \tilde{v}^k \). Then, encoding and decoding algorithms take the following form:

**Encoding Algorithm**
Quincunx Matrix

for \( j = 1, \ldots, N_j - 1 \)

for \( k, k_2 = 1, \ldots, N_j \)

\[
\hat{v}_{k_1,k_2}^{j-2} = v_{2k_1-1,2k_2-1}^{j}
\]

\[
\hat{v}_{k_1,k_2}^{j-1} = v_{2k_1,2k_2}^{j}
\]

end

for \( k_1, k_2 = 1, \ldots, N_j \)

\[
e_{k_1,k_2}^{j-1} = v_{2k_1,2k_2}^{j} - S(v^{j-2})_{2k_1,2k_2}
\]

end

for \( k_1 = 1, \ldots, N_j \)

for \( k_2 = 1, \ldots, N_j; k_2 = k_2 + 2 \)

\[
e_{k_1,2k_2}^{j} = v_{2k_1,k_2}^{j} - S(v^{j-1})_{2k_1,k_2}
\]

\[
e_{k_1,k_2+1}^{j} = v_{2k_1-1,k_2+1}^{j} - S(v^{j-1})_{2k_1-1,k_2+1}
\]

end

end

The encoding algorithm leads to the following representation: \( \mathcal{M}v^{j} = (v^{0}, d^{0}, \ldots, d^{j-1}) \), where \( d^{j} \)
was defined at the end of section III. Making the same reasoning, we can encode the multiresolution decomposition obtained with the hexagonal sampling matrix.

Then, we concatenate matrices \( v^{0}, d^{0}, \ldots, d^{j-1} \) in one matrix \( V \) of dimension \( N^j \) (that is, of the same dimension as \( v^j \)). To build \( V \), we start by writing down matrix \( v^0 \) in its left upper corner. Then, we write down the matrix \( d^1 \) to the right of matrix \( v^0 \) and then \( d^1 \) below matrices \( v^0 \) and \( d^0 \) thus getting a square matrix. We proceed in the same way (writing down matrix \( d^1 \) to the right of \( d^1 - 2 \) if \( j \) is even and below if \( j \) is odd) until \( j = J \). The result of this concatenation is represented on Figure 3.(A) for the quincunx matrix and in Figure 3.(B) for the hexagonal matrix (the subspace \( d^j \) is represented by \( D^j \) of dimension \( T_1^j \times T_2^j \)).
Therefore, we are led to encode $V$. Let us notice that in general the prediction error is larger when the resolution is low. That is, the smaller $j$, the larger the elements of sub-matrix $D^j$. This naturally leads us to use an EZW encoder.

The EZW (Embedded Zero-tree Wavelet) encoding algorithm is an encoder specially developed for wavelet transforms. It is based on progressive encoding: the data is compressed in multiple passes with increasing accuracy. The initial threshold is set to be equal to $T_0 = 2^{\lceil \log_2(\max |V(k_1, k_2)|) \rceil}$, where $\max |V(k_1, k_2)|$ means the maximum element of matrix $V$. The encoder scans next element, compares it with threshold and gives 'p', 'n', 'z' or 't' as an output; if the absolute value is bigger than the threshold, it outputs either 'p', if the value is positive or 'n', if it is negative, else it constructs a tree with the considered element as the root. If it is a zero-tree (that is, the values in the nodes are smaller or equal to the root), the output is 't', otherwise it is 'z' (isolated zero). We encode an element only if we got 'p' or 'n' as output. In this case, we put it in the so-called 'subordinate list' together with either $\frac{3T_0}{2}$, if it is bigger than $T_0$ or $-\frac{3T_0}{2}$ (for elements inferior to $-\frac{T_0}{2}$) and remove it from $V$ (replace it by 0) so that it will not be encoded again. After we have scanned all the elements, we divide the threshold $T_0$ by two and start again the procedure till the threshold is not smaller than a fixed value.

Thereby, to apply an encoder based on the same principle as EZW encoder, we need to introduce a tree structure for them. The trees are designed in such a way that every element of the sub-matrix $D^j$ is a child of its parent.
of some element of sub-matrix $D^{j-1}$, $j = 1, \ldots, 2 \log_2 N_J$ for quincunx matrix, $j = 1, \ldots, 2 \log_4 N_J$ for hexagonal matrix. That is, we get a binary tree for quincunx matrix and a quad-tree for hexagonal matrix. The tree structures for quincunx and hexagonal trees are illustrated on Figure 4.

![Tree structure for quincunx matrix](image1)

![Tree structure for hexagonal matrix](image2)

Fig. 4. (A): Tree structure for quincunx matrix, (B): Tree structure for hexagonal matrix.

It is necessary that matrix $V$ be scanned in such a way that sub-matrix $D^j$ is scanned before $D^{j+1}$ and then to scan the elements of matrix $D^j$, we use Morton scan. The EZW decoding is realized exactly in the same way as for wavelets. We now write the inverse operator $v^J = M^{-1}(v^0, d^0, \ldots, d^{J-1})$.

**Decoding Algorithm**
Quincunx Matrix

for \( j = 2, \ldots, J-1; j = j + 2 \)

for \( k_1, k_2 = 1, \ldots, N_{j-1} \)

\[
\hat{v}_{2k_1-1,2k_2-1}^{j-1} = S(v^{j-2})_{2k_1-1,2k_2-1}
\]

\[
v_{2k_1,2k_2}^{j-1} = e_{k_1,k_2}^{j-1} + S(v^{j-2})_{2k_1,2k_2}
\]

end

end

for \( k_1 = 1, \ldots, N_{j-1} \)

for \( k_2 = 1, \ldots, N_j; k_2 = k_2 + 2 \)

\[
v_{2k_1,k_2}^{j} = e_{k_1,k_2}^{j} + S(v^{j-1})_{2k_1,k_2}
\]

\[
v_{2k_1-1,k_2}^{j} = e_{k_1,k_2+1}^{j} + S(v^{j-1})_{2k_1-1,k_2+1}
\]

end

end

Hexagonal Matrix

for \( j = 2, \ldots, J-1; j = j + 2 \)

for \( k_1, k_2 = 1, \ldots, N_{j-1} \)

\[
v_{4k_1-3,4k_2-3}^{j-1} = e_{k_1,k_2}^{j-2}
\]

\[
v_{4k_1-1,4k_2-2}^{j-1} = e_{k_1,3k_2-2}^{j-1} + S(v^{j-2})_{4k_1-1,4k_2-2}
\]

\[
v_{4k_1-3,4k_2-1}^{j-1} = e_{k_1,3k_2-1}^{j-1} + S(v^{j-2})_{4k_1-3,4k_2-1}
\]

\[
v_{4k_1-1,4k_2}^{j-1} = e_{k_1,3k_2}^{j-1} + S(v^{j-2})_{4k_1-1,4k_2}
\]

end

end

for \( k_1 = 1, \ldots, N_{j-1} \)

for \( k_2 = 1, \ldots, N_j; k_2 = k_2 + 2 \)

\[
v_{4k_1-1,k_2}^{j} = e_{3k_1-1,k_2}^{j} + S(v^{j-1})_{4k_1-1,k_2}
\]

\[
v_{4k_1,k_2}^{j} = e_{3k_1,k_2}^{j} + S(v^{j-1})_{4k_1,k_2}
\]

\[
v_{4k_1-3,k_2+1}^{j} = e_{3k_1-2,k_2+1}^{j} + S(v^{j-1})_{4k_1-3,k_2+1}
\]

\[
v_{4k_1-2,k_2+1}^{j} = e_{3k_1-1,k_2+1}^{j} + S(v^{j-1})_{4k_1-2,k_2+1}
\]

\[
v_{4k_1,k_2+1}^{j} = e_{3k_1,k_2+1}^{j} + S(v^{j-1})_{4k_1,k_2+1}
\]

end

end

VII. Numerical Applications

In this section, we study the improvement brought about either by using nonlinear affine interpolant instead of linear affine interpolant or by using nonlinear quadratic interpolant instead of a linear one. In particular, we will put forward the importance of dealing nonlinearly with the finest scales especially for geometrical images. We carry out the study both for the quincunx and the hexagonal dilation matrix. We also investigate the importance of the degree of the interpolation polynomial in terms of compression results.

A. Interpolation Using Affine Polynomials

Let us now investigate the improvement given by the use of affine nonlinear prediction operators instead of linear ones on the natural 256 \times 256 images of Figure 5. We use nonlinear prediction when \( T_1^j \geq T_1 \) and linear prediction otherwise. That means that for the quincunx matrix, if we take \( T_1 = 64 \) we predict nonlinearly the last finest four subspaces, the others being predicted linearly. We made the simulations both for ENO and WENO methods for \( T_1 = 32, 64, 128 \) for the images of Figure 5.
The results we depict on Figure 6 (A) and (B) correspond to the prediction rules (9) and (10). The cells on which we predict nonlinearly either to the computation of \((C_1)\) or of \((C_2)\).

When the quincunx matrix is used, we observe a significant compression gain when the nonlinear method is used especially at high compression rate (see Figure 6 (A) and (B)). A more careful look shows that one shall predict nonlinearly a larger number of scales when the image is geometrical such as the image of peppers while the number of scales predicted nonlinearly are less importance for natural images such as the image of Lena. Furthermore, we noticed that when one uses the rule \((C_2)\) instead of \((C_1)\) to determine edge-cells, the compression gain is significant for both images and even more for the image of peppers.

The results for WENO prediction are displayed on Figure 6 (C) and (D). Since the WENO prediction is closer to the linear prediction than the ENO one, the compression gain over the linear model is significantly less important than when the ENO method is used. However, the WENO method gives a stable representation. We are currently looking for a different way to define the weights in these methods to preserve compression performance.

For the hexagonal dilation matrix, we consider \(T_1 = 16\) (only the last 4 subspaces are predicted nonlinearly) or \(T_1 = 64\) (only the last 2 subspaces are predicted nonlinearly). From Figure (7).C, we notice that the nonlinear prediction does not bring any improvement in terms of compression results. This could be the consequence of a big loss of information from one scale to another. That is, one can assume that the quality of the compression depends on \(m = |\det(M)|\). Indeed, Another possible direction for future research would be the optimization of the cost function.

Fig. 5. (A): a 256 × 256 Lena image, (B): a 256 × 256 peppers image
Fig. 6. (A): linear prediction (solid line) and ENO prediction for varying $T_1$ using either $C_1$ or $C_2$ to compute cells of interest for Lena image and when the quincunx matrix is used, (B): idem but for the image of peppers, (C): linear prediction (solid line) and WENO predictions for varying $T_1$ using either $C_1$ or $C_2$ to compute cells of interest for Lena image and when the quincunx matrix is used, (D): the same as (C) but for peppers image.

### B. The Quadratic Prediction

We investigated the improvement brought about nonlinear quadratic prediction instead of linear quadratic prediction. The improvement in terms of compression rate occurs essentially at intermediate compression rate, as shown on Figure 7 (A) and (B), whereas with the affine prediction the gain was more significant at high compression rate. We notice again that it is less important to predict nonlinearly more scales with the image of Lena than with the image of peppers for which it appears crucial to predict more scales nonlinearly to beat linear prediction (see Figure 7 (B)).

Finally, when one uses the hexagonal matrix, Figure 7 (D) shows that to use higher degree polynomial does not bring any improvement in terms of compression results. In terms of perspective, future work should To summarize, nonlinear ENO methods gives a significant improvement for the quincunx matrix while the hexagonal dilation matrix does not seem a good choice in our framework. The role of dilation matrix is to be understood as well as the appropriate choice of stencils (depending on the choice of dilation matrix).
Fig. 7. (A): The quadratic linear prediction (solid line) is compared to the nonlinear quadratic prediction when the value of $T_1$ varies for Lena image and when the quincunx matrix is used. (B): idem but for the image of peppers. (C): Affine linear prediction (solid line) is compared to the nonlinear affine ENO prediction when the value of $T_1$ varies for both the image of Lena and the image of peppers and when the hexagonal matrix is used. (D): Same computations as in (C) except that we use quadratic polynomial.

VIII. CONCLUSION

In this paper, we have presented the theory of nonlinear and non-separable multi-scale representation. We have first introduced the relation between nonlinear prediction operators and multi-scale representations. We have then shown that the convergence of the multi-scale representation is related to some properties of the joint spectral radius of the first order difference operator associated to the prediction operator. We have then built some bidimensional nonlinear multi-scale representations based on affine or quadratic interpolatory prediction operators. These were built using either the quincunx or the hexagonal matrix as dilation matrix. To show the pertinence of the approach, we have proposed an application to image compression. The compression results show a clear improvement brought about ENO methods when the quincunx matrix is used. However, when the WENO prediction is used, the compression results are very close to the linear case, which let us think that future work should involve a deeper study of the weights in that method. When the hexagonal matrix is used, we did not manage to obtain as good results as with the quincunx matrix; we believe that a the down-sampling factor (symbolized by the
determinant of the matrix) is the key point, but this still needs further studies. In terms of perspectives, we are also currently investigating potential extensions of our approach to non-interpolatory prediction operators.

**APPENDIX A**

**PROOF:** Let \( v, \tilde{v} \in L^\infty(\mathbb{Z}^2) \). By the definition of WENO prediction we have that

\[
\|a_{k-M}(v) - a_{k-M}(\tilde{v})\|_{L^\infty(\mathbb{Z}^2)} \leq \|S(v) - S(\tilde{v})\|_{L^\infty(\mathbb{Z}^2)} \leq |\alpha_1 - \tilde{\alpha}_1| + |\alpha_2 - \tilde{\alpha}_2|,
\]

where \( \alpha_1, \tilde{\alpha}_1, \alpha_2, \tilde{\alpha}_2 \) are the weights of the stencils for \( v \) and \( \tilde{v} \) respectively. For \( i = 1, 2 \) it holds that

\[
|\alpha_i - \tilde{\alpha}_i| = \left| \frac{a_i}{a_1 + a_2} - \frac{\tilde{a}_i}{\tilde{a}_1 + \tilde{a}_2} \right| \\
\leq \left| \frac{a_i - \tilde{a}_i}{a_1 + a_2} \right| + \left| \tilde{a}_i \left( \frac{1}{a_1 + a_2} - \frac{1}{\tilde{a}_1 + \tilde{a}_2} \right) \right| \\
\leq \frac{1}{a_1 + a_2} |(a_i - \tilde{a}_i) + \tilde{a}_i (a_1 - \tilde{a}_1 + a_2 - \tilde{a}_2) | \\
\leq 2 \frac{1}{a_1 + a_2} \left[ \sum_{i=1}^{2} |a_i - \tilde{a}_i| \right],
\]

then, \( \|S(v) - S(\tilde{v})\|_{L^\infty(\mathbb{Z}^2)} \leq \frac{4}{a_1 + a_2} \sum_{i=1}^{2} |a_i - \tilde{a}_i| \). By simple computations we obtain that: \( |a_i - \tilde{a}_i| = \left| \frac{b_i - \tilde{b}_i}{e^2} \right| \).

From the definition of \( b_i \) we have that \( b_i \leq \|v\|_{L^\infty(\mathbb{Z}^2)} \) and \( |b_i - \tilde{b}_i| \leq \|v - \tilde{v}\|_{L^\infty(\mathbb{Z}^2)} \) and it follows that

\[
a_1 + a_2 = \frac{1}{e+b_1} + \frac{1}{e+b_2} \geq \frac{2}{e+\|v\|_{L^\infty(\mathbb{Z}^2)}},
\]

Therefore,

\[
\sum_{i} |a_i - \tilde{a}_i| \leq \frac{\|v - \tilde{v}\|_{L^\infty(\mathbb{Z}^2)}}{e^2},
\]

and

\[
\|S(v) - S(\tilde{v})\|_{L^\infty(\mathbb{Z}^2)} \leq \frac{e+\|v\|_{L^\infty(\mathbb{Z}^2)}}{e^2} \|v - \tilde{v}\|_{L^\infty(\mathbb{Z}^2)}.
\]

**APPENDIX B**

**PROOF:** Let us consider first that \( k = M k' \) and then we compute (notice that \( \varepsilon_1 = \varepsilon_2 \)):

\[
v^{j,1}_{Mk+\varepsilon_1} - v^{j}_{Mk} = \frac{1}{2} (v^{j-1}_{k} + v^{j-1}_{k+\varepsilon_1}) - v^{j-1}_{k} = \frac{1}{2} (v^{j-1}_{Mk'+\varepsilon_1+\varepsilon_2} - v^{j-1}_{Mk'}) \\
= \frac{1}{2} (v^{j-2}_{k'+\varepsilon_2} - v^{j-2}_{k'})
\]

\[
v^{j,2}_{Mk+\varepsilon_1} - v^{j}_{Mk} = \frac{1}{2} (v^{j-1}_{Mk'+\varepsilon_1} + v^{j-1}_{Mk'+\varepsilon_2}) - v^{j-1}_{Mk'} = \frac{1}{2} (v^{j-1}_{M(k'+\varepsilon_2)+\varepsilon_2} + v^{j-1}_{M(k'+\varepsilon_2)+\varepsilon_2}) - v^{j-2}_{k'}
\]
we then have to consider the different prediction rules for the values $v_{M(k' + e)}^{j-1}$ and $v_{Mk'+e_2}^{j-1}$. If the rule (9) is used in both cases, we have:

$$v_{Mk + e_1}^{j,1} - v_{Mk}^{j} = \frac{1}{4}(v_{k-e_2}^{j-2} - v_{k}^{j-2}) + \frac{1}{2}(v_{k+e_1}^{j-2} - v_{k}^{j-2}) + \frac{1}{4}(v_{k+e_1}^{j-2} - v_{k+e_1}^{j-2}).$$

If the prediction rule (10) is used in each case, we have:

$$v_{Mk + e_1}^{j,1} - v_{Mk}^{j} = \frac{1}{4}(v_{k-e_2}^{j-2} - v_{k-e_2}^{j-2}) + \frac{1}{2}(v_{k+e_1}^{j-2} - v_{k}^{j-2}) + \frac{1}{4}(v_{k+e_2}^{j-2} - v_{k}^{j-2}).$$

If prediction (9) is used for $v_{M(k' - e_2) + e_2}^{j-1}$ and prediction (10) for $v_{Mk' + e_2}^{j-1}$, we may write:

$$v_{Mk + e_1}^{j,1} - v_{Mk}^{j} = \frac{1}{2}(v_{k+e_1}^{j-2} - v_{k}^{j-2}) + \frac{1}{4}(v_{k+e_1}^{j-2} - v_{k+e_1}^{j-2}) + \frac{1}{4}(v_{k+e_2}^{j-2} - v_{k}^{j-2}).$$

If prediction (10) is used to predict $v_{M(k' - e_2) + e_2}^{j-1}$ and prediction (9) for $v_{Mk' + e_2}^{j-1}$, we may write:

$$v_{Mk + e_1}^{j,1} - v_{Mk}^{j} = \frac{1}{4}(v_{k-e_2}^{j-2} - v_{k+e_1}^{j-2}) + \frac{1}{4}(v_{k+e_1}^{j-2} - v_{k}^{j-2}) + \frac{1}{4}(v_{k+e_1}^{j-2} - v_{k+e_1}^{j-2}).$$

From the above equality, we can deduce that when $k = M k'$:

$$\|v_{Mk+e_1}^{j,1} - v_{Mk}^{j}\|_{\ell^{\infty}(\mathbb{Z})} \leq \frac{1}{2}\|\nabla v_{j-2}\|_{L^{\infty}(\mathbb{Z})}$$

$$\|v_{Mk+e_1}^{j,2} - v_{Mk}^{j}\|_{\ell^{\infty}(\mathbb{Z})} \leq \|\nabla v_{j-2}\|_{L^{\infty}(\mathbb{Z})}.$$

Now, let us consider the case where $k = M k' + e_1$. If prediction (9) is used we may write:

$$v_{Mk+e_1}^{j,1} - v_{Mk}^{j} = \frac{1}{2}(v_{Mk'+e_2}^{j-1} + v_{M(k'+e_2)+e_2}^{j-1} - v_{Mk'+e_2}^{j-1})$$

$$= \frac{1}{2}(v_{Mk'+e_2}^{j-1} - v_{Mk'+e_2}^{j-1}).$$

This corresponds to the same situation as that previously studied, and similar computation leads to:

$$\|v_{Mk+e_1}^{j,1} - v_{Mk}^{j}\|_{\ell^{\infty}(\mathbb{Z})} \leq \|\nabla v_{j-2}\|_{L^{\infty}(\mathbb{Z})}.$$

If prediction (10) is used, we may write:

$$v_{Mk+e_1}^{j,2} - v_{Mk}^{j} = \frac{1}{2}(v_{k+e_2}^{j-2} + v_{k+e_2}^{j-2} - v_{Mk'}^{j-1}).$$

If prediction (9) is used to predict $v_{Mk'+e_1}^{j-1}$, then we have:

$$v_{Mk+e_1}^{j,2} - v_{Mk}^{j} = \frac{1}{2}(v_{k+e_2}^{j-2} - v_{k+e_2}^{j-2}),$$

while when prediction (10) is used, we get:

$$v_{Mk+e_1}^{j,2} - v_{Mk}^{j} = \frac{1}{2}(v_{k+e_2}^{j-2} - v_{k+e_2}^{j-2}).$$

From this, we deduce that:

$$\|v_{Mk+e_1}^{j,2} - v_{Mk}^{j}\|_{\ell^{\infty}(\mathbb{Z})} \leq \frac{1}{2}\|\nabla v_{j-2}\|_{L^{\infty}(\mathbb{Z})}.$$

This ends the proof.
APPENDIX C

PROOF: For the sake of simplicity, we will put $\nabla_i$ for $\nabla_{e_i}$.

- If we use $V^1_k$ and $W^1_k$ for prediction, then we have the following behavior for the differences:

1) $l = Mk$, $\nabla_1 v_{MK}^j = \frac{1}{2} \nabla_1 v_k^{j-1}$; $\nabla_2 v_{MK}^j = -\frac{1}{2} \nabla_2 v_{k+e_2}^{j-1} + \frac{1}{4} \nabla_4 v_{k+e_2}^{j-1}$

2) $l = Mk + \varepsilon_1$, $\nabla_1 v_{MK+\varepsilon_1}^j = \frac{1}{2} \nabla_1 v_k^{j-1}$; $\nabla_2 v_{MK+\varepsilon_1}^j = -\frac{1}{2} \nabla_2 v_{k+e_2}^{j-1} + \frac{1}{4} \nabla_4 v_{k+e_2}^{j-1}$

3) $l = Mk + \varepsilon_2$, $\nabla_1 v_{MK+\varepsilon_2}^j = \frac{1}{4} \nabla_1 v_k^{j-1} + \frac{1}{4} \nabla_1 v_{k+e_2}^{j-1}$; $\nabla_2 v_{MK+\varepsilon_2}^j = -\frac{1}{2} \nabla_2 v_{k+e_2}^{j-1} + \frac{1}{3} \nabla v_{k+e_2}^{j-1}$

4) $l = Mk + \varepsilon_3$, $\nabla_1 v_{MK+\varepsilon_3}^j = \frac{1}{4} \nabla_1 v_k^{j-1} + \frac{1}{4} \nabla_1 v_{k+e_2}^{j-1}$; $\nabla_2 v_{MK+\varepsilon_3}^j = -\frac{1}{2} \nabla v_{k+e_2}^{j-1} + \frac{1}{4} \nabla_1 v_{k+e_2}^{j-1}$

- When the stencils $V^2_k$ and $W^2_k$ are used for the prediction, we obtain:

1) $l = Mk$, $\nabla_1 v_{MK}^j = \frac{1}{2} \nabla_1 v_k^{j-1}$; $\nabla_2 v_{MK}^j = -\frac{1}{2} \nabla_2 v_{k+e_2}^{j-1} + \frac{1}{4} \nabla_4 v_{k+e_2}^{j-1}$

2) $l = Mk + \varepsilon_1$, $\nabla_1 v_{MK+\varepsilon_1}^j = \frac{1}{2} \nabla_1 v_k^{j-1}$; $\nabla_2 v_{MK+\varepsilon_1}^j = -\frac{1}{2} \nabla_2 v_{k+e_2}^{j-1} + \frac{1}{4} \nabla_4 v_{k+e_2}^{j-1}$

3) $l = Mk + \varepsilon_2$, $\nabla_1 v_{MK+\varepsilon_2}^j = \frac{1}{4} \nabla_1 v_k^{j-1} + \frac{1}{4} \nabla_1 v_{k+e_2}^{j-1}$; $\nabla_2 v_{MK+\varepsilon_2}^j = -\frac{1}{2} \nabla v_{k+e_2}^{j-1} + \frac{1}{3} \nabla v_{k+e_2}^{j-1}$

4) $l = Mk + \varepsilon_3$, $\nabla_1 v_{MK+\varepsilon_3}^j = \frac{1}{4} \nabla_1 v_k^{j-1} + \frac{1}{4} \nabla_1 v_{k+e_2}^{j-1}$; $\nabla_2 v_{MK+\varepsilon_3}^j = -\frac{1}{2} \nabla v_{k+e_2}^{j-1} + \frac{1}{4} \nabla_1 v_{k+e_2}^{j-1}$

- When the stencils $V^1_k$ and $W^1_k$ are used for the prediction

1) $l = Mk$, $\nabla_1 v_{MK}^j = \frac{1}{2} \nabla_1 v_k^{j-1}$; $\nabla_2 v_{MK}^j = -\frac{1}{2} \nabla_2 v_{k+e_2}^{j-1} + \frac{1}{4} \nabla_4 v_{k+e_2}^{j-1}$

2) $l = Mk + \varepsilon_1$, $\nabla_1 v_{MK+\varepsilon_1}^j = \frac{1}{2} \nabla_1 v_k^{j-1}$; $\nabla_2 v_{MK+\varepsilon_1}^j = -\frac{1}{2} \nabla_2 v_{k+e_2}^{j-1} + \frac{1}{4} \nabla_4 v_{k+e_2}^{j-1}$

3) $l = Mk + \varepsilon_2$, $\nabla_1 v_{MK+\varepsilon_2}^j = \frac{1}{4} \nabla_1 v_k^{j-1} + \frac{1}{4} \nabla_1 v_{k+e_2}^{j-1}$; $\nabla_2 v_{MK+\varepsilon_2}^j = -\frac{1}{2} \nabla v_{k+e_2}^{j-1} + \frac{1}{3} \nabla v_{k+e_2}^{j-1}$

4) $l = Mk + \varepsilon_3$, $\nabla_1 v_{MK+\varepsilon_3}^j = \frac{1}{4} \nabla_1 v_k^{j-1} + \frac{1}{4} \nabla_1 v_{k+e_2}^{j-1}$; $\nabla_2 v_{MK+\varepsilon_3}^j = -\frac{1}{2} \nabla v_{k+e_2}^{j-1} + \frac{1}{4} \nabla_1 v_{k+e_2}^{j-1}$

To complete the proof of the contractivity, we also have to compute:

$$v_{MK+\varepsilon_2}^j - v_{MK+\varepsilon_2}^j = \frac{1}{2} \nabla_2 v_{k+e_2}^{j-1} - \frac{1}{4} \nabla_4 v_{k+e_2}^{j-1}$$

$$v_{MK+\varepsilon_2}^j - v_{MK+\varepsilon_2}^j = \frac{1}{2} \nabla_2 v_{k+e_2}^{j-1} - \frac{1}{4} \nabla_4 v_{k+e_2}^{j-1}$$

and also,

$$v_{MK+\varepsilon_1}^j - v_{MK+\varepsilon_1}^j = \frac{1}{2} \nabla_1 v_{k+e_2}^{j-1}.$$
REFERENCES


