Abstract

In this project a wavelet based interpolation algorithm aimed to deal with 1-dimensional non-uniform sampling is implemented. In addition its performance in terms of approximation accuracy and computational load is analyzed. An extension to the 2-Dimensional interlaced sampling case is also implemented and its application to image resolution enhancement established.

I. INTRODUCTION

Interpolation is the process of determining the values of a function at positions lying between its samples. This is usually achieved by fitting a continuous function through the discrete input samples [1]. The choice of those fitting functions is of great importance, having direct impact on the algorithm performance. In this project a wavelet based multiresolutional basis fitting reconstruction (MBFR) algorithm, developed by Ford and Etter [2][3], is implemented. The later algorithm, initially conceived to interpolate over non-uniformly sampled 1D signals, was extended to the 2D case by Nguyen and Milanfar [4] to interpolate over a bidimensional interlaced sampling structure as an approach towards image resolution enhancement. This extension was also implemented. The details concerning its implementation as well as the results of its application to an example of image resolution enhancement are shown.

The rest of this document is organized as follows: Section II provides the necessary background information on image resolution enhancement, interlaced sampling and multiresolution analysis. In Section III the non-uniform interpolation problem is formally stated. In Section IV a wavelet based interpolation algorithm and its extension to the 2-dimensional case is described. Section V presents results on the application of the described algorithm to 1D and 2D signals. Finally, in Section VI conclusions are made.

II. NOTATION AND BACKGROUND MATERIAL

A. Image resolution enhancement and interlaced sampling fundamentals

This section aims to present some of the fundamental concepts of super-resolution image reconstruction (SRIR) and their relationship to wavelet interpolation. Although several SRIR techniques exist, the focus will be placed on the nonuniform interpolation approach. For a complete picture on the SRIR state of the art the interested reader is referred to [5]. Also a comprehensive review is available in [6].
SRIR techniques aim to restore a high-resolution (HR) image from several degraded and possibly aliased low-resolution (LR) images of the same scene. It should be stressed that there is a fundamental requirement. Sub-pixel information (i.e. information contained in a resolution higher than the actual LR image) cannot be retrieved if the set of input LR images contain the same information. That is, each LR frame should provide a different look of the same scene. The later can be achieved, for example, if each LR image is a sub-pixel shifted version one of the other. Intuitively, it can be seen that a set of integer pixel shifted LR images does not provide more information than only one LR image belonging to that set, at least not to resolve sub-pixel details.

SRIR techniques are based on the observation model depicted in Figure 1. This model describes the imaging process with an intermediate HR image step and provides a “roadmap” pointing the necessary steps to achieve the reconstruction.

![Figure 1. Observation Model [5]](image)

This model can be conveniently described in mathematical terms as

\[
y_k = DBM_k x + n_k \quad 1 \leq k \leq p
\]  

(1)

where \( D \) is a downsampling matrix, \( B \) is a blurring operator, \( M_k \) is the warping operator and \( n_k \) is a noise vector. Now, if the warping is purely translational or has been corrected to be so, the shift invariance property allows to operators \( M_k \) and \( B \) to commute [4]. Hence Equation (1) becomes

\[
y = DBx + n_k \quad 1 \leq k \leq p
\]  

(2)

Equation 2 suggest the SRIR scheme depicted in Figure 2. That is, after the LR have been registered a noisy estimate of \( Bx \) is obtained through non-uniform interpolation. The resulting image is finally denoised and deblurred to obtain an estimate for the HR image \( x \). In this project the focus will be placed on the interpolation stage where wavelets prove useful.

The set of LR sampling grids shown in Figure 2 presents a regularity in its structure that is a key point in the algorithm proposed by Nguyen and Milanfar [4]. This set of samples at LR grids positions conform a 2-Dimensional interlaced sampling scheme where each LR grid corresponds to a frame.
In an interlaced scheme, signals are sampled at points resulting of the union of two or more uniform grids, called frames. For super-resolution image reconstruction to be possible there must exist a sub-pixel shift among these frames.

Considering a set of $n$ LR frames and denoting their height and width in pixels as $h$ and $w$ the set of available data under this type of sampling can be expressed as

$$\{f(pr + \epsilon_i, qr + \epsilon_i)\}, \quad i = 0, \ldots, n$$

(3)

where $r$ is the resolution enhancement factor, $\epsilon_i, \epsilon_i \in \{0, 1, \ldots, r - 1\}$ are the vertical and horizontal shifts of frame $i$ and $p = 0, \ldots, h - 1, q = 0, \ldots, w - 1$. Figure 3 illustrates the meaning of these variables.

B. Multiresolution approximation for 1D signals

A key concept behind the wavelet theory is the representation of a signal at different scales or resolutions. This allows us to analyze a signal at different levels of detail. As a result, global trends can be examined at coarser scales, whereas local phenomena are better analyzed at fine scales. This fundamental building block of the wavelet theory is known as multiresolution approximation (MRA). A formal definition of an MRA is given below [7][8].
Definition 1 (Multiresolution approximation): A sequence \( \{V_p\}_{p \in \mathbb{Z}} \) of closed subspaces of \( L^2(\mathbb{R}) \) is said to be a (dyadic) multiresolution approximation if the following properties hold:

1) \( \forall j \in \mathbb{Z}, V_j \subset V_{j-1} \) (nesting)
2) \( \lim_{j \to -\infty} V_j = \cap_{j \in \mathbb{Z}} V_j = \{0\} \) (separation)
3) \( \lim_{j \to \infty} V_j = \text{clos}(\bigcup_{j \in \mathbb{Z}} V_j) = L^2(\mathbb{R}) \) (density)
4) \( \forall j \in \mathbb{Z}, f(t) \in V_j \iff f(2t) \in V_{j-1} \) (scale invariance)
5) \( \forall k \in \mathbb{Z}, f(t) \in V_0 \iff f(t-k) \in V_0 \) (shift invariance)
6) \( \exists \phi \) such that \( \{\phi(t-k)\}_{k \in \mathbb{Z}} \) is a Riesz basis of \( V_0 \) (shift-invariant Riesz basis)

where \( V_p \) are referred as the approximation spaces and \( \phi(t) \) is known as the scaling function.

It can be shown by using the shift and scale invariance properties and the fact that the set \( \{\phi(t-k)\}_{k \in \mathbb{Z}} \) is a Riesz basis for the approximation subspace \( V_0 \) that dilations and translations of \( \phi(t) \), \( \{\phi_{j,k}(t) = 2^{-j/2} \phi(2^{-j}t-k)\}_{k \in \mathbb{Z}} \) constitute a Riesz basis for \( V_j \). Furthermore, from the fact that \( V_0 \subset V_{-\infty} \) the scaling function satisfies the following relation known as the refinement equation:

\[
\phi(t) = \sum_{k \in \mathbb{Z}} c_k \phi(2t-k),
\]

for some set of expansion coefficients \( c_k \).

Given a function \( f(t) \in L^2(\mathbb{R}) \), the approximation at scale \( j \) is simply obtained by projecting \( f(t) \) onto the subspace \( V_j \). As the value of \( j \) decreases the successive approximations become more accurate. Thus, any function in \( L^2(\mathbb{R}) \) can be approximated with arbitrary accuracy by projecting onto the appropriate subspace \( V_j \).

It is important to note that since \( V_j \) is a proper subspace of \( V_{j-1} \) there must be some space \( W_j \) such that \( W_j \) is the algebraic complement of \( V_j \) in \( V_{j-1} \). That is, there must exist a space \( W_j \) such that:

\[
V_{j-1} = V_j \oplus W_j
\]

Therefore the difference in successive approximations \( g_j(t) = f_{j-1}(t) - f_j(t) \) is a detail signal that belongs to a wavelet subspace \( W_j \).

Equation (5) shows how the signal subspaces can be broken down into a coarse approximation space and a detail space. In view of the density property of an MRA and by successive application of Equation (5) the \( L^2(\mathbb{R}) \) space can be broken down in the following manner:

\[
L^2(\mathbb{R}) = V_J \oplus \bigoplus_{j \leq J} W_j
\]

where the \( W_j \) are known as wavelet spaces.

It can be shown that the wavelets spaces satisfy the following properties [7][8]:

1) \( \text{clos}(\bigoplus_{j \in \mathbb{Z}} W_j) = L^2(\mathbb{R}) \) (density)
2) \( \forall j \in \mathbb{Z}, x(t) \in W_j \iff x(2t) \in W_{j-1} \) (scale invariance)
3) \( \forall k \in \mathbb{Z}, x(t) \in W_0 \iff x(t-k) \in W_0 \) (shift invariance)
4) \( \exists \psi(t) \) such that \( \{ \psi(t-k) \}_{k \in \mathbb{Z}} \) is a Riesz basis of \( W_0 \) (shift-invariant Riesz basis)

As in the case of approximation spaces, a Riesz basis for a given detail space \( W_j \) can be obtained by dilations and translations of \( \psi(t) \). That is the set \( \{ \psi_{j,k}(t) = 2^{-j/2}\psi(2^{-j}t-k) \} \) constitutes a Riesz basis for \( W_j \) for some coefficients \( d_k \).

Furthermore, it can be shown that \( \psi(t) \) satisfies the following wavelet equation:

\[
\psi(t) = \sqrt{2} \sum_{k \in \mathbb{Z}} d[k] \phi(2t-k)
\]

for some coefficients \( d_k \).

C. Multiresolution approximation for 2D signals

The previously described theory of multiresolution approximation spaces can be extended to the 2-dimensional case, that is for finite energy functions \( f(t,s) \in L^2(\mathbb{R}^2) \). A multiresolution approximation of \( L^2(\mathbb{R}^2) \) is a sequence of subspaces \( \{ V_j^{(2)} \}_{j \in \mathbb{Z}} \in L^2(\mathbb{R}^2) \) which satisfies straightforward two-dimensional extension of the nesting, separation, density, scale invariance and shift invariance properties. Moreover it can be shown that there exists a unique scaling function \( \Phi(t,s) \) whose dilation and translation gives an orthonormal basis for each \( V_j^{(2)} \) [8]. A particular case of separable MRA of \( L^2(\mathbb{R}^2) \) was studied by Meyer (as described in [8] and [9]). For such 2-D MRA each subspace \( V_j^{(2)} \) can be decomposed as a tensor product of two identical subspaces of \( L^2(\mathbb{R}) \). That is

\[
V_j^{(2)} = V_j^{(1)} \otimes V_j^{(1)}.
\]

Furthermore, the scaling function \( \Phi(t,s) \) for the \( V_j^{(2)} \) spaces can be decomposed as:

\[
\Phi(t,s) = \phi(t)\phi(s),
\]
where \( \phi(t) \) is simply the scaling function of the 1-D MRA \( \{V^{(1)}_j\}_{j \in \mathbb{Z}} \).

As previously stated, the basis for \( \{V^{(2)}_j\}_{j \in \mathbb{Z}} \) is composed of dilations and translations (vertical and horizontal) of \( \Phi(t) \):

\[
\Phi(t)_{j,k,l}(t,s) = \phi_j(k(t)\phi_j(l(s)) \quad / j,k,l \in \mathbb{Z}
\]

The corresponding 2-D wavelet subspaces \( W^{(2)}_j \) are generated by three wavelets:

\[
\Psi^h(t,s) = \psi(t)\phi(s) \tag{11}
\]
\[
\Psi^v(t,s) = \phi(t)\psi(s) \tag{12}
\]
\[
\Psi^d(t,s) = \psi(t)\psi(s) \tag{13}
\]

where \( h,v,d \) stand for horizontal, vertical and diagonal. The detail information is therefore contained in this three subspaces, in \( \Psi^h \) lives the horizontal detail, in \( \Psi^v \) the vertical detail and in \( \Psi^d \) the diagonal detail.

The corresponding orthonormal wavelet basis for \( W^{(2)}_j \) is obtained by:

\[
\Psi^h_{j,k,l}(t,s) = \psi_j(k(t)\phi_j(l(s)) \tag{14}
\]
\[
\Psi^v_{j,k,l}(t,s) = \phi_j(k(t)\psi_j(l(s)) \tag{15}
\]
\[
\Psi^d_{j,k,l}(t,s) = \psi_j(k(t)\psi_j(l(s)) \quad / j,k,l \in \mathbb{Z} \tag{16}
\]

In the same fashion as in the 1-Dimensional case, any \( f(t,s) \in L^2(\mathbb{R}^2) \) can be decomposed as the direct sum of a coarse approximation plus the detail components,

\[
L^2(\mathbb{R}^2) = V^{(2)}_J \oplus \bigoplus_{j \leq J} W^{(2)}_j. \tag{17}
\]

What allows us to express any \( f(t,s) \in L^2(\mathbb{R}^2) \) as:

\[
f(t,s) = \sum_{k,l \in \mathbb{Z}} a_{J,k,l} \Phi_{J,k,l}(t,s) + \sum_{j \leq J} \sum_{k,l \in \mathbb{Z}} b^h_{j,k,l} \Psi^h_{j,k,l}(t,s) + \sum_{j \leq J} \sum_{k,l \in \mathbb{Z}} b^v_{j,k,l} \Psi^v_{j,k,l}(t,s) + \sum_{j \leq J} \sum_{k,l \in \mathbb{Z}} b^d_{j,k,l} \Psi^d_{j,k,l}(t,s) \tag{18}
\]

where

\[
a_{J,k,l} = \int \int f(t,s) \Phi_{J,k,l}(t,s) dt ds
\]
\[
b^h_{j,k,l} = \int \int f(t,s) \Psi^h_{j,k,l}(t,s) dt ds
\]
\[
b^v_{j,k,l} = \int \int f(t,s) \Psi^v_{j,k,l}(t,s) dt ds
\]
\[
b^d_{j,k,l} = \int \int f(t,s) \Psi^d_{j,k,l}(t,s) dt ds
\]
III. PROBLEM STATEMENT

The purpose of any interpolation algorithm is to determine the values of a function at positions lying between its samples. In essence the idea is to represent an arbitrary continuously defined function as a discrete sum of weighted and shifted basis functions. An important issue is the proper selection of those basis functions [1][10]. In our case we are interested in determining the values of a function at a particular resolution given a nonuniformly sampled subset of those values. In other words, suppose we are looking for $M$ uniformly distributed samples of a discrete signal $f = [f_0, f_1, \ldots, f_{M-1}]^T$, but we only have $P < M$ samples of $f$ on a nonuniformly sampled subset, indexed by $\{t_k \in \{0, \ldots, M\}, k = 0, \ldots, P-1\}$. For simplicity we will consider $t_0 < t_1 < \ldots < t_{P-1}$. So we need to interpolate values at missing points in some way that is consistent with the frequency content of the rest of the signal, given the sampling rate(s) available, as given by $\{t_k\}$ [2].

For the case of a 2-dimensional interlaced sampling grid we are given the samples
\{f(pr + \epsilon_i, qr + \epsilon_s)\}, \quad (19)

where

\[\epsilon_i, \epsilon_s \in \{0, 1, \ldots, r - 1\}, \quad p = 0, \ldots, h - 1, \quad q = 0, \ldots, w - 1, \quad i = 0, \ldots, n.\] \quad (20)

and we want to interpolate on a high resolution grid to obtain \(f(t, s)\) such that \((t, s) \in (0, 1, \ldots, hr - 1; 0, 1, \ldots, wr - 1)\).

IV. Method

A. Interpolation for nonuniformly sampled 1D signals

As indicated in Section III the essential approach will be to find a representation for \(f(t)\) that is a linear combination\(^1\) of some basis functions. Ford and Etter [2][3] propose to use an MRA approach for the selection of these basis functions. One fundamental feature of such an approach is that it takes advantage of the nonuniform sampling by using sections with dense sampling to interpolate high frequencies locally and global information to reconstruct sparser sections. In other words a representation of \(f(t)\) as in Equation (7) that best fits the available data will be sought.

We will consider the sampling grid given by the finest resolution level desired, that is the evenly spaced samples on \([0, 1, \ldots, M - 1]\), as the resolution level 0 of the MRA (i.e. \(V_0\)) which can be decomposed as:

\[V_0 = V_J \oplus \bigoplus_{j=1}^{J} W_j\]

implying that \(f(t)\) at the resolution level 0 can be expressed as:

\[f(t_i) = \sum_{k} a_{J,k} \phi_{J,k}(t_i) + \sum_{j=1}^{J} \sum_{k} b_{j,k} \psi_{j,k}(t_i)\] \quad (21)

where \(i = 0, 1, \ldots, M - 1\).

Therefore we need to find the coefficients \(a_{J,k}\) and \(b_{j,k}\) so that the right hand side of Equation (21) best fits each \(f(t_i)\). But, since we only have available \(P\) samples, the best we can do is to find those coefficients to best approach the available points. To achieve that we evaluate (21) at each available point thus constructing the following system of linear equations:

\(^1\)Strictly speaking a linear combination is a sum involving a finite quantity of terms. Since the function to represent and the basis functions that will be chosen are of finite support the approximation will involve a finite quantity of terms.
\[
f(t_0) = \sum_k a_{J,k} \phi_{J,k}(t_0) + \sum_j \sum_k b_{j,k} \psi_{j,k}(t_0)
\]
\[
f(t_1) = \sum_k a_{J,k} \phi_{J,k}(t_1) + \sum_j \sum_k b_{j,k} \psi_{j,k}(t_1)
\]
\[
\vdots
\]
\[
f(t_{P-1}) = \sum_k a_{J,k} \phi_{J,k}(t_{P-1}) + \sum_j \sum_k b_{j,k} \psi_{j,k}(t_{P-1})
\]

which by using the scaling and shifting properties of \( \phi(t) \) and \( \psi(t) \) can be expressed as:

\[
f(t_0) = \sum_k a_{J,k} \phi_{J,k}(t_0) + \sum_j \sum_k b_{j,k} \psi_{j,k}(t_0)
\]
\[
f(t_1) = \sum_k a_{J,k} \phi_{J,k}(t_1) + \sum_j \sum_k b_{j,k} \psi_{j,k}(t_1)
\]
\[
\vdots
\]
\[
f(t_{P-1}) = \sum_k a_{J,k} \phi_{J,k}(t_{P-1}) + \sum_j \sum_k b_{j,k} \psi_{j,k}(t_{P-1})
\]

It is worthy to note that only a finite numbers of terms in the preceeding summations are nonzero, this being a consequence of \( f(t) \), \( \phi(t) \) and \( \psi(t) \) having finite support. To see this consider

\[
a_{J,k} = \int_{t_{\min}}^{t_{\max}} f(t) \phi\left(\frac{t - k}{2^J}\right) dt
\]

where \([t_{\min}, t_{\max}]\) is the support of \( f(t) \).

Now, if \([0, N]\) is the support for \( \phi(t) \) →

\[
\text{if} \left[ \left( t_{\min} - k \right), \left( t_{\max} - k \right) \right] \cap [0, N] = \emptyset \rightarrow a_{J,k} = 0
\]

That is, if the intersection of support for the two functions is the empty set then \( a_{J,k} = 0 \). Therefore, to obtain the interval for \( k \) where this does not happen we need:

\[
\frac{t_{\max}}{2^J} - k \geq 0 \quad \text{and} \quad \frac{t_{\min}}{2^J} - k \leq N
\]

which is equivalent to:

\[
\left\lfloor \frac{t_{\min}}{2^J} \right\rfloor - N \leq k \leq \left\lceil \frac{t_{\max}}{2^J} \right\rceil
\]

Similar arguments can be made for \( b_{j,k} \) making use of the finite support of \( \psi(t) \).

Now, let us denote the set of shifts that lead to \( a_{J,k} \) non necessary zero as \( S_j \). Then \( S_j = \left\lfloor \frac{t_{\min}}{2^J} \right\rfloor - N, \ldots, \left\lceil \frac{t_{\max}}{2^J} \right\rceil \).

This allows us to rewrite the system of equations (23) as:
\[
f(t_0) = \sum_{k \in S_j} a_{J,k} \phi\left(\frac{t_0}{2^J} - k\right) + \sum_{j=1}^{J} \sum_{k \in S_j} b_{j,k} \psi\left(\frac{t_0}{2^j} - k\right)
\]
\[
f(t_1) = \sum_{k \in S_j} a_{J,k} \phi\left(\frac{t_1}{2^J} - k\right) + \sum_{j=1}^{J} \sum_{k \in S_j} b_{j,k} \psi\left(\frac{t_1}{2^j} - k\right)
\]
\[
\vdots
\]
\[
f(t_{p-1}) = \sum_{k \in S_j} a_{J,k} \phi\left(\frac{t_{p-1}}{2^J} - k\right) + \sum_{j=1}^{J} \sum_{k \in S_j} b_{j,k} \psi\left(\frac{t_{p-1}}{2^j} - k\right)
\]

The system of equations 26 can be conveniently expressed using vector notation as:

\[
f = G_J a_J + \sum_{j=1}^{J} H_j b_j
\]

where

\[
f \quad = \quad (f(t_i))_{i=0,\ldots,P-1},
\]
\[
G_J \quad = \quad (\phi_{j,k}(t_i))_{i=0,\ldots,P-1, j=1}^{k \in S_j}
\]
\[
a_J \quad = \quad (a_{J,k})_{k \in S_j}
\]
\[
H_j \quad = \quad (\psi_{j,k}(t_i))_{i=0,\ldots,P-1, j=1}^{k \in S_j}
\]
\[
b_j \quad = \quad (b_{j,k})_{k \in S_j}
\]

As can be seen from (28) to compute the matrices \(G_J\) and \(H_j\) we need the values of the scaling and wavelet functions at the sampling points. Usually there is not a closed expression for these functions, however we can approximate those values. One method, followed in [4], is to generate a grid of dyadic points that closely approximates to the sampling points \(\{t_i\}\). Then, the exact values of the scaling and wavelet functions at dyadic points can be easily obtained by solving an eigenproblem. Alternatively, and this is the approach followed for this project, one can approximate \(\phi(t)\) and \(\psi(t)\) by means of the cascade algorithm [7].

For practical purposes one can approximate (27) by just considering a coarse scale approximation, that is

\[
f \approx G_J a_J.
\]

Now (29) can be solved for the coefficients \(a_J\) in a least-square sense. It should be noted that (29) is a system of \(P\) equations and \(\text{card}(S_j) = \lfloor \frac{t_{\text{max}}}{2^J} \rfloor - \lceil \frac{t_{\text{min}}}{2^J} \rceil + N + 1\) (see Equation (25)) unknowns. As a consequence, the selection of the scale factor \(J\) affects directly the number of unknowns. The criteria suggested by Ford and Etter [3] is to pick the scale factor \(J\) corresponding to the finest (highest resolution) scale so that \(P > (2N + \lfloor \frac{M}{2^J} \rfloor)\), this is equivalent to choosing \(J\) as
\[ J = \lceil \log_2 \left( \frac{M}{P - 2N} \right) \rceil. \]  

(30)

This way we ensure that (29) is overdetermined and has a good condition number. Now, because of the low energy in the tails of the scaling functions \( G_J \) can become ill-conditioned if there are few samples of the signal ends [2]. Therefore, (29) is solved for a regularized least-squares [11] as:

\[ \hat{a}_J = \arg\min_{a_J} \| f - G_J a_J \| + \lambda \| a_J \| \]  

(31)

(See appendix II for details on regularization and solution of the overdetermined system of equations).

Equation (31) is equivalent to [12]:

\[ \hat{a}_J = (G_J^T G_J + \lambda I)^{-1} G_J^T f \]  

(32)

As can be observed finding the set of coefficients \( a_J \) involves the inversion of a potentially large matrix. As suggested by Nguyen and Milanfar [4] that procedure can be avoided by solving the regularized least-squares problem using conjugate gradient optimization. Details on how to formulate the problem as a quadratic optimization problem and the algorithmic implementation of the method (as in [13]) are given on appendices II-B and II-D respectively.

The approximate solution of the system (29) leads to a coarse-scale estimate at scale \( J \) of \( f \) denoted by \( \hat{f}_J \).

\[ \hat{f}_J = G_J \hat{a}_J \]  

(33)

Further detail information can be added to this approximation by considering the information not used that lives in the wavelet subspace \( W_J \). That is, an approximation for the wavelet coefficients \( b_J \) can be obtained by solving

\[ g_J = f - \hat{f}_J \]

\[ g_J = f - G_J \hat{a}_J \]

\[ g_J \approx H_J b_J \]  

(34)

Again, system (34) can be solved in a regularized least-square sense to obtain an estimate for \( b_J \), \( \hat{b}_J \). Finally the values of \( f(t) \) at the high resolution grid points \( t = 0, 1, \ldots, M - 1 \) can be estimated as:

\[ f(t) \approx \sum_{k \in S_J} \hat{a}_{J,k} \phi_{J,k}(t) + \sum_{k \in S_J} \hat{b}_{J,k} \psi_{J,k}(t), \quad t = 0, 1, \ldots, M - 1. \]  

(35)
B. Interpolation for interlaced 1D signals

As noted previously a special type of sampling for wavelet superresolution, interlaced sampling, is being considered. The previously described method to interpolate on non-uniform sampling grids can be applied to the special case of interlaced sampling. For that, suppose that we have \( n \) uniformly spaced low resolution frames. Let \( r \) be the resolution enhancement factor and \( m \) the number of samples in each frame. Therefore, the available samples are:

\[
\{ f(\epsilon_i), f((r + \epsilon_i), \ldots, f((m - 1) r + \epsilon_i) \}, \quad 0 \leq \epsilon_i < r, \quad i = 1, \ldots, n.
\]  

(36)

So, given these \( mn \) sample points \( f(t) \) is to be reconstructed for the high resolution sampling points \( t = 0, 1, \ldots, mr - 1 \). Denoting the \( i^{th} \) sample frame as \( f(i) \) and taking advantage of the vector notation the following system can be solved to obtain first the coarse scale coefficients \( a_J \):

\[
f(i) \approx G_J a_J, \quad i = 1, \ldots, n
\]  

(37)

where

\[
f(i) = (f(pr + \epsilon_i))_{p=0}^{m-1}
\]

\[
G_J(i) = (\phi_J,k(pr + \epsilon_i))_{k \in S_J}^{p=0} \ldots m-1
\]

And similarly for the detail information at scale \( J \)

\[
g_J(i) = f(i) - G_J a_J
\]

\[
g_J(i) \approx H_J b_J
\]  

(38)

where

\[
H_J(i) = (\psi_J,k(pr + \epsilon_i))_{k \in S_J}^{p=0} \ldots m-1
\]

(39)

C. Interpolation for interlaced 2D signals

The procedure to interpolate 1D interlaced signals can be extended with a moderate increase in complexity to the 2-Dimensional case. Suppose that we have a set of \( n \) low resolution frames so that each frame is \( h \) pixels high and \( w \) pixels wide. The available samples are then

\[
\{ f(pr + \epsilon_i, qr + \epsilon_i) \},
\]  

(40)

where
\[ 0, 1, \ldots, r - 1 \] \quad p = 0, \ldots, h - 1, \quad q = 0, \ldots, w - 1, \quad i = 0, \ldots, n. \quad (41) \\

So, from these \( nhw \) points on low resolution grids, the goal is to interpolate all the values of \( f(t, s) \) over the high resolution grid. That is, we want \( f(t, s) \) such that \((t, s) \in (0, 1, \ldots, hr - 1; 0, 1, \ldots, wr - 1)\).

By virtue of Equation (18) the coarse scale approximation of \( f(t, s) \) over the frame \( i \) is

\[ f(pr + \epsilon_i t, qr + \epsilon_i s) \approx \sum_{k \in S_J^t} \sum_{l \in S_J^s} a_{J,k,l} \phi_{J,k,l}(pr + \epsilon_i t, qr + \epsilon_i s) \quad (42) \]

but, since \( \Phi \) is separable

\[ f(pr + \epsilon_i t, qr + \epsilon_i s) \approx \sum_{k \in S_J^t} \sum_{l \in S_J^s} a_{J,k,l} \phi_{J,k}(pr + \epsilon_i t) \phi_{J,l}(qr + \epsilon_i s) \quad (43) \]

This double summation can be compactly noted as the Kronecker product of the 1D wavelet transform matrices as

\[ f^{(i)} \approx (G^{(i)}_J \otimes G^{(i)}_J) a_J \quad (44) \]

where \( f^{(i)} \) is a vector with the values of the \( i^{th} \) frame reordered rowwise, \( a_J \) is the vector of unknown coarse scale coefficients, and

\[ G^{(i)}_J = \left( \phi_{J,k}(pr + \epsilon_i t) \right)_{k=0, \ldots, h-1} \quad G^{(i)}_J = \left( \phi_{J,l}(qr + \epsilon_i s) \right)_{l=0, \ldots, w-1} \quad (45) \]

which can be computed in the same manner as for the 1D case.

When \( n \) frames are available (44) leads to:

\[
\begin{pmatrix}
  f^{(0)} \\
f^{(1)} \\
  \vdots \\
f^{(n-1)}
\end{pmatrix} \approx 
\begin{pmatrix}
  G^{(0)}_J \otimes G^{(0)}_J \\
  G^{(1)}_J \otimes G^{(1)}_J \\
  \vdots \\
  G^{(n-1)}_J \otimes G^{(n-1)}_J
\end{pmatrix} a_J \\
(46)
\]

As it was done for the 1D problem, this system can be solved in a regularized least-squares sense to obtain an approximate solution \( \hat{a}_J \) for the coarse scale coefficients

\[ \hat{a}_J = \text{argmin}_{a_J} \| f - G_J a_J \| + \lambda \| a_J \| \quad (47) \]

where
Now since the Kronecker product of two $u \times v$ matrices results in a $u^2 \times v^2$ matrix, $G_J$ can result in a very large matrix. Therefore, the inversion of $G_J$ is avoided and the regularized least-squares is solved again by means of the conjugate gradient method.

Once an estimation $\hat{a}_J$ for $a_J$ is calculated, the horizontal, vertical and diagonal detail coefficients can be estimated by using the residuals in the same fashion as for the 1-D case.

Let

$$H_{J_t}^{(i)} = (\psi_{J,k}(pr + \epsilon_i))_{p=0,...,h-1}^{k \in S_{J_t}} \quad H_{J_s}^{(j)} = (\psi_{J,l}(qr + \epsilon_j))_{q=0,...,w-1}^{l \in S_{J_s}}$$

The first residuals $g_J$ can be used to calculate the horizontal detail coefficients at scale $J$, $b^h_J$, by solving

$$g_J = f - G_J \hat{a}_J, \quad g_J \approx \begin{pmatrix} H_{J_t}^{(0)} \otimes G_{J_s}^{(0)} \\ H_{J_t}^{(1)} \otimes G_{J_s}^{(1)} \\ \vdots \\ H_{J_t}^{(n-1)} \otimes G_{J_s}^{(n-1)} \end{pmatrix} b^h_J$$

The remaining residuals can be used to calculate the vertical and diagonal detail coefficients $b^v_J$ and $b^d_J$.

Finally, once the coarse scale coefficients and the detail coefficients at scale $J$ are estimated the HR image can be reconstructed as

$$f_{HR} = \begin{pmatrix} f(0) \\ f(1) \\ \vdots \\ f(hw-1) \end{pmatrix} \approx (G_{J_{HR}} \otimes G_{J_{HR}}) a_J + (H_{J_{HR}} \otimes G_{J_{HR}}) b^h_J + (G_{J_{HR}} \otimes H_{J_{HR}}) b^v_J + (H_{J_{HR}} \otimes H_{J_{HR}}) b^d_J$$

where the $HR$ subindex indicates that the scaling and wavelet functions are evaluated along the high resolution grid for $G$ and $H$ respectively.

V. RESULTS

To analyze the performance of the wavelet based interpolation algorithm the 1D and 2D versions were implemented using MATLAB. In Section V-A results from the 1-dimensional version are presented while a comparison with linear and cubic interpolation algorithms is made. Section V-B presents preliminary results on the implementation of the 2-dimensional version of the algorithm.
A. 1D interpolation

The algorithm performance to interpolate over a non-uniform sample set with arbitrary time coordinates was tested. For that, a test signal was generated as a sum of two linear chirp functions. After that, the signal was uniformly sampled at 100 points ($M = 100$) and a random set of samples were kept as the available samples of the original signal. The 1D algorithm using the Daubechies wavelet with four vanishing moments (DB-4) [9] (Figure 5) was applied. The results are detailed in Figure 6.

![Fig. 5. Daubechies 4, a - Scaling function, b - Wavelet function](image)

Note in Figure 6 how the approximation is improved when the detail information at scale $J$ is added. Figure 7 details this improvement between samples 70 and 90.

During the implementation of the algorithm it was noted that its performance is notoriously affected by the selection of the regularization parameters. Before comparing this interpolation method to others a proper selection for the regularization parameters is sought. Figure 8 shows the variation of the root mean squared error between the interpolation and the original signal as the regularization parameter is changed. Each point was obtained by running the interpolation over 100 random sample sets of the test signal. Three different cases are explored at using different sample survival rates. The sample survival rate $r$ is a measure of the quantity samples that fall in the random selection from all the original samples, as $r$ increases more samples become available.

The results obtained suggest that a high value of regularization parameter, in relative terms, is desirable when few samples are available (low $r$). However, as more information from the original signal becomes available the regularization parameter should be kept low.

An equivalent test was run to study the influence of the regularization parameter used to estimate the detail coefficients at scale $J$. Results are shown in Figure 9. It can be observed that when a good portion of the samples are available it is still convenient for the regularization parameter to be above certain value.

The studied method performance was compared to linear and cubic interpolation methods in terms of root mean squared error and execution time. Figure 10 summarizes the obtained results. Again each point was obtained by running the interpolation over 100 random sample sets of the test signal.

For the wavelet-based method no advantages in terms of the two proposed metrics (RMS error and execution time) can be identified. However it should be pointed out that these results originate from only one test signal.
Fig. 6. 1D wavelet interpolation. (a) Original signal and its uniform sampling positions (×). (b) Random selection of samples to interpolate over. (c) Coarse approximation over available samples. (d) Coarse approximation + detail information at scale $J$ over available samples. (e) Coarse approximation + detail information at scale $J$ over all samples.

Fig. 7. The approximation improves when detail information at scale $J$ is added.
Fig. 8. RMS error as function of coarse approximation coefficients regularization parameter.

Fig. 9. RMS error as function of detail coefficients at scale $J$ regularization parameter.
Finally in Figure 11 the test signal along with an example of each interpolation technique are shown.

B. 2D interpolation

In this section preliminary results of the application of the 2D extension of the algorithm are shown. A $172 \times 172$ pixels version of the 'baboon','lena' and 'peppers' images were low pass filtered and subsampled by a factor of 4 to generate 16 different frames for each of them. The low pass filtered version of the 'baboon' image and three of its LR generated frames are shown in Figure 12.

Figure 13 shows the result of interpolating over 2 LR frames.

Figure 14 shows original image and the result of interpolating over 4 LR frames. Finally figures 15 and 16 show the reconstruction results for the 'lena' and 'peppers' images.

VI. CONCLUSIONS AND FUTURE WORK

In this project a 1-dimensional wavelet based interpolation technique for non-uniformly sampled data was implemented. In addition its extension to a 2-dimensional interlaced sampling scheme was implemented and its application to image resolution enhancement established.
From the performance study of the 1D technique no advantages were identified in terms of the selected metrics. However, the test was not exhaustive and only one test signal was used. Moreover it remains to be determined if there is other metric than the mean squared error that better reflects the interpolation performance.

The Daubechies-4 wavelet was selected as it was the same used by Ford and Etter [2][3], it remains to be established what would be a better choice for that important parameter. Moreover, an interesting question to answer would be if there exists an optimal choice.

Using a separable orthonormal wavelet basis the extension to the 2-dimensional interlaced sampling is relatively simple and entails a moderate increase in the computational complexity. However it should be pointed out that a considerable increase in the storage capacity is needed to store the system matrices.

The system matrices present a banded structure in which an important amount of values are zero or close to zero, it remains to be studied if such fact could be exploited in order to improve the performance in terms of speed and storage requirements.

APPENDIX I

KRONECKER PRODUCT OF MATRICES

The Kronecker product $A \otimes B$ of an $m \times n$ matrix $A$ by an $o \times p$ is the $mo \times np$ matrix defined as [14]:
Fig. 12. Low pass filtered 'baboon' image (top left) and three of the 16 generated LR frames.

Fig. 13. Sample LR frame (left) and the reconstructed image using 2 LR frames (right).
Fig. 14. Original image (left) and the reconstructed image using 4 LR frames (right).

Fig. 15. Sample LR frame (left) and reconstructed image using 2 LR frames (right).

\[
A \odot B = \begin{pmatrix}
a_{11}B & \ldots & a_{1n}B \\
\vdots & \ddots & \vdots \\
a_{m1}B & \ldots & a_{mn}B
\end{pmatrix}
\]

APPENDIX II

LEAST SQUARES SOLUTION OF THE OVERDETERMINED SYSTEM OF EQUATIONS

A. Associated normal system method

This method relies on the following fact: (See [12] for a proof)
For any linear system $Ax = b$, the associated normal system $A^T Ax = A^T b$ is consistent, and all solutions of the normal system are least squares solutions of $Ax = b$.

Then from the solution of the associated normal system $x = (A^T A)^{-1}A^T b$, so it is necessary to invert $(A^T A)$.

B. Optimization formulation

Details on how to convert the regularized least-squares problem into a multidimensional quadratic form optimization problem are shown in what follows:

First consider an overdetermined linear system of equations

$$Ax = b$$

the least-squares method seeks for $x$ such that $||Ax - b||^2$ is minimized.

Now

$$||Ax - b||^2 = (Ax - b)^T (Ax - b)$$

$$= (Ax)^T - b^T (Ax - b)$$

$$= x^T A^T - b^T (Ax - b)$$

$$= x^T A^T Ax - x^T A^T b - b^T Ax + b^T b$$

$$= x^T A^T Ax - 2x^T A^T b + b^T b$$

defining
\[ H = 2A^T A \quad \hat{b} = -2A^T b \quad a = b^T b \] (54)

the problem reduces to

\[
\text{Minimize } \left( f(x) = \frac{1}{2} x^T H x + x^T \hat{b} + a \right) \tag{55}
\]

A wide variety of techniques are available to find the solution of problem (55) without having to explicitly calculate \( H^{-1} \). The conjugate gradient method is one of them.

C. Tikhonov regularization

The Tikhonov regularization method is widely used to improve the conditioning of ill-posed problems. The regularized formulation is as follows:

\[
\text{Minimize } \left( \|Ax - b\|^2 + \lambda \|x\|^2 \right) \tag{56}
\]

where \( \lambda \) is known as the Tikhonov factor.

Now, under this new objective function the formulation becomes:

\[
\text{Minimize } \left( f(x) = \frac{1}{2} x^T \hat{H} x + x^T \hat{b} + a \right) \tag{57}
\]

with

\[
\hat{H} = H + 2\lambda I. \tag{58}
\]

D. Conjugate gradient method

The algorithmic outline of the conjugate gradient method is as follows [13]

- **STEP 1**: Input \( x_0 \) and initialize tolerance \( \epsilon \).
- **STEP 2**: Compute gradient \( g_0 \) and set search direction \( d_0 = -g_0 \), \( k = 0 \).
- **STEP 3**: Input \( H_k \), the Hessian at \( x_k \) and compute \( \alpha_k = \frac{g_k^T g_k}{d_k^T H_k d_k} \). Set \( x_{k+1} = x_k + \alpha_k d_k \) and calculate \( f_{k+1} = f(x_{k+1}) \).
- **STEP 4**: If \( \|\alpha_k d_k\| < \epsilon \) then \( x^* = x_{k+1} \). Stop.
- **STEP 5**: Compute \( g_{k+1} \) and \( \beta_k = \frac{g_{k+1}^T g_{k+1}}{g_k^T g_k} \). Generate new direction \( d_{k+1} = -g_{k+1} + \beta_k d_k \). Set \( k = k + 1 \) and go to **STEP 3**.
REFERENCES


