Eidgenössische Technische Hochschule Zürich

Ecole polytechnique fédérale de Zurich
Politecnico federale di Zurigo
Swiss Federal Institute of Technology Zurich

Geometric multiscale decompositions of dynamic low-rank matrices

P. Grohs

Research Report No. 2012-03
January 2012

Seminar für Angewandte Mathematik
Eidgenössische Technische Hochschule
CH-8092 Zürich
Switzerland

Geometric Multiscale Decompositions of Dynamic Low-Rank Matrices *

P. Grohs

February 7, 2012

Abstract

The present paper is concerned with the study of manifold-valued multiscale transforms with a focus on the Stiefel manifold. For this specific geometry we derive several formulas and algorithms for the computation of geometric means which will later enable us to construct multiscale transforms of wavelet type. As an application we study compression of piecewise smooth families of low-rank matrices both for synthetic data and also real-world data arising in hyperspectral imaging. As a main theoretical contribution we show that the manifold-valued wavelet transforms can achieve an optimal N-term approximation rate for piecewise smooth functions with possible discontinuities. This latter result is valid for arbitrary manifolds.

AMS Subject Classification: primary 42C40, 65N12 secondary 65N15, 65N30, 42C99

Keywords: Riemannian data, low-rank approximation, N-term approximation, compression, manifold-valued wavelet transforms.

Contents

1 Introduction **2**
- 1.1 Contributions . . . 3
- 1.2 Relation to Other Work . . . 3
- 1.3 Outline . . . 4
- 1.4 Notation . . . 4

2 Geometric Preliminaries **4**
- 2.1 Retractions and Averages . . . 4
- 2.2 The Stiefel Manifold . . . 7
  - 2.2.1 Connection with Low Rank Matrices . . . 7
  - 2.2.2 Metric Properties . . . 8
  - 2.2.3 Retraction Pairs . . . 8

*The research of P. Grohs was partly supported by the European Research Council under grant ERC AdG 247277.

3 Wavelet-Like Decompositions of Manifold-Valued Data **13**
3.1 Interpolating Wavelet Constructions 13
3.1.1 Linear Case 13
3.1.2 Nonlinear Case 15
3.2 Average Interpolating Wavelets 16
3.2.1 Linear Case 16
3.2.2 Nonlinear Case 17
3.3 Approximation Properties 19

4 Computational Experiments and Applications **22**
4.1 Synthetic Data 22
4.2 Hyperspectral Image Compression 22
4.3 Obtaining Smooth Low-Rank Approximations 25

1 Introduction

This paper is concerned with two themes, namely the handling of manifold-valued data in general and more specifically the efficient compression and analysis of sequences of low-rank matrices.

Low-rank matrix or tensor-valued sequences arise in many important applications, whenever low-rank approximations of a parameterized family of large matrices or tensors (which otherwise would not be accessible computationally) are considered.

Examples include, but are not limited to, signal processing (in particular video and hyperspectral imaging), latent semantic indexing in information retrieval, dynamic graphs (and the associated graph Laplacians), solutions of high-dimensional time-dependent PDE's (such as the chemical master equation or the Schrödinger equation), or more general parameter dependent linear operators.

In order to connect this specific type of data to geometry, write a matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ of rank p as

$$\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{V}^\top, \tag{1}$$

where

$$\mathbf{U},\ \mathbf{V} \in \mathbb{R}^{n\times p}, \quad \mathbf{U}^\top\mathbf{U} = \mathbf{V}^\top\mathbf{V} = \mathbf{I}_p \tag{2}$$

and

$$\mathbf{S} \in \mathbb{R}^{p\times p}.$$

One particular way to arrive at a decomposition of the form (1) is to apply a truncated singular value decomposition to $\mathbf{A}$ in which case $\mathbf{S}$ is always a diagonal matrix. Due to (2), the matrices $\mathbf{U},\ \mathbf{V}$ lie in the orthogonal Stiefel manifold $\mathrm{St}(n,p)$, see Section 2.2 below.

In this work we exploit this geometric structure to develop wavelet-type multiscale decompositions for sequences of low-rank matrices of a fixed rank. These decompositions, which are in the spirit of [35], turn out to be optimally adapted to piecewise smooth data which in practice most often occurs. We prove this for general manifolds and confirm our results numerically. As an application we develop a compression scheme for hyperspectral image data.

1.1 Contributions

The contributions of this paper are in two directions. Firstly, we study general manifold-valued multiscale decompositions and their properties. In this regard, one of our main findings is an existence theorem for geometric averaging operations based on arbitrary choices of retractions onto the manifold analogous to the exponential mapping, see Section 2.1. In [27] similar averages are developed which however require the efficient computability of the exponential and logarithm mapping of the manifold – an assumption which for instance in the case of the Stiefel manifold is not fulfilled. Based on geometric averaging operations we show how wavelet-like multiscale decompositions can be constructed. These constructions are much in the spirit of [35] but with more freedom in the choice of averaging operations.

After introducing these constructions we prove our main theoretical result. In Theorem 3.6 we show that the manifold-valued wavelet decompositions achieve an optimal best N-term approximation rate for piecewise smooth functions. For linear wavelet constructions this result is of course also true and it may be regarded as one of the pillars of their success. Our result states that this behavior can be retained in arbitrary manifolds.

Secondly, we study a number of results, algorithms and applications tailored to the specific case of the Stiefel manifold. This geometry which – as we have already outlined – is of tremendous importance for many applications is not treated in [35] and presents many challenges. For instance, [35] assumes the efficient computability of exponential and logarithm mapping, which is not given in the Stiefel manifold, especially for high dimensional matrices. As a remedy, we develop alternative retractions in Section 2.2.3 and present practical algorithms for their computation. By our general results these constructions perform as well as those based on the exponential mapping, but with a huge gain in efficiency.

To demonstrate the practical usefulness of our algorithms we develop a compression scheme for hyperspectral image data in Section 4.2.

1.2 Relation to Other Work

To put this paper in perspective we comment on some related work. In regards to multiscale analysis of geometric data there exists by now a substantial body of work [35, 21, 32, 38, 20]. Our paper builds on these results and ideas. On the other hand, low rank approximation is probably the oldest and by far most popular and significant model reduction technique in computational science.

The theory of dynamic low-rank approximation resulting in piecewise smooth families of low-rank matrices has been studied in [28, 29, 33, 2, 4, 12]. We also refer to this work for further information on application areas.

Structured low-rank approximation has been studied in [7]. We expect our work to be extendible to decompositions of the type studied in that paper.

Other applications of data with values in the Stiefel manifold and related geometries can be found in [1, 18, 24].

Even though in the present paper we confine the discussion to low-rank matrices, we expect an extension to the tensor case to be feasible, see also [29].

1.3 Outline

To facilitate the reading we give a brief outline of the paper. In Section 2.1 we start by defining geometric notions of averages in general manifolds. The main result of this section is a proof of well-definedness of these constructions, extending classical results in [27]. After that, in Section 2.2.3 we study specific choices for retractions on the Stiefel manifold which are necessary in the definition of a geometric average. We obtain explicit and to our knowledge new expressions and algorithms for several computationally attractive retractions and their inverses.

In Sections 3.2 and 3.1 we give two constructions of multiscale transforms of manifold-valued functions, inspired by [35] but based on our more general averaging operations. Then, in Section 3.3 we prove our main theoretical result, namely that these constructions attain an optimal approximation rate for piecewise smooth functions.

Section 4 contains applications and computational experiments, first with synthetic data and later, in Section 4.2, we present a method for the compression of hyperspectral image data. Finally, in Section 4.3 we comment on the interesting work [28] for the purpose of obtaining piecewise smooth low-rank approximations to a given matrix curve.

1.4 Notation

We use the symbol $A \lesssim B$ to describe that a quantity A is bounded by a fixed constant times the quantity B. Matrices are always written in boldface letters. We use the symbol $\mathbf{I}_n$ for the $n \times n$ identity matrix and $\mathbf{0}_n$ for the $n \times n$ zero matrix.

2 Geometric Preliminaries

The present section contains basic geometric facts which we will use later on. In the first Section 2.1 we treat general manifolds and exhibit useful methods of defining intrinsic notions of a weighted average of points. These results universally apply to any Riemannian manifold and are of independent interest.

Then, in Section 2.2 we specify these findings to the geometry of the Stiefel manifold. In particular we construct several useful notions of retractions which, along with their inverses, turn out to be efficiently computable.

2.1 Retractions and Averages

For now we consider a general Riemannian manifold M with metric $\langle \cdot, \cdot \rangle_M$. We assume some familiarity with the basics of Riemannian geometry, a good entrance point into this topic is [13]. An important operation in our analysis will be the addition of a tangent vector to a point. Classically, this is done via the exponential mapping $\exp_x(\xi)$ which maps a tangent vector $\xi \in T_x M$ to the endpoint of the (locally) unique geodesic curve emanating from x with initial speed ξ. The exponential mapping also possesses an inverse which maps two points $x,\ y \in M$ to the (locally) unique vector $\log(x, y) \in T_x M$ such that

$$\exp_x (\log(x, y)) = y. \tag{3}$$

In order to arrive at computationally efficient algorithms we are also considering alternative pairs (f, g) of functions, analogous to the pair $(\exp, \log)$.

Definition 2.1. *A pair (f, g) of functions*

$$f : TM \to M, \quad g : M \times M \tag{4}$$

is called a retraction pair *if*

$$f(x, g(x, y)) = y, \quad \text{for all } x,\ y \in M, \tag{5}$$

and

$$f(x, 0) = x,\ \frac{\partial}{\partial \xi} f(x, 0) = Id \quad \text{for all } x \in M. \tag{6}$$

In general f will only be locally defined around M and g around the diagonal of $M \times M$. Certainly, the pair $(\exp, \log)$ satisfies the above assumptions [13] and therefore forms a retraction pair.

Given a retraction pair (f, g), points x_i, $i = 1, \ldots, d$ and weights $(w_i)_{i=1}^d \in \mathbb{R}$ such that

$$\sum_{i=1}^{d} w_i = 1 \tag{7}$$

we define the average

$$\mathrm{av}_g\left((w_i), (x_i)\right)$$

as the unique point x_* which satisfies

$$\sum_{i=1}^{d} w_i g(x_*, x_i) = 0. \tag{8}$$

Remark 2.2. *The motivation for (8) comes from the linear theory, where $g(x, y) = y - x$.*

We now show the important fact that for any retraction pair, the definition (8) yields a well-defined notion of average.

Proposition 2.3. *The average is locally well-defined and unique for any retraction pair (f, g).*

Even if the Riemannian metric does not play a role in Proposition 2.3, we employ it as an auxiliary device in the proof because we want to make use of the following result which is proven in [23, Corollary 2.6 (a)].

Theorem 2.4. *Assume that $B_\rho(p)$ is a convex geodesic ball around $x \in M$ with ρ small enough (see [23] for precise details). There exists $\varepsilon > 0$ such that for every vector field Υ defined on B_ρ such that*

$$\|\Upsilon\| \leq \varepsilon \text{ and } \|Id + \nabla \Upsilon\| \leq \varepsilon$$

(∇ denoting the covariant derivative), there exists a unique point $x_* \in B_\rho$ such that

$$\Upsilon(x_*) = 0.$$

Proof of Proposition 2.3. A short computation involving (6) and (5) shows that by putting

$$\Upsilon_y(x) := g(x, y)$$

to be the vector field of difference vectors pointing to $x \in M$, we get

$$\nabla_\xi \Upsilon_x(x) = -\xi, \quad x \in M,\ \xi \in T_x M, \tag{9}$$

where ∇ denotes the covariant derivative. Indeed, this may be seen by first differentiating (5) w.r.t. x (if necessary in a chart) which yields

$$\frac{\partial}{\partial x} f(x, g(x, y)) + \frac{\partial}{\partial \xi} f(x, g(x, y)) \frac{\partial}{\partial x} g(x, y) = 0.$$

Using (6) this implies that with $x = y$

$$Id + \frac{\partial}{\partial x} g(x, y) = 0.$$

Finally, we note that, using the fact that $\Upsilon_y(y) = 0$, we deduce that

$$\frac{\partial}{\partial x} \Upsilon_y(x)\big|_{x=y} \xi = \nabla_\xi \Upsilon_y(y),$$

which gives (9). Therefore, since additionally $g(y, y) = 0$, and using (7), we have for all $\varepsilon > 0$ and provided that the points x_i are all contained in a sufficiently small geodesic ball B, that

$$\left\| \sum_{i=1}^{d} w_i g(x, x_i) \right\| \leq \varepsilon \tag{10}$$

and

$$\left\| Id + \nabla \sum_{i=1}^{d} w_i \Upsilon_{x_i}(x) \right\| \leq \varepsilon. \tag{11}$$

Equations (10) and (11) suffice to establish the claim, using Theorem 2.4. □

Remark 2.5. *It might appear puzzling that in our proof we used a property (9) of the covariant derivative $\nabla \Upsilon_y$ even though the statement of the result is completely independent of the choice of a Riemannian metric on M. The justification lies in the fact that, since $\Upsilon_x(x) = 0$, the expression (9) is independent of the metric, after all.*

We have used the results of [23] in our proof of well-definedness of the averaging operation. In this work one can also find algorithms for the computation of $\mathrm{av}_g((w_i), (x_i))$. For completeness we mention a simple iterative fixed-point method in Algorithm 1, which converges to the desired average. A better choice would be a Newton-type method, see again [23] and also [1] for more information. In our experiments we have found that the iteration in Algorithm 1 typically requires $2 - 3$ iterations until it reaches machine precision.

Algorithm 1: Calculate $\mathrm{av}_g\left((w_i),(x_i)\right)$.

$y \leftarrow x_1$
while $\left\|\sum_{i=1}^d w_i g(y,x_i)\right\| > \mathrm{TOL}$ **do**
  $y \leftarrow f\left(y, \sum_{i=1}^d w_i g(y,x_i)\right)$
end while
return y

2.2 The Stiefel Manifold

We now specify our previous findings to the geometry of the Stiefel manifold. But first we start with a short motivation why we are interested in this particular geometry. After that we will exhibit several useful choices for retractions.

2.2.1 Connection with Low Rank Matrices

The fact that geometry plays a role in low rank approximation has already been noted in [1, 18] (and possibly much earlier elsewhere) where it is observed that the space

$$\mathcal{M}_{n,p} := \left\{\mathbf{A} \in \mathbb{R}^{n\times n} : \ \mathrm{rank}\,(\mathbf{A}) = p\right\}$$

of $n \times n$ matrices of rank p admits a manifold structure. In this paper we will actually not use this structure but factorizations of the form (1) with matrices $\mathbf{U},\ \mathbf{V}$ which lie on the *Stiefel manifold*

$$\mathrm{St}(n,p) := \left\{\mathbf{X} \in \mathbb{R}^{n\times p} : \ \mathbf{X}^\top\mathbf{X} = \mathbf{I}_p\right\}.$$

The main result which enables us to do this is the existence of a smooth singular value decomposition:

Theorem 2.6 ([6], Theorem 2.4)**.** *Assume that $\gamma : t \mapsto \mathbf{A}(t) \in \mathcal{M}_{n,p}$ is a C^s curve. Then there exist C^s matrix curves*

$$\gamma_V : t \mapsto \mathbf{U}(t) \in \mathrm{St}(n,p),\ \gamma_V : t \mapsto \mathbf{V}(t) \in \mathrm{St}(n,p),\ \gamma_S : t \mapsto \mathbf{S}(t) \in \mathbb{R}^{p\times p}$$

such that

$$\mathbf{A}(t) = \mathbf{U}(t)\mathbf{S}(t)\mathbf{V}(t)^\top.$$

We now assume that we are given a piecewise smooth curve $\gamma : I \to \mathcal{M}_{n,p}$, where $I \subset \mathbb{R}$ is an interval. This curve induces three piecewise smooth families

$$\gamma_U : t \mapsto \mathbf{U}(t) \in \mathrm{St}(n,p),\ \gamma_V : t \mapsto \mathbf{V}(t) \in \mathrm{St}(n,p)$$

and

$$\gamma_S : t \mapsto \mathbf{S}(t) \in \mathbb{R}^{p\times p}.$$

The curve γ_S is simply a vector valued curve which can be further processed with standard data processing methods such as wavelets. For the curves $\gamma_U,\ \gamma_V$ this is not the case since they take values in a nonlinear space. When we wish to compress these curves it is important to respect the underlying nonlinear constraints so that the columns of the resulting matrices are maximally uncorrelated, ensuring minimal redundancy.

2.2.2 Metric Properties

It is easy to see that the set $\mathrm{St}(n,p)$ carries the structure of a smooth manifold by viewing it as the zero set of the submersion

$$F : \left\{ \begin{array}{ccc} \mathbb{R}^{n\times n} & \to & \mathbb{R}^{p\times p} \\ \mathbf{X} & \mapsto & \mathbf{X}^\top\mathbf{X} - \mathbf{I}_p \end{array} \right.$$

From this we also see that

$$\dim\left(\mathrm{St}(n,p)\right) = np - \frac{1}{2}p(p+1).$$

Furthermore, it is straightforward to obtain the following simple expression for the tangent space attached to a point $\mathbf{X} \in \mathrm{St}(n,p)$:

$$T_{\mathbf{X}}\mathrm{St}(n,p) = \left\{\xi \in \mathrm{R}^{n\times p} :\ \xi^\top\mathbf{X} + \mathbf{X}^\top\xi = \mathbf{0}_p\right\}.$$

The metric properties of the Stiefel manifold are slightly more complicated. In particular there exist two nonequivalent canonical choices for a metric on $\mathrm{St}(n,p)$. The first one views the manifold as a factor of the orthogonal group $\mathrm{O}(n)$:

$$\mathrm{St}(n,p) = \mathrm{O}(n)/\mathrm{O}(n-p),$$

by identifying the Stiefel manifold $\mathrm{St}(n,p)$ with the equivalence classes

$$[\mathbf{Q}] := \left\{\mathbf{Q}\begin{pmatrix} \mathbf{I}_p & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_{n-p} \end{pmatrix} :\ \mathbf{Q}_{n-p} \in \mathrm{O}(n-p)\right\},$$

$\mathbf{Q} \in \mathrm{O}(n)$. The tangent space of $\mathrm{St}(n,p)$ at $[\mathbf{Q}]$ is identified with the horizontal space at $\mathbf{Q}$ consisting of vectors

$$\mathbf{Q}\begin{pmatrix} \mu & -\eta^\top \\ \eta & \mathbf{O}_{n-p} \end{pmatrix}, \tag{12}$$

where $\mu \in \mathbb{R}^{p\times p}$ is skew symmetric, see [18]. The Riemannian metric is defined by restriction of the metric on $\mathrm{O}(n)$. The so-defined metric is sometimes called the *canonical metric*. Another approach defines the Riemannian metric by viewing $\mathrm{St}(n,p)$ as a Riemannian submanifold of $\mathbb{R}^{n\times p}$ with the metric coming from the inner product

$$\langle \xi, \eta\rangle := \mathrm{tr}\left(\xi^\top\eta\right).$$

We will mainly focus on the latter approach which we call the *embedded metric*.

2.2.3 Retraction Pairs

We now turn to the problem of defining retractions on the Stiefel manifold. A natural choice would be to take the exponential function. Of course, the functions exp and log depend on the Riemannian metric imposed on $\mathrm{St}(n,p)$.

Exponential function for canonical metric. For the canonical metric the exponential and logarithm functions can be easily defined using the matrix exponential

$$\exp(\xi) := \sum_{j\geq 0}\frac{\xi^j}{j!}, \quad \xi \in \mathfrak{sl}(n)$$

and its local inverse log. Then, for a vector $\xi \in T_{\mathbf{X}}\mathrm{St}(n,p)$, $\mathbf{X} = [\mathbf{Q}] \in \mathrm{St}(n,p)$, where ξ has a repesentation of the form (12) we can put

$$\exp_{\mathbf{X}}(\xi) := \left[\mathbf{Q} \exp\left(\begin{pmatrix} \mu & -\eta^\top \\ \eta & \mathbf{O}_{n-p} \end{pmatrix}\right)\right].$$

Algorithmically, these formulas require us to deal with $n \times n$ matrix operations which is clearly impractical if p is much smaller than n. There exist other expressions for the exponential function associated with the canonical metric which only require the handling of $n \times p$ matrices, see [18], but no simple expression for the logarithm function is known to us.

Exponential function for embedded metric. For the embedded metric, a formula for the computation of the exponential function is given by

$$\exp_{\mathbf{X}}(\xi) = (\mathbf{X}, \xi) \exp\begin{pmatrix} \mathbf{X}^\top\xi & -\xi^\top\xi \\ \mathbf{I}_p & \mathbf{X}^\top\xi \end{pmatrix} \begin{pmatrix} \mathbf{I}_p \\ \mathbf{0}_p \end{pmatrix} \exp\left(-\mathbf{X}^\top\xi\right),$$

see e.g. [18, 1]. Again, we are not aware of a computationally simple expression for the logarithm mapping.

Closest Point Projection.

Our requirement is that f and g should both be computable efficiently. Probably the most efficient choice for f is given by

$$f(\mathbf{X}, \xi) := \pi(\mathbf{X} + \xi),$$

where $\xi \in T_{\mathbf{X}}\mathrm{St}(n,p)$ and

$$\pi : \mathbb{R}^{n\times p} \to \mathrm{St}(n,p)$$

denotes the closest-point projection onto $\mathrm{St}(n,p)$ with respect to the metric induced by $\langle\cdot,\cdot\rangle$. This can be computed via a polar decomposition of a matrix $\mathbf{X} = \mathbf{RH}$ with $\mathbf{R} \in \mathrm{St}(n,p)$ and $\mathbf{H} \in \mathbb{R}^{p\times p}$ is symmetric. This decomposition is unique whenever $\mathbf{X}$ is nonsingular and can be computed efficiently [30, 25]. Algorithm 2 gives a quadratically convergent iterative scheme. Furthermore, for this choice of f, the equation (6) holds true.

Algorithm 2: Calculate $\mathbf{Y} = \pi(\mathbf{Z})$, the closest point projection of $\mathbf{Z} \in \mathbb{R}^{n\times p}$ onto $\mathrm{St}(n,p)$.

```
Y ← Z
while ‖Y^⊤Y − I_p‖ > TOL do
  Y ← Y (I_p + ½ (I_p − Y^⊤Y))
end while
return Y
```

We also need to find a simple expression for the inverse g of f. It turns out that such an expression exists.

Proposition 2.7. *Assume that* $\mathbf{X}^\top\mathbf{Y}$ *has only eigenvalues with positive real part (in other words,* $-\mathbf{X}^\top\mathbf{Y}$ *is a* Hadamard matrix*). Then we have*

$$g(\mathbf{X},\mathbf{Y}) = 2\mathbf{Y}\int_0^\infty \exp(-t\mathbf{X}^\top\mathbf{Y})\exp(-t\mathbf{Y}^\top\mathbf{X})\mathrm{d}t - \mathbf{X}$$

or

$$\pi\left(2\mathbf{Y}\int_0^\infty \exp(-t\mathbf{X}^\top\mathbf{Y})\exp(-t\mathbf{Y}^\top\mathbf{X})\mathrm{d}t\right) = \mathbf{Y}.$$

Before we get to the proof we collect some further auxiliary results regarding the embedded geometry of the Stiefel manifold. For a general matrix $\mathbf{Z}\in\mathbb{R}^{n\times p}$ and $\mathbf{X}\in \mathrm{St}(n,p)$, the orthogonal projection $\pi_{T_\mathbf{X}\mathrm{St}(n,p)}$ onto $T_\mathbf{X}\mathrm{St}(n,p)$ is given by

$$\pi_{T_\mathbf{X}\mathrm{St}(n,p)}(\mathbf{Z}) = \mathbf{X} + \mathbf{Z} - \mathbf{X}\mathrm{sym}\left(\mathbf{X}^\top\mathbf{Z}\right), \tag{13}$$

where

$$\mathrm{sym}(\mathbf{A}) := \frac{1}{2}(\mathbf{A}+\mathbf{A}^\top)$$

denotes the symmetric part of a matrix. We also need a description of the normal space $N_\mathbf{X}\mathrm{St}(n,p)$ of $\mathrm{St}(n,p)$ at a point $\mathbf{X}\in\mathrm{St}(n,p)$. We have

$$N_\mathbf{X}\mathrm{St}(n,p) = \left\{\mathbf{XS}:\ \mathbf{S}\in\mathbb{R}^{p\times p},\ \mathbf{S}\text{ symmetric}\right\}$$

and the orthogonal projection onto the normal space is given by

$$\pi_{N_\mathbf{X}\mathrm{St}(n,p)}(\mathbf{Z}) = \mathbf{Y}\mathrm{sym}\left(\mathbf{Y}^\top\mathbf{Z}\right). \tag{14}$$

With these preparations we can now proceed with the proof of the representation of g.

Proof of Proposition 2.7. Clearly $g(\mathbf{X},\mathbf{Y})$ is uniquely defined as the matrix ξ which lies both in the tangent space at $\mathbf{X}$ and the normal space of $\mathbf{Y}$. Therefore, we must have that

$$\xi = \mathbf{YS},\quad \mathbf{S}\in\mathbb{R}^{p\times p},\ \mathbf{S}\text{ symmetric},$$

such that

$$\mathbf{X}^\top(\xi-\mathbf{X}) + (\xi-\mathbf{X})^\top\mathbf{X} = \mathbf{0}_p.$$

Therefore we have

$$\mathbf{X}^\top(\mathbf{YS}-\mathbf{X}) + (\mathbf{YS}-\mathbf{X})^\top\mathbf{X} = \mathbf{0}_p.$$

Now we use that $\mathbf{X}^\top\mathbf{X} = \mathbf{I}_p$ and obtain

$$\mathbf{X}^\top\mathbf{YS} + \mathbf{SY}^\top\mathbf{X} - 2\mathbf{I}_p = \mathbf{0}_p. \tag{15}$$

Equations of the form (15) are well-known and frequently studied in the area of control theory, and it turns out that an explicit solution formula for $\mathbf{S}$ symmetric is given by the expression of our proposition, see [36]. □

In practice we will not use Proposition 2.7 directly, but compute $g(\mathbf{X},\mathbf{Y})$ via the iterative procedure outlined in Algorithm 3:

Using (14) and (13), this iteration can be computed very efficiently.

Algorithm 3: Calculate $g(\mathbf{X}, \mathbf{Y})$ for f the closest point projection onto $\mathrm{St}(n,p)$.

$\mathbf{Y} \leftarrow \mathbf{X}_1$
while $\max\left(\left\|\mathbf{Y} - \pi_{N_{\mathbf{X}}\mathrm{St}(n,p)}(\mathbf{Y})\right\|, \left\|\mathbf{Y} - \pi_{T_{\mathbf{X}}\mathrm{St}(n,p)}(\mathbf{Y})\right\|\right) > \mathrm{TOL}$ **do**
  $\mathbf{Y} \leftarrow \pi_{N_{\mathbf{X}}\mathrm{St}(n,p)}(\mathbf{Y})$
  $\mathbf{Y} \leftarrow \pi_{T_{\mathbf{X}}\mathrm{St}(n,p)}(\mathbf{Y})$
end while
return $\mathbf{Y}$

Lemma 2.8. *The iteration in Algorithm 3 converges towards* $g(\mathbf{X}, \mathbf{Y})$.

Proof. The iterative procedure in Algorithm 3 is an instance of a very common way to iteratively compute the intersection of two convex sets (in our case subspaces) by alternating projections. A proof of convergence of these algorithms can be found in [3]. □

Again, in our experiments we have found that the iteration in Algorithm 3 typically requires $2-3$ iterations until it reaches machine precision.

Vertical Retraction. For the construction of the pair (f, g) we started with the choice of f and found its inverse g. Another approach to construct a retraction is to define the function g first and then find out whether the corresponding f is efficiently computable. A natural choice is to put

$$\tilde{g}(\mathbf{X}, \mathbf{Y}) := \pi_{T_{\mathbf{X}}\mathrm{St}(n,p)}(\mathbf{Y}) = \mathbf{X} + \mathbf{Y} - \mathbf{X}\mathrm{sym}\left(\mathbf{X}^\top \mathbf{Y}\right), \tag{16}$$

the orthogonal projection onto the tangent space in $\mathbf{X}$. In contrast to the previous example, it is now not so clear how to compute the corresponding retraction $\tilde{f}$. The next result, which can be found in [24, Chapter IV.9], shows how to easily compute it. For the convenience of the reader we also provide a short proof.

Proposition 2.9. *The function* $\tilde{f}$*, satisfying*

$$\tilde{f}(\mathbf{X}, \tilde{g}(\mathbf{X}, \mathbf{Y})) = \mathbf{Y} \quad \text{for all } \mathbf{Y} \in \mathrm{St}(n,p) \tag{17}$$

and $\tilde{g}$ *as in (16) can be taken to be* $\tilde{f}(\mathbf{X}, \xi) := \tilde{\pi}_{\mathbf{X}}(\mathbf{X} + \xi)$*, where*

$$\tilde{\pi}_{\mathbf{X}}(\mathbf{Y}) := \mathbf{Y} + \mathbf{X}\mathbf{S}, \tag{18}$$

where $\mathbf{S} \in \mathbb{R}^{p\times p}$ *is symmetric and satisfies the* algebraic Riccati equation

$$\mathbf{Y}^\top\mathbf{Y} + \mathbf{S}\mathbf{Y}^\top\mathbf{X} + \mathbf{Y}^\top\mathbf{X}\mathbf{S} + \mathbf{S}^2 - \mathbf{I}_p = \mathbf{0}_p. \tag{19}$$

Proof. First note that, if we want (17) to hold, then certainly

$$\tilde{f}(\mathbf{X}, \xi) = \mathbf{X} + \xi + \mathbf{X}\mathbf{S}$$

for some symmetric matrix $\mathbf{S} \in \mathbb{R}^{p\times p}$. Therefore, since also

$$\tilde{f}(\mathbf{X}, \xi)^\top \tilde{f}(\mathbf{X}, \xi) = \mathbf{I}_p,$$

we get the equation (19). □

We would like to remark that the Equation (19) can be solved efficiently, see for instance [34].

We briefly summarize what we have achieved so far. For the Stiefel manifold we have constructed efficiently computable pairs (f, g) of functions satisfying (4), (5) and (6). These will be crucial in our later construction of a multiscale decomposition of Stiefel-valued curves.

Using the above defined retraction pairs we can now define a geometric averaging operation on $\mathrm{St}(n,p)$ which satisfies certain natural invariance properties.

There exists a natural transitive action τ of the group $\mathrm{O}(n)$ on $\mathrm{St}(n,p)$ via

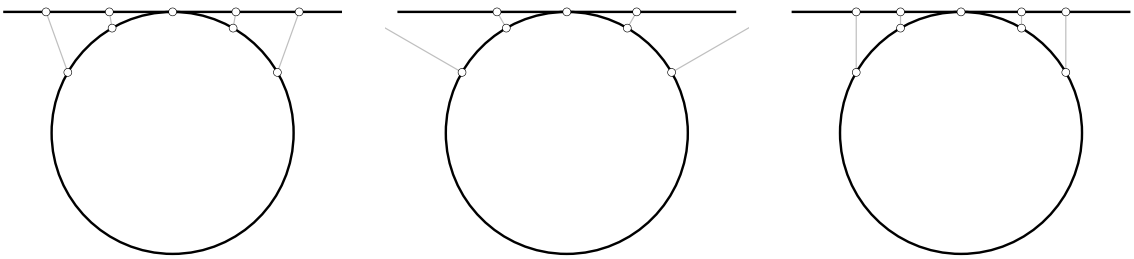
$$\tau_{\mathbf{Q}}(\mathbf{X}) := \mathbf{Q}\mathbf{X}, \quad \mathbf{Q} \in \mathrm{O}(n).$$

Lemma 2.10. *For all functions g considered above, the average is invariant under the action τ: we have*

$$av_g\left((w_i), (\tau_{\mathbf{Q}}(\mathbf{X}_i))\right) = \tau_{\mathbf{Q}}\left(av_g\left((w_i), (\mathbf{X}_i)\right)\right).$$

Proof. This follows from the fact that the functions g are always invariant under the action of τ. □

Lemma 2.10 shows that the geometric averaging operation defined in this section is invariant under an orthogonal change of coordinates, as it should be. Figure 1 shows the different choices of retractions for the circle which is equal to $\mathrm{St}(2,1)$.



(a) Retraction pair based on exponential map

(b) Retraction pair based on closest point projection

(c) Retraction pair based on vertical projection

Figure 1: Different retractions for the circle.

Remark 2.11. *In order to compute a weighted average of points $(\mathbf{X}_i)_{i=1}^d$ a minimal requirement is that there exists a point $\mathbf{X}_*$ such that all operations $g(\mathbf{X}_*, \mathbf{X}_i)$, $i = 1, \ldots, d$ are well-defined. In this respect, the inverse of the closest-point projection appears to be the worst choice, see Figure 1. If this requirement is not fulfilled, some ad-hoc strategies seem necessary. In our experiments we simply compute the linear average of the points $(\mathbf{X}_i)_{i=1}^d$ and project the result back onto the manifold.*

3 Wavelet-Like Decompositions of Manifold-Valued Data

In this section we describe the wavelet-type decomposition of manifold valued data, using a retraction pair of functions (f, g) as in the previous section. All results of the present section apply to general manifolds M, therefore, for now, we shall consider a manifold M, together with a pair of functions (f, g) such that (4), (6) and (5) hold. Furthermore, we assume that an averaging operation defined by (8) is well-defined.

For instance in the general case of a Riemannian manifold M and $(f, g) = (\exp, \log)$, where exp and log denote the exponential resp. the logarithm function on M, it can be shown that all these assumptions are valid. Another example is, of course, M the Stiefel manifold and the functions (f, g) defined in the previous section.

Up to minor variations, the material in Section 3.1 and Section 3.2 is taken from [35]. The main difference is that we allow for more general retractions than the exponential function and that we utilize a different, in our view more natural notion of geometric average.

One of the main theoretical results of this paper is contained in Section 3.3 where it is shown that manifold valued wavelets are capable of approximating *piecewise* smooth functions with the same efficiency as globally smooth functions.

The general idea for the construction of manifold valued wavelet transforms is to start with a linear construction and replace all averaging operations by geometric averages and all difference operations by the point-point difference function g. For more information on wavelets in general we recommend [10, 8].

Convention: *Since our wavelet constructions will be based on geometric averages which in general are only locally defined we understand all the theoretical results in this section under the tacit assumption that all underlying operations are well-defined.*

3.1 Interpolating Wavelet Constructions

3.1.1 Linear Case

Here, we describe the construction of interpolating wavelets in manifolds using geometric averaging operations, see [35, 21, 16] for more information. The corresponding linear construction is due to Donoho [14]. We first describe this linear construction. The basic idea is that a smooth function $u : \mathbb{R} \to \mathbb{R}$ can be well approximated by a polynomial. We now define the sampling operator

$$u_h := (u(hi))_{i \in \mathbb{Z}}$$

and pick an odd integer D. We try to exploit possible redundancies between the samples $u_{2^{-j}}$ and $u_{2^{-(j+1)}}$ by computing a simple estimate

$$\hat{u}_{2^{-(j+1)}} \sim u_{2^{-(j+1)}} \tag{20}$$

based only on the knowledge of $u_{2^{-j}}$. We have already noted that, whenever u is smooth, it can be well approximated by a polynomial p of degree D. Therefore, it seems natural to define $\hat{u}_{2^{-(j+1)}}$ in such a way that $\hat{p}_{2^{-(j+1)}} = p_{2^{-(j+1)}}$ for all polynomials of degree D.

One can achieve this as follows: Denote $\pi_{k,j}$ the degree D polynomial which interpolates the data

$$\begin{pmatrix} 2^{-j}(k-\lfloor D/2\rfloor) & \dots & 2^{-j}k & \dots & 2^{-j}(k+\lfloor D/2\rfloor) \\ u\left(2^{-j}(k-\lfloor D/2\rfloor)\right) & \dots & u\left(2^{-j}k\right) & \dots & u\left(2^{-j}(k+\lfloor D/2\rfloor)\right) \end{pmatrix}.$$

Then we can put

$$\hat{u}_{2^{-(j+1)}}(k) := \pi_{k,j}\left(2^{-(j+1)}k\right),\ k \in \mathbb{Z}.$$

In general, the relation (20) will *not* be an equality, therefore we need to store the error

$$\alpha_j := (\alpha_j(k))_{k\in\mathbb{Z}} := \left(u\left(2^{-j}(k+1/2)\right) - \hat{u}_{2^{-(j+1)}}(2k+1)\right)_{k\in\mathbb{Z}}.$$

We have now transformed the sequence $u_{2^{-(j+1)}}$ into two sequences $(u_{2^{-j}}, \alpha_j)$, each of which is half the size of the original sequence. Figure 2 shows an illustration of this scheme. The point of this procedure is that, provided u is smooth, the coefficients of α_j measure the local polynomial approximation error at scale 2^{-j} which decays very fast. Therefore, most of the coefficients of α_j can be set to zero in this case. It is also clear how to invert the transformation by adding the coefficients of α_j to the polynomial imputation of the coarser sampling $f_{2^{-j}}$. We can iterate this construction and arrive at a decomposition

$$u_{2^{-J}} \rightsquigarrow (u_{2^{-J_0}}, \alpha_{J_0+1}, \dots, \alpha_J)$$

with a minimal scale J_0. Here is a fundamental result regarding the decay properties of the coefficients α_j:

Theorem 3.1. *Assume that $u \in C^s$ with $s < D$. Then we have*

$$\|\alpha_j\|_\infty \lesssim 2^{-sj} \quad \text{for } j \in \mathbb{N}.$$

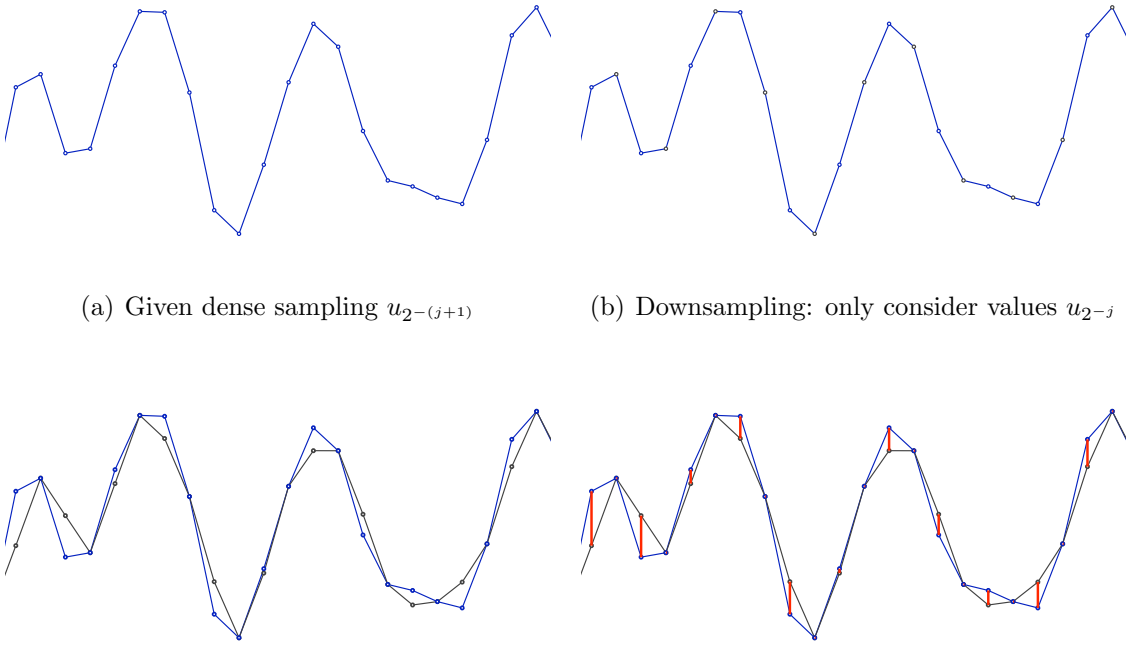
Several signals which appear in practice are not globally smooth but only *piecewise smooth*. Due to the local nature of the wavelet construction, also piecewise smooth functions can be approximated efficiently.

Theorem 3.2. *Assume that u is supported on $[-1,1]$ and that $u\big|_{[-1,0)},\ u\big|_{[0,1]} \in C^s$ with a possible discontinuity of u in 0 and $s < D$. Then, by keeping only $N\log(N)$ wavelet coefficients, one can get an approximation u^N of u such that*

$$\|u - u^N\|_2 \le N^{-s}.$$

Proof. We only sketch the proof. Assume w.l.o.g. that $N = 2^j$. Then we can set all wavelet coefficients of scale $> j$ to zero except for those which contribute to the singularity. For any fixed scale J, there is a finite number of such coefficients. We also keep these coefficients up to scale $2sj$. Thus, all in all we keep order $N\log(N)$ coefficients, and all the others are set to zero. By Theorem 3.1, this approximation gives an error of order N^{-s} in the L_∞ norm (and thus also in the L_2-norm) away from an interval I of size $\sim 2^{-2sj}$ around the singularity. It remains to estimate the error on the interval I Here, the L_2-error is also of order N^{-s}, due to the small size of the interval. □

Theorem 3.2 is of tremendous importance. It states that a piecewise smooth signal can be encoded up to an error N^{-s} by using about $N\log(N)$ coefficients. This is, up to a logarithmic factor which is usually considered negligible, as good as if the singularity were not present at all! Also, it can be shown that this result is optimal in the sense that essentially no better compression strategies for piecewise smooth functions exist [17].



(a) Given dense sampling $u_{2^{-(j+1)}}$

(b) Downsampling: only consider values $u_{2^{-j}}$

(c) Compute prediction $\hat{u}_{2^{-(j+1)}}$ via Deslauriers-Dubuc subdivision

(d) Keep prediction errors as wavelet coefficients

Figure 2: Illustration of interpolating wavelet scheme.

3.1.2 Nonlinear Case

Our first goal is to extend the constructions and results of Section 3.1.1 to the manifold-valued setting. To some extent this has been done in [16, 35, 21]. Our construction is somewhat different from the one in [35] in that we use another notion of geometric mean, namely the one developed in Section 2.1. This is a minor difference for the nonlinear construction of interpolating wavelets, but it *will* make a difference for the construction of nonlinear average interpolating wavelet transforms which we study below in Section 3.2.

The starting point is the fact that the mapping

$$S : u_{2^{-j}} \mapsto \hat{u}_{2^{-(j+1)}}$$

has a very simple structure:

$$Su_{2^{-j}}(2k) = u_{2^{-j}}(k), \quad Su_{2^{-j}}(2k+1) = \sum_{i=-\lfloor D/2 \rfloor}^{\lfloor D/2 \rfloor} l^D(i) u_{2^{-j}}(k-i), \tag{21}$$

with some sequence l satisfying

$$\sum_{i=-\lfloor D/2 \rfloor}^{\lfloor D/2 \rfloor} l^D(i) = 1. \tag{22}$$

In the literature, the operator S is usually called the *Deslauriers-Dubuc subdivision scheme* [11]. It is now immediate to generalize the prediction operator to the nonlinear setting. Assume we have a pair (f, g) satisfying (4), (5) and (6). Then by Proposition 2.3, the operator S can also be defined in the M-valued setting: For a function $u : \mathbb{R} \to M$ we again use the notation u_h^M to denote uniform sampling of meshwidth $h > 0$. Then we can define

$$\hat{u}^M_{2^{-(j+1)}}(2k) := u^M_{2^{-j}}(k), \quad \hat{u}^M_{2^{-(j+1)}}(2k+1) := \mathrm{av}_g\left((l^D(i))_{i=-\lfloor D/2\rfloor}^{\lfloor D/2\rfloor}, \left(u^M_{2^{-j}}(k-i)\right)_{i=-\lfloor D/2\rfloor}^{\lfloor D/2\rfloor}\right). \tag{23}$$

Similar to the linear case, the wavelet coefficients α_j^M are defined as the prediction error arising from this procedure. However, in contrast to the linear case we cannot simply subtract two points. Nevertheless the function g gives us a notion of difference vector between points. Using this notion we can define the wavelet coefficients as follows:

$$\alpha_j^M(k) := g\left(\hat{u}^M_{2^{-(j+1)}}(2k+1), u^M_{2^{-(j+1)}}(2k+1)\right) \in T_{\hat{u}^M_{2^{-(j+1)}}(2k+1)}M, \ k \in \mathbb{Z}.$$

For a function $u : \mathbb{R} \to M$ we get a decomposition

$$u^M_{2^{-J}} \rightsquigarrow \left(u^M_{2^{-J_0}}, \alpha^M_{J_0}, \ldots, \alpha^M_{J-1}\right)$$

The reconstruction procedure is obvious, for instance to go from $(u^M_{2^{-j}}, \alpha^M_{j+1})$ to $u^M_{2^{-(j+1)}}$ we first compute $\hat{u}^M_{2^{-(j+1)}}$ from $u^M_{2^{-j}}$ using (23). Then we have

$$u^M_{2^{-(j+1)}}(2k) = u^M_{2^{-j}}(k), \quad u^M_{2^{-(j+1)}}(2k+1) = f\left(\hat{u}^M_{2^{-(j+1)}}(2k+1), \alpha^M_{j+1}(k)\right). \tag{24}$$

Observe however that the reconstruction procedure constitutes a nonlinear operation in each step which makes it much more difficult to analyze theoretically.

3.2 Average Interpolating Wavelets

If we are dealing with data which is free of noise, the interpolating wavelet transform is an appropriate choice. However, in practice one often has to work with noisy data which, due to the instability of point evaluations, presents difficulties for the interpolating transform [35]. A remedy to this problem is to work not with point values but with averages of the function over dyadic boxes. In [15] such constructions are given. We briefly describe the construction which is a special case of the biorthogonal wavelet transform framework introduced in [9].

3.2.1 Linear Case

The idea is to discretize not by point evaluation but by integration over dyadic boxes. The resulting discretization operator for a function $u : \mathbb{R} \to \mathbb{R}$ is defined via

$$\bar{u}_h = \left(\int_{[hk, h(k+1)]} u(t)\mathrm{d}t\right)_{k\in\mathbb{Z}}.$$

Again we need an operator to get from $\bar{u}_{2^{-j}}$ to a prediction $\hat{\bar{u}}_{2^{-(j+1)}} \sim \bar{u}_{2^{-(j+1)}}$. This is done in a similar way as for the interpolating wavelet transform by picking for each k a

polynomial $\bar{\pi}_{k,j}$ of degree D, D even, which interpolates the averages of f over dyadic boxes of size 2^{-j} around k. Then we define

$$\hat{\bar{u}}_{2^{-(j+1)}}(2k) = \int_{[2^{-j}k, 2^{-j}(k+1/2)]} \bar{\pi}_{j,k}(t)\mathrm{d}t \text{ and } \hat{\bar{u}}_{2^{-(j+1)}}(2k+1) = \int_{[2^{-j}(k+1/2), 2^{-j}(k+1)]} \bar{\pi}_{j,k}(t)\mathrm{d}t.$$

Observe the redundancies

$$\frac{1}{2}\left(\hat{\bar{u}}_{2^{-(j+1)}}(2k) + \hat{\bar{u}}_{2^{-(j+1)}}(2k+1)\right) = \bar{u}_{2^{-j}}(k) \text{ for all } k \in \mathbb{Z}. \tag{25}$$

and

$$\frac{1}{2}\left(\bar{u}_{2^{-(j+1)}}(2k) + \bar{u}_{2^{-(j+1)}}(2k+1)\right) = \bar{u}_{2^{-j}}(k) \text{ for all } k \in \mathbb{Z}. \tag{26}$$

Again we can define wavelet coefficients

$$\beta_j := (\alpha_j(k))_{k\in\mathbb{Z}} := \left(\bar{u}_{2^{-(j+1)}}(2k+1) - \hat{\bar{u}}_{2^{-(j+1)}}(2k+1)\right)_{k\in\mathbb{Z}}.$$

Using the redundancies (25), (26) we can reconstruct the values of $\bar{u}_{2^{-(j+1)}}$ from $\bar{u}_{2^{-j}}$ and α_j via

$$\bar{u}_{2^{-(j+1)}}(2k+1) = \hat{\bar{u}}_{2^{-(j+1)}}(2k+1) + \beta_j(k),\ \bar{u}_{2^{-(j+1)}}(2k) = \bar{u}_{2^{-j}}(k) - \left(\bar{u}_{2^{-(j+1)}}(2k+1) - \bar{u}_{2^{-j}}(k)\right),$$

where $k \in \mathbb{Z}$. Note that also for this wavelet transform Theorems 3.1 and 3.2 hold true [15].

3.2.2 Nonlinear Case

Average interpolating wavelet transforms can also be defined in the manifold-valued context, at least to some extent. Several problems arise, mostly due to the fact that it is not clear how to define an averaging operation of an N-valued function over a dyadic box. Certainly, such operations exist in a geometric setting, see [27], but the problem with these constructions is that in general the redundancy relation (26) fails.

At least for practical purposes a partial remedy is to start from a *midpoint pyramid*

$$\bar{u}_J^M, \bar{u}_{J-1}^M, \ldots, \bar{u}_{J_0}^M \tag{27}$$

satisfying the *consistency condition*

$$\bar{u}_j^M(k) = \mathrm{av}_g\left(\left(\frac{1}{2}, \frac{1}{2}\right), \left(\bar{u}_{j+1}^M(2k), \bar{u}_{j+1}^M(2k+1)\right)\right) \tag{28}$$

for a minimal scale J_0 and a maximal scale J and $J_0 \le j < J$. In practice, one is given a sequence of data $\bar{u}_J^M$, say of length 2^J. Then one can define a midpoint pyramid (27) by taking (28) as a definition.

Next we note the fact that in the linear case the prediction operator which maps $\bar{u}_{2^{-j}}$ to $\hat{\bar{u}}_{2^{-(j+1)}}$ can be written as

$$\hat{\bar{u}}_{2^{-(j+1)}}(2k+1) = \sum_{i=-D/2+1}^{D/2} r^D(i)\bar{u}_{2^{-j}}(k-i) \tag{29}$$

for some sequence r^D satisfying

$$\sum_{i=-D/2+1}^{D/2} r^D(i) = 1.$$

Motivated by (29) and (25) we put

$$\hat{\bar{u}}^M_{j+1}(2k+1) = \mathrm{av}_g\left(\left(r^D(i)\right)_{i=-D/2+1}^{D/2}, \left(\bar{u}^M_j(k-i)\right)_{i=-D/2+1}^{D/2}\right) \tag{30}$$

and

$$\hat{\bar{u}}^M_{j+1}(2k) = f\left(\bar{u}^M_j(k), -g\left(\bar{u}^M_j(k), \hat{\bar{u}}^M_{j+1}(2k+1)\right)\right). \tag{31}$$

By (31) we get the redundancy relation

$$\bar{u}^M_j(k) = \mathrm{av}_g\left(\left(\frac{1}{2}, \frac{1}{2}\right), \left(\hat{\bar{u}}^M_{j+1}(2k), \hat{\bar{u}}^M_{j+1}(2k+1)\right)\right). \tag{32}$$

One should compare the relations (32) and (28) with the linear relations (25) and (26).

We can now define wavelet coefficients

$$\beta^M_j(k) := g\left(\hat{\bar{u}}^M_{j+1}(2k+1), \bar{u}^M_{j+1}(2k+1)\right) \in T_{\hat{\bar{u}}^M_{j+1}(2k+1)}M,\ k \in \mathbb{Z}.$$

The reconstruction procedure proceeds similarly to the linear one. Given $(\bar{u}^M_j, \beta^M_j)$ we first compute $\hat{\bar{u}}^M_{j+1}$ using (30) and (31). Then we get the reconstruction of the odd indices via

$$\bar{u}^M_{j+1}(2k+1) = f\left(\bar{u}^M_j(k), \beta^M_j(k)\right).$$

To reconstruct the even indices we make use of (28) and (32) to find that

$$\bar{u}^M_{j+1}(2k) = f\left(\bar{u}^M_j(k), -g\left(\bar{u}^M_j(k), \bar{u}^M_{j+1}(2k+1)\right)\right).$$

We have constructed a way to transform a midpoint pyramid into a wavelet decomposition

$$(\bar{u}^M_{J_0}, \beta^M_{J_0}, \ldots, \beta^M_{J-1})$$

and vice versa.

Remark 3.3. *Our construction easily extends to the regular multivariate case by defining the notion of midpoint pyramid as a uniformly weighted average of four points. Note that in [35] a different generalization to the multivariate case is given which e.g. in the bivariate setting first applies a wavelet transform in the vertical direction and then in the horizontal direction. Doing this one runs into trouble when computing the wavelet coefficients corresponding to the diagonal direction which requires the computation of a univariate wavelet transform of tangent vectors. We think that our approach of computing two dimensional transforms using geometric averages is more natural.*

3.3 Approximation Properties

We now turn to the theoretical properties of the manifold-valued wavelet constructions. The average interpolating construction described in Section 3.2.2 starts from a midpoint pyramid, which in practice can be constructed from a discrete sequence of data via (28) . However, it is not clear how to associate to a given function $u : \mathbb{R} \to M$ an infinite midpoint pyramid satisfying the downsampling relation (28), or a similar one. This fact makes an asymptotic analysis of the decay of the wavelet coefficients impossible. For this reason we focus on the interpolating wavelet transform for manifold-valued data described in Section 3.1.2.

In the remainder of this section we once and for all fix an odd integer $D > 0$ and associate to an arbitrary function $u : \mathbb{R} \to M$ its nonlinear interpolating wavelet transform

$$\left(u^M_{2^{-J_0}}, \alpha^M_{J_0}, \dots\right)$$

as in Section 3.1.2.

Before we formulate our main results we need to introduce some notation.

For a sequence α of vectors $(\alpha(k))_{k\in\mathbb{Z}}$ we use the symbol

$$\|\alpha\|_\infty = \sup_{k\in\mathbb{Z}} \|\alpha(k)\|_M,$$

where $\|\cdot\|_M$ indicates the Riemannian metric on M. Further, we want to be able to compare two sequences $u = (u(k))_{k\in\mathbb{Z}}$, $v = (v(k))_{k\in\mathbb{Z}}$. Here we use the notation

$$\operatorname{dist}_M(u, v)_\infty := \sup_{k\in\mathbb{Z}} \operatorname{dist}_M(u(k), v(k)),$$

where dist_M denotes the geodesic distance in M. Finally, with α as above and β of the same form, we also want to compare these two sequences. Denoting for two vectors $\gamma \in T_xM$, $\delta \in T_yM$

$$\operatorname{dist}_{TM}(\gamma, \delta) := \operatorname{dist}_M(x, y) + \|\gamma - \operatorname{Pt}^x_y(\delta)\|_M,$$

Pt^x_y denoting parallel transport from y to x [13] we write

$$\operatorname{dist}_{TM}(\alpha, \beta)_\infty := \sup_{k\in\mathbb{Z}} \operatorname{dist}_{TM}(\alpha(k), \beta(k)).$$

Then the following analog of Theorem 3.1 holds:

Theorem 3.4. *Assume that $u \in C^s$ for $s < D$. Then we get the following decay rate for the wavelet coefficients:*

$$\|\alpha^M_j\|_\infty \lesssim 2^{-sj} \tag{33}$$

with the implied constant uniform on compact sets. Conversely, if (33) holds for the wavelet decomposition of u, then for all $\varepsilon > 0$ we have $u \in C^{s-\varepsilon}$.

Proof. This has been proven in [21]. □

Theorem 3.4 tells us something about the decay of the wavelet coefficients associated with a smooth function. The potential use of this result is clear: by setting all small coefficients to zero, one should expect to still obtain a very good approximation of the original function. This intuition turns out to be correct, but it is far from trivial in the nonlinear case. The key issue lies in the stability properties of the reconstruction procedure. More specifically, one needs to address the question of how a change of the wavelet coefficients affects the reconstruction. To this end we consider the reconstruction operator

$$R_j : \left(u^M_{2^{-J_0}}, \alpha^M_{J_0}, \dots, \alpha^M_{j-1}\right) \mapsto u^M_{2^{-j}}$$

using (24).

The following result holds.

Theorem 3.5. *Assume that (f, g) satisfy (6), (5) and (4). Assume we are given two wavelet decompositions*

$$\mathfrak{U} := \left(u^M_{2^{-J_0}}, \alpha^M_{J_0}, \dots, \alpha^M_{j-1}\right)$$

and

$$\mathfrak{V} := \left(v^M_{2^{-J_0}}, \beta^M_{J_0}, \dots, \beta^M_{j-1}\right)$$

Then there exists a constant C independent of j such that

$$\mathrm{dist}_M\left(R_j(\mathfrak{U}), R_j(\mathfrak{V})\right)_\infty \leq C\left(\mathrm{dist}_M\left(u_{2^{-J_0}}, v_{2^{-J_0}}\right)_\infty + \sum_{i=J_0}^{j-1} \mathrm{dist}_{TM}\left(\alpha^M_i, \beta^M_i\right)\right).$$

Proof. This has been shown in [20]. □

Using the above stability result, we can now prove a nonlinear analog of Theorem 3.2. In order to avoid technical complications we assume that $u : [-1, 1) \to M$ is periodic.

Theorem 3.6. *Assume that $u : [-1, 1) \to M$ is a periodic C^s function ($s < D$) apart from a discontinuity at 0. Then, using $N \log(N)$ wavelet coefficients, one can obtain an approximation u_N such that*

$$\int_{[-1,1]} dist_M\left(u(t), u_N(t)\right)^2 \mathrm{d}t \lesssim N^{-2s}.$$

Proof. Using Theorem 3.5, the proof proceeds with similar arguments as the proof of the linear version, Theorem 3.2. Without loss of generality, we put $N = 2^j$. Let

$$\left(u^M_{2^{-J_0}}, \alpha^M_{J_0}, \dots\right)$$

be the wavelet decomposition of f. Assume we have the full data $u_{2^{-2sj}}$ at hand. We first examine what happens if we build an approximation $\tilde{u}$ of u from $u_{2^{-2sj}}$ by setting all wavelet coefficients of higher scale to zero. By Theorem 3.5, Theorem 3.4 and the locality of the wavelet reconstruction, we observe that, away from an interval I of size $\sim 2^{-2sj}$ around 0, we get an approximation error

$$\sup_{t \in [-1,1) \setminus I} \mathrm{dist}(u(t), \tilde{u}(t)) \lesssim 2^{-2s^2 j}.$$

See also [39, 19] for more details. Therefore we get that

$$\int_{[-1,1)\setminus I} \operatorname{dist}_M\left(u(t),\tilde{u}(t)\right)^2 \mathrm{d}t \lesssim 2^{-4s^2 j}. \tag{34}$$

On the other hand, the error

$$\operatorname{dist}_M\left(u(t),\tilde{u}(t)\right)$$

is bounded on I. Therefore we have

$$\int_I \operatorname{dist}_M\left(u(t),\tilde{u}(t)\right)^2 \mathrm{d}t \lesssim \int_I 1 \mathrm{d}t \lesssim 2^{-2sj}. \tag{35}$$

Putting together (34) and (35) we arrive at the estimate

$$\int_{[-1,1)} \operatorname{dist}_M\left(u(t),\tilde{u}(t)\right)^2 \mathrm{d}t \lesssim 2^{-2sj}. \tag{36}$$

Equation (36) gives the right magnitude of approximation error but clearly way too many coefficients are needed for $\tilde{u}$, which requires order 2^{2sj} nonzero wavelet coefficients. We now perform a further compression step, reducing the number of nonzero wavelet coefficients to order $j2^j$. To do this, we observe that by Theorem 3.5 it would be sufficient to threshold the wavelet coefficients

$$\mathfrak{U} := \left(u^M_{2^{-J_0}}, \alpha^M_{J_0}, \ldots, \alpha^M_{2sj-1}\right)$$

to

$$\mathfrak{V} := \left(u^M_{2^{-J_0}}, \beta^M_{J_0}, \ldots, \beta^M_{2sj-1}\right)$$

such that

$$\operatorname{dist}_M\left(R_{2sj}(\mathfrak{U}), R_{2sj}(\mathfrak{V})\right)_\infty \lesssim 2^{-sj}, \tag{37}$$

and such that the number of nonzero wavelet coefficients in $\mathfrak{V}$ is of order $N\log(N)$. To do this we note that at each scale k, only a fixed finite number $2L+1$ of wavelet coefficients

$$\left(\alpha^M_k(-L), \ldots, \alpha^M_k(L)\right)$$

gets affected by the singularity at 0. All the other wavelet coefficients

$$\alpha^{M,\mathrm{smooth}}_k := \left(\ldots, \alpha^M_k(-L-2), \alpha^M_k(-L-1), \alpha^M_k(L+1), \alpha^M_k(L+2), \ldots\right)$$

are only affected by the smooth part of u and therefore (by the locality of the construction and Theorem 3.4) decay of order 2^{-sk}. We now obtain our compression $\mathfrak{V}$ by setting to zero all coefficients $\alpha^{M,\mathrm{smooth}}_k$ for all scales $j, j+1, \ldots, 2sj$. Due to the decay properties of $\alpha^{M,\mathrm{smooth}}_k$ and Theorem 3.5, we get (37). It remains to count the number of nonzero coefficients in $\mathfrak{V}$. There are order 2^j coefficients up to scale j. For the scales $j+1, j+2, \ldots, 2sj$ we have $2L+1$ coefficients which amounts to a total of order $j2^j$ coefficients. This proves the theorem. □

Theorem 3.6 confirms that the nonlinear wavelet transform indeed satisfies the same important properties as its linear counterpart.

Remark 3.7. *We already mentioned that an asymptotic analysis of the nonlinear average interpolating wavelet construction is not possible. Nevertheless, one could still analyze the heuristic construction of Section 3.2.2 for finite data. We postpone this for future work. Note also that the possibilities for constructing more general wavelet transforms in manifolds are limited, cf. [22].*

4 Computational Experiments and Applications

We return to the specific case of the Stiefel manifold and present some computational experiments, starting first with synthetic data. For our experiments we have chosen the pair (f, g) constructed in 2.2.3 based on closest point projection. For the plots of the wavelet coefficients in Figures 3 and 4 we made use of the plotting routines of the Symmlab100 MATLAB package, available from `http://www.stanford.edu/~slansel/SymmLab/`. We are grateful to the authors of this package for making their software codes publicly available.

4.1 Synthetic Data

Our first experiment deals with a smooth matrix curve (and its svd)

$$\mathbf{A}(t) = \mathbf{U}(t)\mathbf{\Sigma}(t)\mathbf{V}(t)^\top = \sin(t\mathbf{E}) + \text{noise} \in \mathbb{R}^{10\times 10}, \tag{38}$$

with some random 10×10 matrix $\mathbf{E}$. We have used the first 3 columns of $\mathbf{U}(t)$ as a curve in $\mathrm{St}(10, 3)$ and performed a nonlinear average interpolating wavelet transform of this curve with $D = 4$. We performed a thresholding operation by setting all wavelet coefficients to zero which exceed the threshold of 0.01. The results can be seen in Figure 3. We can observe two things: First, the thresholding operation serves as a denoising operation. We remark that this is in general *not* the case with the interpolating wavelet transform. Second, if the noise level is set to zero, almost all wavelet coefficients are negligible, resulting in huge compression rates.

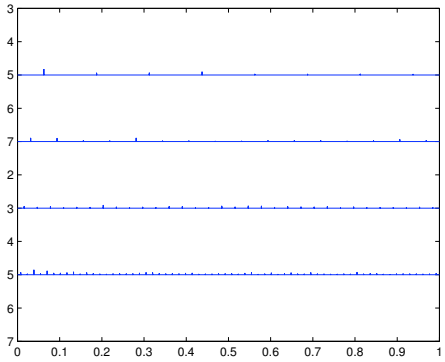
Our second experiment deals with a matrix curve which possesses a point singularity. We chose the curve

$$\mathbf{A}(t) = \mathbf{U}(t)\mathbf{\Sigma}(t)\mathbf{V}(t)^\top = \sin(t\mathbf{E}) + \left(1 - t^2\right)^{1/2} \cos(t\mathbf{F}) \in \mathbb{R}^{10\times 10}, \tag{39}$$

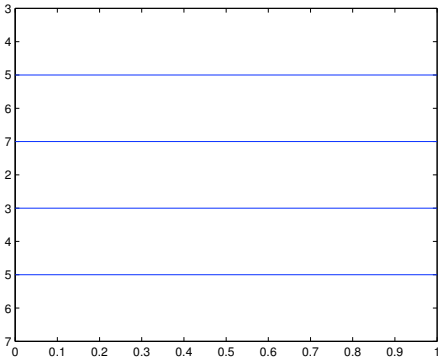
$\mathbf{E}$, $\mathbf{F}$ random 10×10 matrices, and again used the first 3 columns of $U(t)$ as a curve in $\mathrm{St}(10, 3)$. This time our analysis was performed using the interpolating wavelet transform with $D = 3$. The results are shown in Figure 4. We can nicely observe that all coefficients which do not correspond to the point singularity are almost negligible and only those around the singularity must be kept. The wavelet transform finds pointwise singularities by itself. On the right of Figure 4, a thresholding operation has been performed which results in overall savings of more than ninety percent.

4.2 Hyperspectral Image Compression

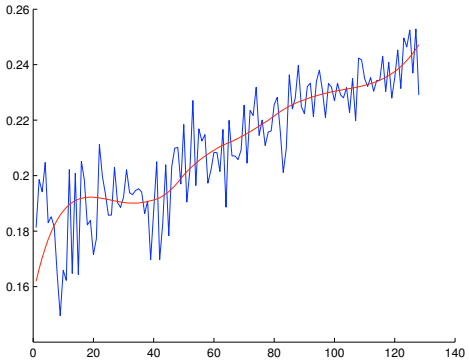
After some preliminary experiments on synthetic data in Section 4.1 we now apply our methods to a real problem in imaging science namely the compression of hyperspectral



(a) Stickplot of norms of wavelet coefficients of the first three columns of $\mathbf{U}(t)$ as defined in (38).



(b) Hard thresholding with threshold 0.01



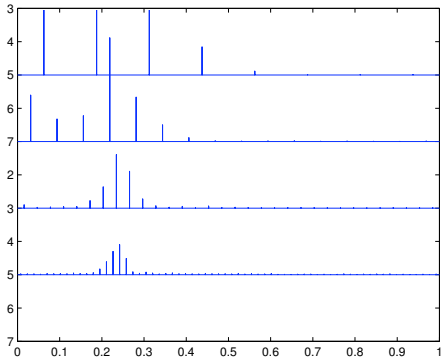
(c) Blue: Original curve of coefficient with index $(2, 2)$. Red: Reconstruction after thresholding.

Figure 3: Wavelet transform of (38).

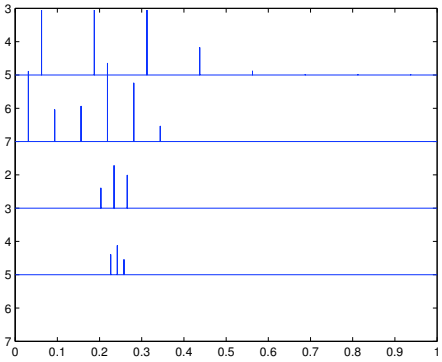
images. In our numerical experiments on real data we have always used the average interpolating construction from Section 3.2.2 due to its noise insensitivity. We stress once again that this is just one selected application, there are many other situations where smooth curves of low-rank matrices appear naturally, for instance latent semantic indexing, dynamic graph compression or linear operators depending on a parameter.

In our experiments we mainly focused on data compression. Since wavelet methods are capable of detecting large temporal variations in data, several other tasks such as feature detection, inpainting, classification, ... could be tackled. We postpone a more detailed study to future work. Before we describe our application we give a brief introduction to hyperspectral imaging.

A hyperspectral image is an image where intensities corresponding to many different wavelengths are recorded separately [5]. The result is a parameterized family $\mathbf{A}(\lambda) \in \mathbb{R}^{N\times N}$ of matrices, λ being the wavelength. We would like to emphasize that, when the wavelength λ varies continuously we get a smooth curve $\mathbf{A}(\lambda)$. In practice, only a finite number of wavelengths is recorded, typically at the order of hundreds or more. A related imaging technique is inferometry, where even more frequency bands are recorded. It is to



(a) Stickplot of norms of wavelet coefficients of the first three columns of $\mathbf{U}(t)$ as defined in (39).



(b) Hard thresholding with threshold 0.01

Figure 4: Wavelet transform of (39).

be expected that in the near future technology will advance to develop sensors capable of recording many more spectral bands, in the order of many thousand or even millions. This technique is called *ultraspectral imaging* [31]. For completeness we also mention the term *multispectral imaging* which refers to only a few frequency bands.

Due to its capacity to record information across the full range of the spectrum and recent technological advances, the scope of applicability of hyperspectral imaging is tremendous. We restrict ourselves to mentioning only a few important ones: In mineralogy, hyperspectral imaging is used for the detection of oil fields, while the US military uses this technique for the surveillance and detection of humans. A less controversial application is the monitoring of the constitution of plants and crops in agriculture.

Usually, the size of a hyperspectral image can be quite large which calls for efficient means to compress such data. Since a hyperspectral image can be interpreted as dense samples of a smooth matrix curve, it is reasonable to use a compression strategy which takes advantage of this fact.

In Algorithm 4 we propose a simple strategy for the compression of a hyperspectral image

$$\mathbf{A}_i \in \mathbb{R}^{N\times N},\ i = 1, \ldots, T$$

corresponding to recordings $\mathbf{A}(\lambda_i)$ at wavelength λ_i.

Algorithm 4: Compression scheme for hyperspectral image stack $\mathbf{A}_i \in \mathbb{R}^{N\times N},\ i = 1, \ldots, T$.

1. Decompose each matrix $\mathbf{A}_i$, $i = 1, \ldots, T$ into small blocks, say, of size 12×12.
2. Approximate each block by a rank 2 approximation for each $i = 1, \ldots, T$.
3. In each block perform wavelet compression of the rank 2 approximation. This amounts to wavelet thresholding for data with values in the Stiefel manifold.

Remark 4.1. *There is nothing exceptional about our choice of blocksize or the rank of the low-rank approximation in each block in Algorithm 4, any other choice would do. There*

is of course a lot of room for improvement; for instance the rank of the low-rank approximation could be chosen differently for each block by an adaptive procedure. We confine ourselves to our simple algorithm since the primary goal of this work is to demonstrate the usefulness of manifold-valued wavelet transforms, especially for the Stiefel manifold.

Remark 4.2. *Note that the more frequency bands are recorded, the better the performance of the wavelet compression is to be expected. This is because one needs at least 8-16 sampling points at the lowest scale (depending on D) in order to render the wavelet scheme well-defined. This limits the achievable compression rate. In our experiments, almost all wavelet coefficients could be discarded and we conjecture that it would be possible to achieve much higher compression rates for ultraspectral images with more frequency bands using our algorithm.*

Figure 5 shows an example of Algorithm 4 applied to real hyperspectral image data. The data is taken from `http://earthexplorer.usgs.gov/` and represents a sample of the Dehradun region in India, recorded with the Hyperion sensor aboard NASA's Earth Observation-1 (EO1) satellite. The sensor records wavelengths in the range of $355 - 2577nm$ at a sampling interval of about $10nm$. The dataset we used is an image cube of size $144 \times 144 \times 128$.

The resulting compression only requires about 7% of the original data, compared to about 30% by only using the low rank approximation without wavelet thresholding. In view of the present article, one should compare the difference between the low rank approximation and its wavelet compression, which is barely noticeable, see Figure 5. This represents a compression of the low-rank approximation by a factor about $1 : 5$, essentially without any noticeable degradation.

Remark 4.3. *So far we have not specified how we actually carry out Step 2 in Algorithm 4. We will comment on this issue later in Section 4.3 below. In addition we would like to point to the recent work [37] where sensing devices are constructed which directly sense a sparse representation without going through the usual procedure of first recording the full image and then throwing away almost all of the data. It seems conceivable that in the future sensors capable of directly recording a low rank approximation of an image can be constructed, although this is just speculation.*

Remark 4.4. *We reiterate that we do not claim that the compression scheme for hyperspectral images presented in this work is superior to state-of-the-art methods which are currently in use. The point is to demonstrate the usefulness of our flexible compression scheme for curves of low rank matrices. Of particular importance is the fact that our scheme* provably *achieves (asymptotically) optimal compression rates for piecewise smooth families of low-rank matrices. In fact, we are not aware of any other method in the literature which satisfies a similar property. Another distinctive property of the approach presented in this paper is that features such as point singularities can be detected automatically and non-adaptively simply by looking for the largest coefficients.*

4.3 Obtaining Smooth Low-Rank Approximations

In all our analysis the underlying assumption is that the curve of low-rank matrices is smooth, or at least piecewise smooth. This seems like a problem since usually low-rank

approximations are computed via svd which is nonunique, therefore one should not expect low-rank approximation curves computed via svd to be smooth.

Fortunately, a remedy exists. In [28] a method is given to obtain smooth dynamical low-rank approximations to matrix curves $\mathbf{A}(t) \in \mathbb{R}^{n\times n}$, $t > 0$.

The idea is to not compute a low-rank approximation for each matrix but to solve the ODE

$$\dot{\mathbf{X}}(t) = \min_{\xi \in T\mathcal{M}_{n,p}} \left\| \dot{\mathbf{A}}(t) - \xi \right\|_2, \quad \mathbf{X}(0) = \mathbf{X}_0, \tag{40}$$

where $\mathbf{X}_0$ is some initial rank p approximation to $\mathbf{A}(0)$. The computation of the solution of (40) can be done using geometric ODE integrators [26]. This is very efficient since typically the matrices $\dot{\mathbf{A}}(t)$ are much sparser than the matrices $\mathbf{A}(t)$. Moreover, in [28] it is shown that the approximation quality of $\mathbf{X}(t)$ is equivalent to the best possible approximation by a rank p matrix, at least as long as the first singular values stay well-separated. If singular values interlace, e.g. new singular vectors enter the best approximation, then the algorithm (40) needs to be restarted. By this procedure one obtains a piecewise smooth rank p approximation of $\mathbf{A}(t)$. In [33] several important applications of this algorithm are given, each one of them producing piecewise smooth curves on the Stiefel manifold which can be processed by the Stiefel-valued wavelet transform developed in this paper.

Acknowledgements

We thank Achim Nonnenmacher for sharing data related to [33] and Othmar Koch for sharing the MATLAB codes related to [28]. We also thank Johannes Wallner and Christian Lubich for their useful comments.

References

[1] P. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, 2008.

[2] M. Baumann and U. Helmke. Singular value decomposition of time-varying matrices. *Future Generation Computer Systems*, 19(3):353–361, 2003.

[3] L. Bregman. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR computational mathematics and mathematical physics*, 7(3):200–217, 1967.

[4] A. Bunse-Gerstner, R. Byers, V. Mehrmann, and N. Nichols. Numerical computation of an analytic singular value decomposition of a matrix valued function. *Numerische Mathematik*, 60(1):1–39, 1991.

[5] C. Chang. *Hyperspectral imaging: Techniques for spectral detection and classification*. Springer, 2003.

[6] J. Chern and L. Dieci. Smoothness and periodicity of some matrix decompositions. *SIAM Journal on Matrix Analysis and Applications*, 22:772–792, 2001.

[7] M. Chu, R. Funderlic, and R. Plemmons. Structured low rank approximation. *Linear algebra and its applications*, 366:157–172, 2003.

[8] C. Chui. *An Introduction to Wavelets*. Academic Press, 1992.

[9] A. Cohen, I. Daubechies, and J. Feauveau. Biorthogonal bases of compactly supported wavelets. *Communications on pure and applied mathematics*, 45(5):485–560, 1992.

[10] I. Daubechies. *Ten Lectures on Wavelets*. SIAM, 1992.

[11] G. Deslauriers and S. Dubuc. Symmetric iterative interpolation processes. *Constructive approximation*, 5(1):49–68, 1989.

[12] L. Dieci and T. Eirola. On smooth decompositions of matrices. *SIAM Journal on Matrix Analysis and Applications*, 20(3):800–819, 1999.

[13] M. DoCarmo. *Riemannian Geometry*. Birkhäuser, 1992.

[14] D. Donoho. Interpolating wavelet transforms. *Technical Report, Department of Statistics, Stanford University*, 1992.

[15] D. Donoho. Smooth wavelet decompositions with blocky coefficient kernels. *Recent Advances in Wavelet Analysis*, 3:259, 1994.

[16] D. Donoho. Wavelet-type representation of Lie-valued data. talk at the IMI "Approximation and Computation" meeting, May 12–17, 2001, Charleston, South Carolina, 2001.

[17] D. L. Donoho. Unconditional bases are optimal bases for data compression and for statistical estimation. *Applied and Computational Harmonic Analysis*, 1:100–115, 1993.

[18] A. Edelman, T. Arias, and S. Smith. The geometry of algorithms with orthogonality constraints. *SIAM Journal on Matrix Analysis and Applications*, 20(2):303–353, 1999.

[19] P. Grohs. Approximation order from stability of nonlinear subdivision schemes. *Journal of Approximation Theory*, 162:1085–1094, 2010.

[20] P. Grohs. Stability of manifold-valued subdivision schemes and multiscale transformations. *Constructive Approximation*, 32:569–596, 2010.

[21] P. Grohs and J. Wallner. Interpolatory wavelets for manifold-valued data. *Applied and Computational Harmonic Analysis*, 27(3):325–333, 2009.

[22] P. Grohs and J. Wallner. Definability and stability of multiscale decompositions for manifold-valued data. *Journal of the Franklin Institute*, 2011. in press.

[23] D. Groisser. Newton's method, zeroes of vector fields, and the Riemannian center of mass. *Advances in Applied Mathematics*, 33(1):95–135, 2004.

[24] E. Hairer, C. Lubich, and G. Wanner. *Geometric numerical integration: structure-preserving algorithms for ordinary differential equations*, volume 31. Springer Verlag, 2006.

[25] N. J. Higham. Matrix nearness problems and applications. In *Applications of Matrix Theory*, pages 1–27. University Press, 1989.

[26] A. Iserles, H. Munthe-Kaas, S. Nørsett, and A. Zanna. Lie-group methods. *Acta Numerica 2000*, 9(1):215–365, 2000.

[27] H. Karcher. Mollifier smoothing and Riemannian center of mass. *Communications on Pure and Applied Mathematics*, 30:509–541, 1977.

[28] O. Koch and C. Lubich. Dynamical low-rank approximation. *SIAM Journal on Matrix Analysis and Applications*, 29(2):434–454, 2008.

[29] O. Koch and C. Lubich. Dynamical tensor approximation. *SIAM Journal on Matrix Analysis and Applications*, 31:2360–2375, 2010.

[30] Z. Kovarik. Some iterative methods for improving orthonormality. *SIAM Journal on Numerical Analysis*, 7(3):386–389, 1970.

[31] A. Meigs, L. Otten, and T. Cherezova. Ultraspectral imaging: A new contribution to global virtual presence. *Aerospace and Electronic Systems Magazine, IEEE*, 23(10):11 –17, 2008.

[32] E. Navayazdani and T. P.-Y. Yu. On Donoho's Log-Exp subdivision scheme: Choice of retraction and time-symmetry. To appear in SIAM Multiscale Modeling and Simulation, August 2010.

[33] A. Nonnenmacher and C. Lubich. Dynamical low-rank approximation: applications and numerical experiments. *Mathematics and Computers in Simulation*, 79(4):1346–1357, 2008.

[34] T. Pappas, A. Laub, and N. Sandell Jr. On the numerical solution of the discrete-time algebraic Riccati equation. *IEEE Transactions on Automatic Control*, 25(4):631–641, 1980.

[35] I. U. Rahman, I. Drori, V. C. Stodden, D. Donoho, and P. Schröder. Multiscale representations for manifold-valued data. *Multiscale Modeling and Simulation*, 4(4):1201–1232, 2006.

[36] E. Sontag. *Mathematical control theory: deterministic finite dimensional systems*, volume 6. Springer, 1998.

[37] V. Studer, J. Bobin, M. Chahid, H. Moussavi, E. Candes, and M. Dahan. Compressive fluorescence microscopy for biological and hyperspectral imaging. *Arxiv preprint arXiv:1201.0650*, 2012.

[38] A. Weinmann. Interpolatory multiscale representation for functions between manifolds. *SIAM Journal on Mathematical Analysis*, 2012. to appear.

[39] G. Xie and T. P.-Y. Yu. Approximation order equivalence properties of manifold-valued data subdivision schemes. *IMA Journal of Numerical Analysis*, 2011. to appear.

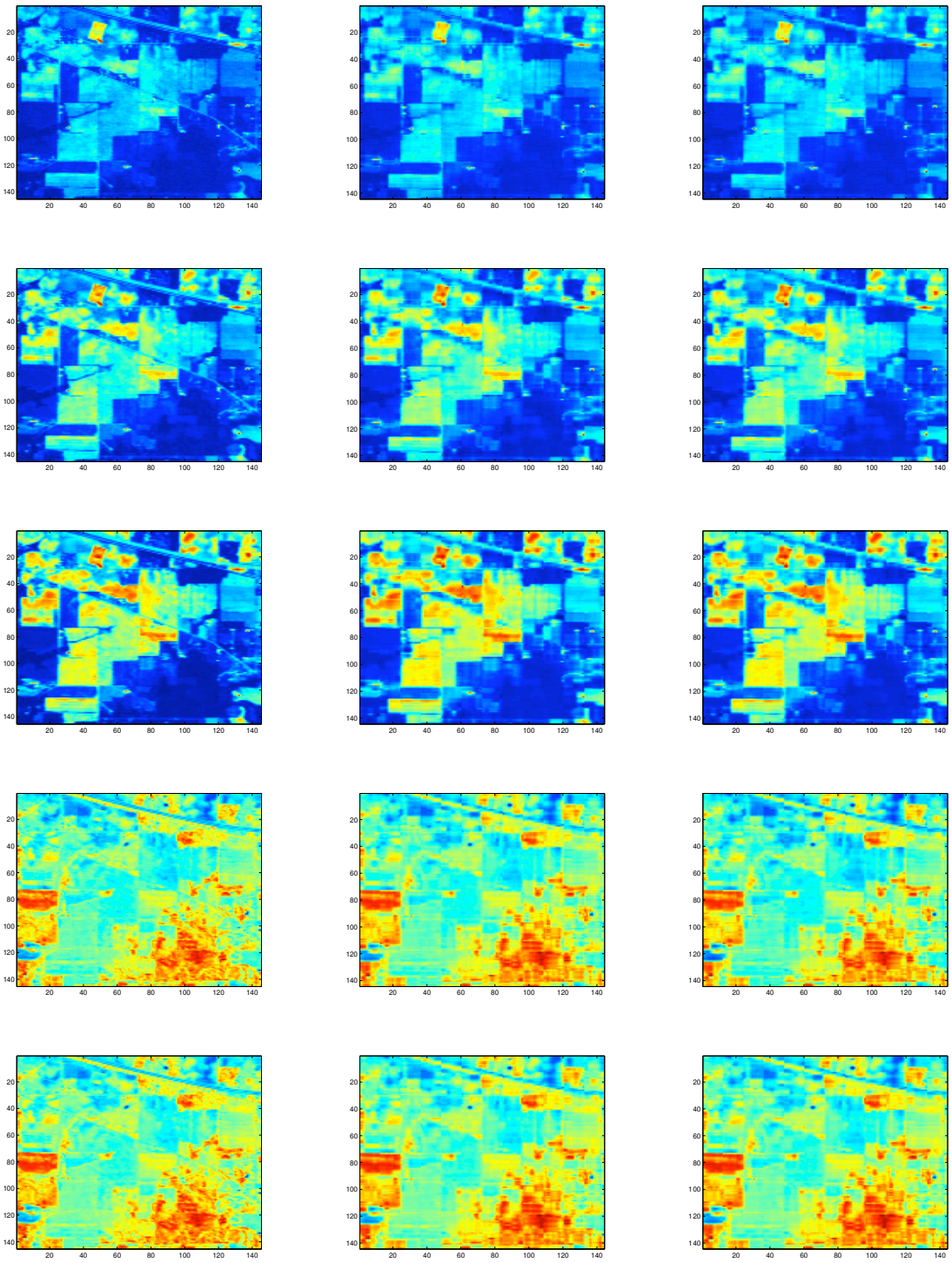


Figure 5: Algorithm 4 applied to hyperspectral image. Shown are wavelengths of about (from top to bottom): 355nm, 455nm, 555nm, 655nm, 755nm, 855nm. Left column: original frame. Middle column: low-rank approximation. Right column: wavelet compression.

Research Reports

No.	Authors/Title
12-03	*P. Grohs* Geometric multiscale decompositions of dynamic low-rank matrices
12-02	*D. Kressner and C. Tobler* htucker - A Matlab toolbox for tensors in hierarchical Tucker format
12-01	*F.Y. Kuo, Ch. Schwab and I.H. Sloan* Quasi-Monte Carlo methods for high dimensional integration - the standard (weighted Hilbert space) setting and beyond
11-72	*P. Arbenz, A. Hiltebrand and D. Obrist* A parallel space-time finite difference solver for periodic solutions of the shallow-water equation
11-71	*M.H. Gutknecht* Spectral deflation in Krylov solvers: A theory of coordinate space based methods
11-70	*S. Mishra, Ch. Schwab and J. Šukys* Multi-level Monte Carlo finite volume methods for shallow water equations with uncertain topography in multi-dimensions
11-69	*Ch. Schwab and E. Süli* Adaptive Galerkin approximation algorithms for partial differential equations in infinite dimensions
11-68	*A. Barth and A. Lang* Multilevel Monte Carlo method with applications to stochastic partial differential equations
11-67	*C. Effenberger and D. Kressner* Chebyshev interpolation for nonlinear eigenvalue problems
11-66	*R. Guberovic, Ch. Schwab and R. Stevenson* Space-time variational saddle point formulations of Stokes and Navier-Stokes equations
11-65	*J. Li, H. Liu and H. Sun* Enhanced approximate cloaking by SH and FSH lining
11-64	*M. Hansen and Ch. Schwab* Analytic regularity and best N-term approximation of high dimensional parametric initial value problems
11-63	*R. Hiptmair, G. Phillips and G. Sinha* Multiple point evaluation on combined tensor product supports