SHM NDT
International Symposium on Structural Health Monitoring and Nondestructive Testing
4-5 October 2018, Saarbruecken, Germany

EDGE PRESERVING COMPRESSION OF CT SCANS USING WAVELETS

A. Michael Stock[1,2*], Gabriel Herl[3,4], Tomas Sauer[1,2], Jochen Hiller[3,4]

[1]Fraunhofer CTMT, Innstraße 43, 94032 Passau, Germany
[2]FORWISS, University of Passau, Innstraße 43, 94032 Passau, Germany
[3]Fraunhofer CTMT, Dieter-Görlitz-Platz 2, 94469 Deggendorf, Germany
[4]Deggendorf Institute of Technology, Dieter-Görlitz-Platz 1, 94469 Deggendorf, Germany

[*]Corresponding Author: stock@forwiss.uni-passau.de

ABSTRACT

This work addresses the subject of efficient storage of computed tomography (CT) data with an emphasis on the quality of surfaces. Industrial dimensional metrology often requires high measurement accuracy and we show that this is retained using wavelet-based compression methods. The applied techniques include a tensor product wavelet transform and soft wavelet shrinkage. In our tests on real objects, we compared dimensional CT measurements of compressed and uncompressed volumes. We were able to reduce the necessary storage space significantly with a minimal loss of accuracy. For a multi sphere phantom, we decreased the storage space to 4.7% (from 638 MB to 30 MB) with an average deviation below 1 µm from the original volume.

KEYWORDS: Computed Tomography, Compression, Wavelets, Dimensional Metrology, Storage

1. INTRODUCTION

Efficient storage methods for handling huge amounts of data are of fundamental relevance in times where technology allows the generation of measurements occupying tens and hundreds of gigabytes of storage space. In this work, we study the efficiency of three-dimensional wavelet-based compression methods on CT scans and their influence on the accuracy of the boundary surfaces of the objects. In order to achieve this, we compare dimensional measurements of uncompressed and compressed CT volumes.

For decades, wavelet-based methods have been used to denoise and compress signals. Wavelets have even become part of the JPEG2000 standard for image compression. In the next section, we discuss an approach to extend these concepts to volume data compression that includes processing and transforming the data using tensor product wavelet methods, keeping a given percentage of the resulting wavelet coefficients after soft thresholding, storing the remaining coefficients in an efficient way in main memory, and writing these coefficients to a file, which can be read later for partial or full decompression. One main advantage of this approach is that CT scans consuming even substantially more than 10 GB of memory are now easily manageable, even on PCs that are not equipped with enough RAM to load such CT scans. The quality of this transformation is evaluated in the following section where we perform experiments with a calibrated multi sphere phantom and metal test specimens.

2. WAVELET-BASED COMPRESSION

This work is motivated and inspired by the well-known JPEG2000 standard. This standard defines how biorthogonal tensor wavelets can be used to compress 2D and 3D data [1]. It is based on the concept of a discrete wavelet transform (DWT) originating from a tensor product multiresolution analysis [2]. However, instead of quantizing the wavelet coefficients as in the JPEG2000 standard, we apply a soft-thresholding technique to the wavelet coefficients using the following assumption: In CT scans of industrial component parts, we usually obtain 'cartoon-like' data, i.e. largely homogeneous regions that are blurred by measurement noise. Furthermore, we can assume that the measured values in these regions yield only a small number of clusters that represent the materials of the scanned component part. For example, in an histogram of a scan of a motor piston, one can immediately see that the piston scan consists of only three materials, i.e. air, aluminum and iron, even if artifacts and blurring are present. Moreover, soft thresholding of wavelet coefficients with only one vanishing moment is closely related to total variation based denoising [3], which is often considered the state of the art in denoising nowadays. Thus, we are using soft thresholding instead of hard quantization.

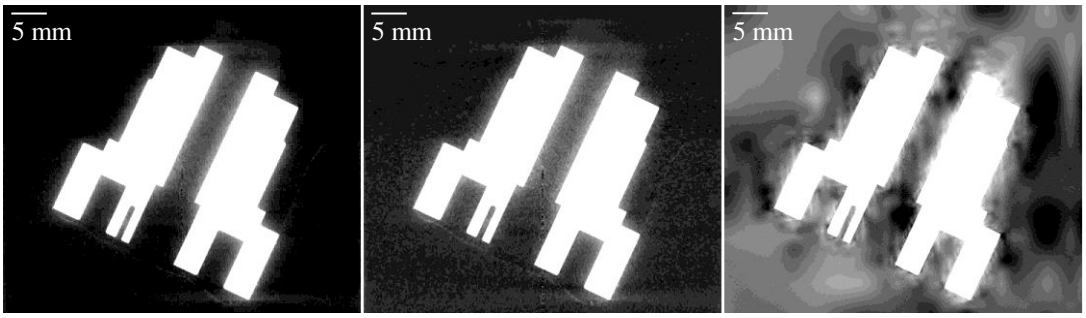

Fig. 1 Decompressed slices of an aluminum specimen after soft wavelet shrinkage for the Haar (left), CDF 5/3 (middle) and CDF 9/7 (right) wavelet transformations while keeping 1% of the wavelet coefficients. Brightness and contrast have been increased.

Choosing the wavelets. In the following, we will consider the biorthogonal Cohen-Daubechies-Feauveau (CDF) 5/3 wavelet transform that is used in JPEG2000, and due to the assumption that the given data is cartoon-like, we will also examine the Haar wavelet transform. However, we will omit the CDF 9/7 wavelet transform that appears in the JPEG2000 standard since it does not yield visually satisfying results when applying the proposed method, see Fig. 1. Recall that the choice of wavelet is always a trade-off between vanishing moments that give fast decay rates and good compression properties for smooth, non-constant objects and locality. Haar wavelets provide optimal localization since they have the smallest support of all discrete wavelets. Their main disadvantage, namely the fact that they work well only on piecewise constant images, is mostly irrelevant in the context of industrial workpieces. In view of the specific application, this work describes how core ideas of JPEG2000 can be modified to efficiently exploit the structure of cartoon-like data and massively reduce the memory requirements while still being able to make use of the powerful tool of the DWT.

2.1 THE ALGORITHM

The multiresolution analysis of the data is realized by performing discrete tensor wavelet transformations. We stress the fact that we do not work by means of layers. Instead, we apply a fully three-dimensional transform to the voxel dataset. This avoids inconsistencies between layers that otherwise would occur in the course of layerwise compression.

One-dimensional wavelet transform. In short, the DWT of a one-dimensional signal yields two new signals that represent the lowpass and the highpass parts of the original signal. These signals can be computed using convolutions. Due to the properties of the wavelet transform, we can perform downsampling by a factor of 2 of the two resulting signals without deleting information necessary for perfect reconstruction. Thus, the wavelet transform of a discrete signal uses as much space as the original signal. When reconstructing the original signal, we have to upsample first, which is done by inserting zeros between the lowpass and highpass signal elements, convolving again using appropriate filters, and finally adding the results, see Fig. 2 and the illustration in [1, p. 8]. The filter design defines the wavelets and vice versa.

The core building block of the proposed method is the one-dimensional DWT that is calculated using the lifting technique, see [1] and [4]. In this work, we consider the Haar wavelet transform and the CDF 5/3 wavelet transform. When performing CDF 5/3 lifting, we proceed as proposed in [1, p. 6], i.e., the coefficients on the border are mirrored. These one-dimensional wavelets and their associated filterbanks are then used to construct 3D wavelets by means of the standard tensor product construction. Though modern techniques such as shearlets could provide even higher compression rates, their higher computational effort prohibits the application to big data sets (100 GB and larger) so far.

The volume data is given by a 3D array usually filled with 16-bit unsigned integers. The volume is first embedded into a cube with an edge length of 2^n where n is a natural number. During this process, the data type of the voxels is changed to 64-bit floating-point numbers for maximal accuracy since floating-point arithmetic is used for the calculations (in contrast to JPEG2000's implementation of the CDF 5/3 integer-to-integer transform).

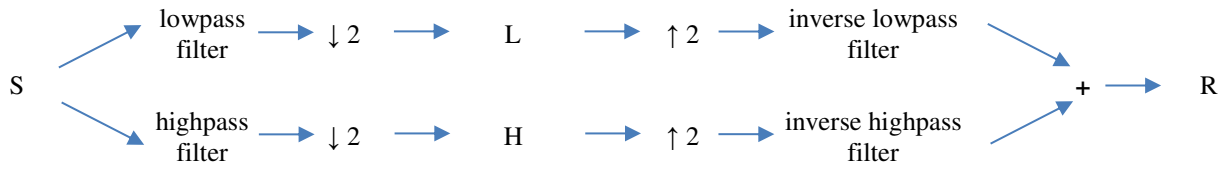

Fig. 2 Transforming a one-dimensional signal S into its lowpass (L) and highpass (H) components and backwards (R).

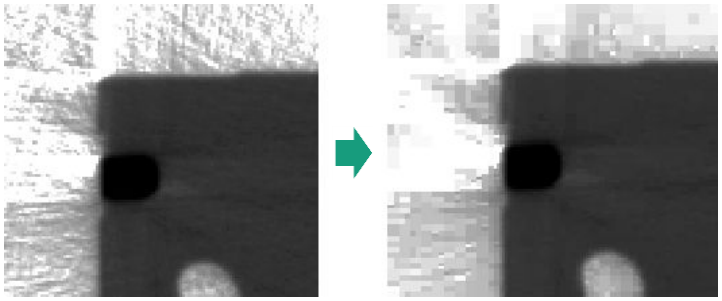

Fig. 3 Part of the uncompressed (left) and compressed (right) reconstruction of a motor piston using Haar wavelets: The soft wavelet shrinkage keeps sharp edges while reducing noise. The gray values have been inverted and logarithmically scaled.

The use of floating-point numbers allows us to work with arbitrary wavelets, not only those contained in the JPEG2000 standard. Recall that the CDF 5/3 wavelet has the property that the associated filter coefficients are dyadic numbers and thus the transform can be computed in some sort of fixed-point, i.e. integer, arithmetic even on weaker or specialized hardware such as digital signal processors. This arithmetic could, if needed, also be integrated into our approach.
Then, we perform a one-dimensional wavelet analysis transform in each direction. Note that the result is independent of the order of the directional transforms. Afterwards, we apply the same transform to the lowpass subcube of the transformed volume and iterate this process until we obtain a single voxel as lowpass content. Also, note that the computation time of the complete wavelet transform of a volume is linear with respect to the number of voxels.
At this point, we emphasize again that this wavelet transform truly operates in three dimensions, whereas some other wavelet-based methods consider 3D volume data as collection of subsequent 2D images that are compressed separately. Thus, the proposed method follows the JPEG2000 standard in this point.

Soft wavelet shrinkage. Instead of quantizing the resulting wavelet coefficients, we apply global soft thresholding on the detail coefficients based on their absolute value. Let τ be a positive real number denoting the threshold parameter. In contrast to hard thresholding, where all coefficients with an absolute value smaller than τ are set to zero, soft thresholding additionally shrinks the remaining coefficients towards zero. This is done using the shrinkage function S_τ, see [3]:

$$S_\tau(x) = \begin{cases} x - \tau \operatorname{sign}(x) & \text{if} \quad |x| > \tau, \\ 0 & \text{if} \quad |x| \leq \tau. \end{cases} \tag{1}$$

Note that there is an important relationship between the combination of the Haar wavelet transform with subsequent soft thresholding and total variation regularization. In the basic case of two pixels, the two transforms are even equivalent [3]. To highlight this relation, Fig. 3 illustrates how soft wavelet shrinkage mimics a total variation regularization that respects sharp boundaries and reduces noise. In particular, in connection with the Haar wavelet, the loss in our compression method is mainly the measurement noise.

Storing the coefficients. The remaining non-zero coefficients after thresholding are then transformed into two parts. The first part contains information about the coefficients regarding their position in the volume. This is done via run-length encoding (RLE) of zeros and non-zero data. The second part contains the non-zero coefficients only. Fig. 4 shows how a sparse 3D signal is transformed using this method. For this transformation, we interpret the 3D volume data (with dimensions X, Y, and Z) as a 1D signal where we first sort the coefficients according to their Z, Y and then X positions. The encoded data can be kept in memory, written to a file or read from a file. Table 1 provides details about the datasets, whereas Table 2 shows how much storage space can be saved using the proposed method. Note that keeping 1% of the wavelet coefficients yields a theoretical minimal storage consumption of 4% due to the change of the voxel type from 16-bit unsigned integers to 64-bit floating-point numbers, see the fourth column of Table 2. This could be improved by quantization, which is, however, not in the scope of this work. Since we do not yet use entropy coding there is still potential to reduce the required storage space, see the last column of Table 2.

2.2 LOCALITY AND MULTIRESOLUTION

Given the local nature of the wavelet transform, we demonstrate how the proposed techniques can be integrated into CT volume data analysis.

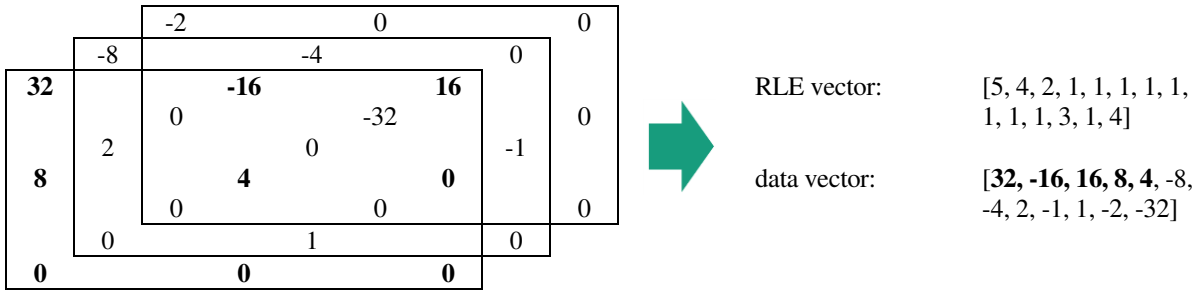

Fig. 4 Transforming a sparse 3D signal into zero-run-length encoded vectors. The values of the first XY slice are printed in bold type, the values of the second XY slice are printed in italic type. The input signal begins with 5 non-zero values (32, -16, 16, 8, 4), followed by 4 zeros that are followed by 2 non-zero values (-8, -4) and so on.

Volume Data	Voxel Data Type	Resolution	Embedding Cube Edge Length
motor piston	unsigned 16-bit integer	(464, 464, 414)	512
aluminum specimen	unsigned 16-bit integer	(992, 992, 850)	1024
Bavarian meat loaf in a bun	unsigned 16-bit integer	(2048, 2048, 1736)	2048

Table 1 Details about volume data examples.

Volume Data	Original Storage Space	Compressed Storage Space in Memory (RLE)	Compressed Storage Space on Hard Disk (RLE)	RLE and ZIP on Hard Disk
motor piston	170.0 MB	50.1 MB (29.5%)	7.61 MB (4.5%)	5.54 MB (3.3%)
aluminum specimen	1.55 GB	135.1 MB (8.5%)	73.9 MB (4.7%)	54.9 MB (3.5%)
Bavarian meat loaf in a bun	14.7 GB	778.3 MB (5.2%)	640.0 MB (4.3%)	426.0 MB (2.8%)

Table 2 Comparison of original and compressed storage space for three volumes using 1% of the Haar wavelet coefficients.

Local in-memory decompression. We consider a region of interest (ROI), which we assume to be a cuboid. For example, one of the main tasks in a GUI is to go through a volume slice by slice, where a slice is simply a flat cuboid. To decompress a ROI from the compressed volume, one has to obtain the necessary wavelet coefficients. The identification whether a voxel is needed depends on the filter length of the underlying wavelet. Then we locally apply the inverse DWT to obtain the decompressed image in the ROI. Note that both the extraction of the necessary wavelet coefficients and the (inverse) wavelet transform itself can easily be parallelized.

Multiresolution decompression. Depending on the desired resolution of a ROI, the decompression can be stopped at a specific level: When the volume is first compressed as a cube with length 2^n, there are n compression levels and thus, n different resolutions for the decompressed ROI are possible without expensive postprocessing. For example, when displaying slices, visualization software can use the information about the resolution on the screen to reduce the number of computations and to minimize the loading times.

2.3 EXAMPLES

To illustrate the performance of the proposed method, we first consider the aluminum specimen that is analyzed on a system equipped with a quadcore CPU at 4.20 GHz, 32 GB of RAM and a HDD.

Small data. Table 3 shows the runtimes of compressing and decompressing the aluminum specimen dataset from Table 1 while keeping 1% of the wavelet coefficients. Table 3 also shows that extracting a slice can be done in real time. Obviously, the order of dimension used when creating the run-length encoded representation of the wavelet coefficients increases the extraction time for XZ and YZ slices.

Big data. The JPEG2000 standard proposes tiling to process input data. Here, this approach is necessary especially if the input data is too large to fit into main memory. Therefore, we extract subcubes, process them separately, and combine the encoded data in the last step. In this work, we present results for tiling of large volumes using Haar wavelets.

	Compress the Data (RLE)	Read File from Hard Disk (RLE)	XY Slice (fixed Z coordinate)	XZ Slice (fixed Y coordinate)	YZ Slice (fixed X coordinate)
minimum		0.20 s	0.06 s	0.14 s	0.14 s
average	45 s	0.22 s	0.07 s	0.15 s	0.16 s
maximum		0.23 s	0.08 s	0.16 s	0.17 s

Table 3 Processing times (creating RLE file, reading from RLE file, and decompressing different types of slices in RLE memory) in the case of the aluminum specimen, 10 repetitions each except for creating the RLE file.

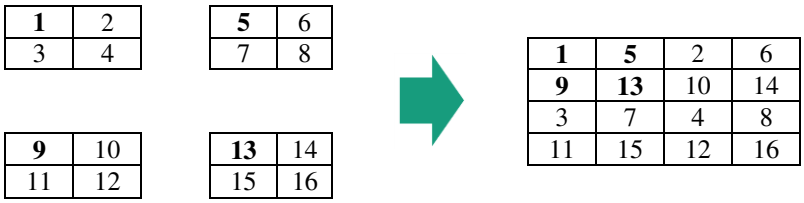

Fig. 5 Combining four 2D matrices in such a way that the resulting matrix contains the wavelet coefficients at the correct position. The lowpass coefficients are printed in bold type, the coefficients for X/highpass and Y/lowpass (the values in the top right corners of the 2D tiles) are printed in italic type.

	Compress the Data (RLE)	Read File from Hard Disk (RLE)	XY Slice (fixed Z coordinate)	XZ Slice (fixed Y coordinate)	YZ Slice (fixed X coordinate)
minimum		1.62 s	0.37 s	0.86 s	0.86 s
average	12 min	1.64 s	0.38 s	0.87 s	0.89 s
maximum		1.66 s	0.39 s	0.92 s	0.92 s

Table 4 Processing times (creating RLE file, reading from RLE file, and decompressing different types of slices in RLE memory) of the 'Bavarian meat loaf in a bun' volume dataset, 10 repetitions each except for creating the RLE file.

slice resolution	(512, 434)	(1024, 868)	(2048, 1736)
minimal loading time	0.17 s	0.37 s	0.86 s
average loading time	0.19 s	0.40 s	0.89 s
maximal loading time	0.22 s	0.42 s	0.92 s

Table 5 Comparison of loading times for a single YZ slice of the 'Bavarian meat loaf in a bun' volume, 10 repetitions each.

Fig. 5 illustrates how the merging algorithm works for four 2D tiles where the values in the top left corners are to be interpreted as the lowpass wavelet coefficients. Although the illustration shows the 2D case with only one level of details for simplicity, the 3D case is treated analogously. In order to obtain a complete wavelet transformation of the combined data, the lowpass coefficients undergo a final DWT and soft thresholding before the compressed data is written. Our merging algorithm is designed in such a way that the final output RLE volume can be written in an out-of-core manner, i.e., the RLE tiles are accessed simultaneously and, for each tile, all compression levels are also accessed simultaneously.

As an example, we consider the 'Bavarian meat loaf in a bun' dataset of size (2048, 2048, 1736) with a voxel size of 69.9056 µm and keep 1% of the wavelet coefficients. Table 4 shows the processing times of the 14.7 GB dataset using cubic tiles with an edge length of 1024. In this situation, processing the tiles takes about 9 minutes and combining the tiles needs about 3 minutes. Note that the amount of RAM necessary to perform the transformation depends on the size of the tiles and on the compression rate. In Table 4 we can see that the time required to extract slices from the compressed volume is almost negligible on the system specified above. Table 5 demonstrates how stopping the slice decompression at different resolutions affects the computation time.

3. EXPERIMENTS

3.1 SETUP

Several experiments were performed in order to examine the effect of the wavelet compression on dimensional metrology with CT. All of the specimens were scanned on a Werth TomoScope HV 500.

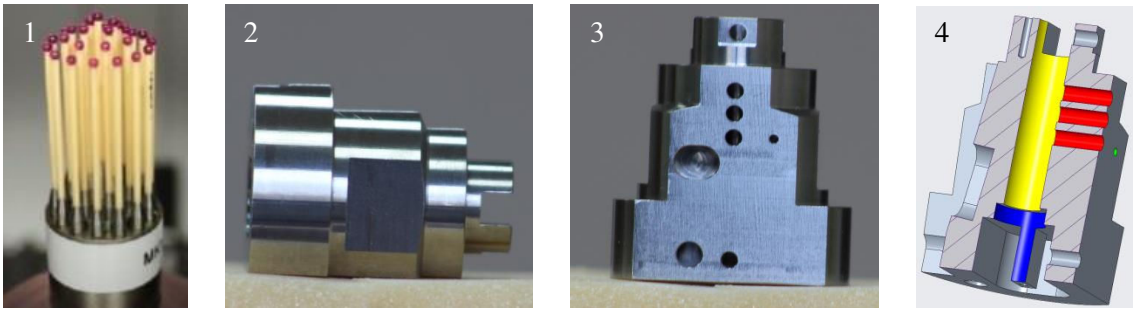

Fig. 6 The multi sphere phantom (1), the aluminum test specimen in two different positions (2 and 3), and a visualization of the six measurement areas (blue, yellow, red and green) of the test specimen (4).

After reconstruction with the Feldkamp reconstruction algorithm [5], the CT volumes were measured with the WinWerth metrology software for the first time. Then, we compressed every volume with the described method using only 1% of the wavelet coefficients. In order to be able to use the WinWerth metrology software, we decompressed these volumes and then measured them again.
On one hand, we performed experiments on a multi sphere phantom with hundreds of distances and diameters. On the other hand, we examined diameters on test specimens made of aluminum and steel in order to simulate a realistic metal workpiece.

Multi sphere phantom. The first specimen consists of 27 ruby spheres (approx. 2 mm diameters) mounted on a steel plate (see Fig. 6). We performed 15 scans with current 140 µA, voltage 180 kV, exposure time 666 ms and 800 projections. We measured 351 distances between spheres and the diameters of the 27 spheres. The reconstruction consists of voxels with a size of 35.2916 µm and it has a resolution of (991, 991, 341).

Metal test specimens. The metal specimens are made of aluminum (see Fig. 6) and steel. 20 scans of the aluminum specimen were performed with current 110 µA, voltage 225 kV, exposure time 1 s, 800 projections and a 1.0 mm tin filter. Another 20 scans of the steel specimen were performed with current 380 µA, voltage 225 kV, exposure time 1 s, 800 projections and a 3.0 mm tin filter. The reconstructions have a voxel size of 50.0339 µm and a resolution of (992, 992, 850). We measured 6 diameters on the aluminum specimen (see Fig. 6) and 5 diameters on the steel specimen (see Fig. 6, except the diameter of the blue area).

3.2 EXPERIMENT RESULTS

On a computer with two 2.40 GHz processors, every compression took about 30 to 35 seconds. The compression saved approximately 95.3% of the storage space, see Table 6. The results are visualized in Fig. 7, 8 and 9. Observe that these figures visually confirm the denoising effect of the compression process as predicted by the theory.

Specimen	Original Storage Space	Compressed Storage Space	Percentage
multi sphere phantom	638 MB	30 MB	4.7%
metal specimens	1.55 GB	74 MB	4.7%

Table 6 Storage space of the original and the compressed CT volumes containing 1% of the original wavelet coefficients.

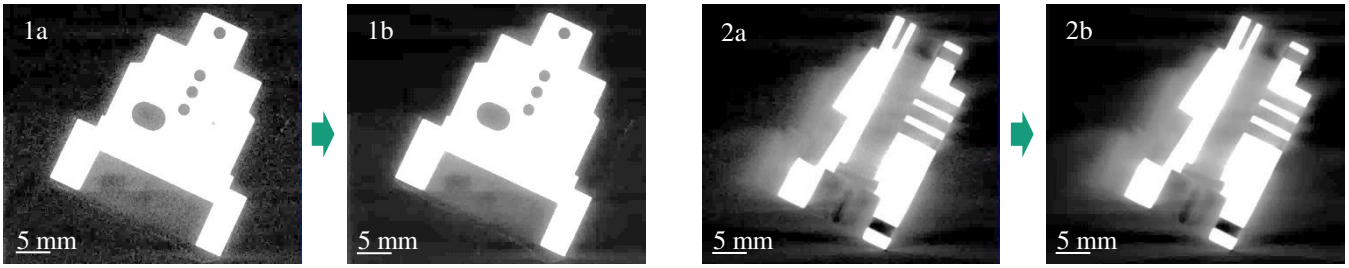

Fig. 7 Original volume (1a) and compressed volume (1b, 1% of wavelet coefficients) of an aluminum specimen; original volume (2a) and compressed volume (2b, 1% of wavelet coefficients) of a steel specimen.

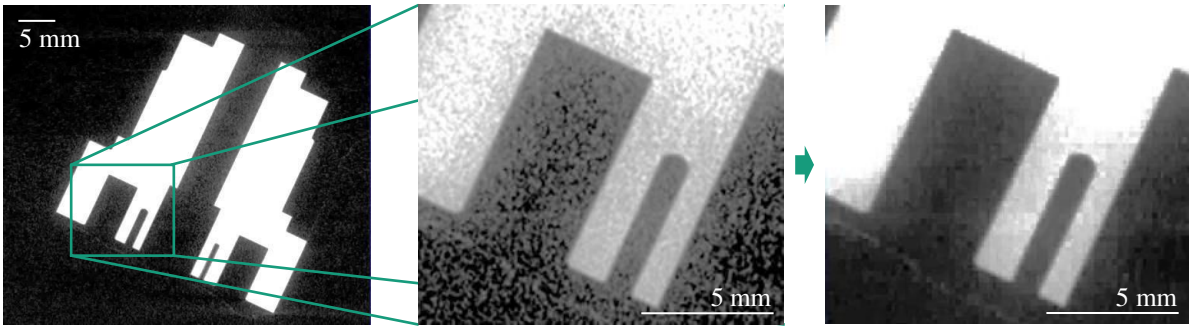

Fig. 8 Another slice of the original aluminum specimen volume (left), a cutout of the original volume (middle) and a cutout of the compressed volume (right, 1% of wavelet coefficients). The zoomed pictures have higher contrast for better visualization of the different noise and resolution.

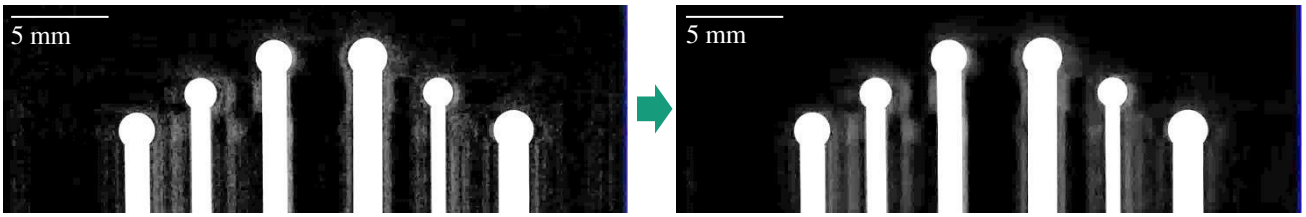

Fig. 9 Original volume (left) and compressed volume (right, 1% of wavelet coefficients) of the multi sphere phantom with very high contrast for visualization of the difference.

The dimensional metrology experiments showed very promising results for both the Haar wavelet compression (see Tables 7 and 9) and the CDF 5/3 wavelet compression (see Tables 8 and 9). At first, we compared the deviation of the compressed volumes to the original, uncompressed volumes (see Tables 7 and 8). In the best case (the distances of the multi sphere phantom), the measurements deviated by only 0.002% with the Haar wavelet compression and by 0.001% with the CDF 5/3 wavelet compression. In the worst case (the diameters of the steel specimen), the measurements deviated by 0.11% with the Haar wavelet compression and by 0.85% with the CDF 5/3 wavelet compression.

The results seem to indicate that more accuracy is lost by the presented compression method when the influence of metal artifacts rises. Although the results for the Haar wavelet and the CDF 5/3 wavelet compressions differ significantly, there is no compression method that clearly beats the other.

We also examined the deviation of all volumes to the calibrated measurements (see Table 9). All measurements but the diameters of the multi sphere phantom are calibrated with coordinate measuring machines accredited by the DAkkS GmbH. Therefore, we did not compare the diameters of the multi sphere phantom. In the best case (the multi sphere phantom distances), the measurements of the original deviated by 0.0087%, those of the Haar wavelet compression by 0.0084%, and those of the CDF 5/3 wavelet compression by 0.0085%. In the worst case (the diameters of the steel specimen), the measurements of the original deviated by 0.4093%, those of the Haar wavelet compression by 0.4655%, and those of the CDF 5/3 wavelet compression by 0.4523%. The measurements of the multi sphere phantom distances slightly improved after compression, which we suppose to be due to the intrinsic denoising property of the compression method that works similarly to a smoothing filter with edge preservation. Therefore, it is not even clear if the measurement of the uncompressed or the compressed image is more accurate in the absence of metal artifacts.

	Average Deviation	Standard Deviation	Maximum Deviation	0.95 Quantile
multi sphere phantom – distances	0.22 µm	0.24 µm	1.3 µm	0.5 µm
multi sphere phantom – diameters	0.61 µm	0.46 µm	2.9 µm	1.5 µm
aluminum specimen – diameters	1.35 µm	1.20 µm	5.4 µm	3.6 µm
steel specimen – diameters	3.82 µm	7.98 µm	53.4 µm	9.2 µm

Table 7 Dimensional metrology deviations of the Haar wavelet compression in comparison to the original, uncompressed reconstruction.

	Average Deviation	Standard Deviation	Maximum Deviation	0.95 Quantile
multi sphere phantom – distances	0.14 µm	0.19 µm	1.2 µm	0.4 µm
multi sphere phantom – diameters	0.91 µm	0.64 µm	3.8 µm	2.8 µm
aluminum specimen – diameters	2.03 µm	1.58 µm	8.1 µm	5.0 µm
steel specimen – diameters	3.02 µm	6.87 µm	50.8 µm	7.9 µm

Table 8 Dimensional metrology deviations of the CDF 5/3 wavelet compression in comparison to the original, uncompressed reconstruction.

	Uncompressed Average Deviation	Haar Wavelet Compression Average Deviation	CDF 5/3 Wavelet Compression Average Deviation
multi sphere phantom – distances	0.96 µm	0.93 µm	0.94 µm
aluminum specimen – diameters	7.86 µm	7.96 µm	8.45 µm
steel specimen – diameters	14.75 µm	16.78 µm	16.33 µm

Table 9 Dimensional metrology deviations of all volumes described in section 3.1 to the calibrated measurements.

3.3 STUDY OF DIFFERENT COMPRESSION RATES

In order to examine the influence of the compression rate, we compressed the aluminum specimen several times with a different percentage of remaining Haar wavelet coefficients. The original volume required 1.55 GB storage space. The results can be seen in Fig. 10 and Table 10. In the volume with only 0.1% of the remaining wavelet coefficients, some of the measurements could not be performed anymore.

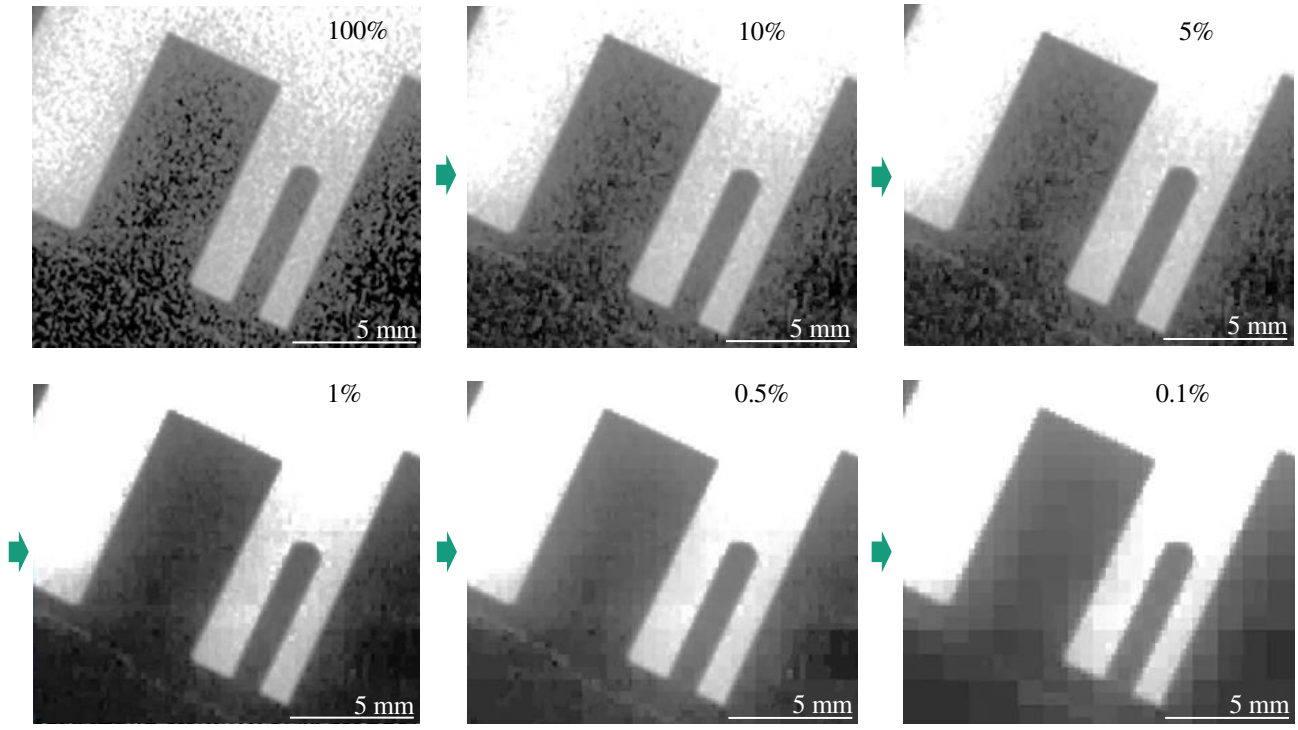

Fig. 10 Comparison of compressed volumes with a different percentage of remaining wavelet coefficients of the same region as in Fig. 8.

Percentage of Remaining Wavelet Coefficients	Average Deviation	Standard Deviation	Storage Space	Percentage of Original Storage Space
10%	0.81 µm	0.74 µm	730 MB	47.01%
5%	1.03 µm	0.91 µm	369 MB	23.81%
1%	1.61 µm	1.05 µm	73.9 MB	4.77%
0.5%	1.11 µm	1.37 µm	36.7 MB	2.37%
0.1%	-	-	7.29 MB	0.47%

Table 10 Comparison of different compression settings for Haar wavelet compressions. The columns feature the percentage of remaining wavelet coefficients, the average deviation from the uncompressed volume, the standard deviation of these deviations, the storage space of the volumes and the percentage of the original storage space.

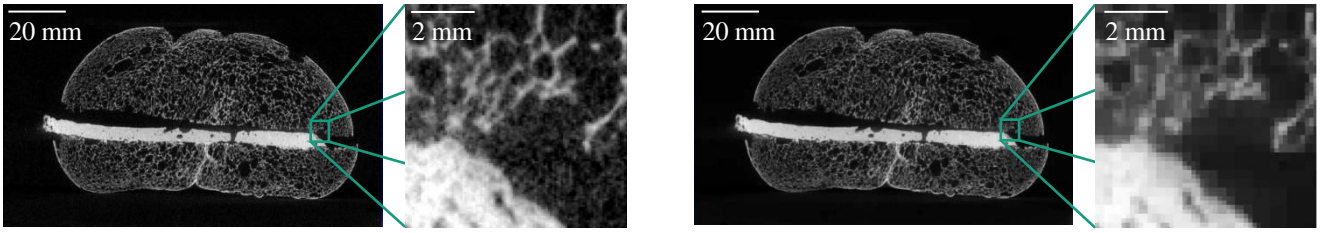

Fig. 11 Original (left) and compressed (right) CT volume of a typical Bavarian meat loaf in a bun.

3.4 AN INHOMOGENEOUS EXAMPLE

As explained, the proposed wavelet compression method works best on homogeneous objects such as most objects in industrial use cases. In order to show a negative example, we performed the method on an inhomogeneous object, too. Therefore, we scanned a typical Bavarian meat loaf in a bun. We compressed the volume using 1% of the wavelet coefficients. In the compressed volume, noise is reduced again, but significantly stronger than in the experiments before, blurring and a loss of resolution is visible, see Fig. 11.

4. CONCLUSION

We have shown that a massive reduction of storage space while obtaining sharp edges, good image quality and even dimensional metrology is possible with a fully three-dimensional wavelet transform followed by soft thresholding of the coefficients. Obviously, the rate of compression has to be matched to the digitalization or measurement task. For many use cases, more than sufficient accuracy is retained and hence resources can be saved with the correct compression approach. Moreover, the compression even has a denoising effect on the image.

This is especially important for many prospective use cases such as deep learning, which have to accumulate huge amounts of data. In order to optimize the method for these use cases, there should be an adapted compression approach that could, for example, learn the wavelet coefficients. Therefore, we expect a lot of further scientific impact from wavelet and related approaches in CT volume compression.

ACKNOWLEDGMENT

This work is part of the project ‘Digitalisierung, Verarbeitung und Analyse kultureller und industrieller Objekte: Wertschöpfung aus großen Daten und Datenmengen – Big Picture’ that has been promoted and funded by the Bavarian State Government under Grant No. AZ.: 43-6623/138/2 since 2017.

REFERENCES

[1] M. Rabbani and R. Joshi, “An overview of the JPEG 2000 still image compression standard,” Signal Processing: Image Communication. 2002.
[2] S. Mallat, A Wavelet Tour of Signal Processing. 2009.
[3] G. Steidl, J. Weickert, T. Brox, P. Mrázek, and M. Welk, “On the Equivalence of Soft Wavelet Shrinkage, Total Variation Diffusion, Total Variation Regularization, and SIDEs,” SIAM J. Numer. Anal., 2004.
[4] W. Sweldens, “The Lifting Scheme: A Construction of Second Generation Wavelets,” SIAM J. Math. Anal., 1998.
[5] L. A. Feldkamp, L. C. Davis, and J. W. Kress, “Practical cone-beam algorithm,” J. Opt. Soc. Am. A, 1984.