Proc. Vth International Workshop, Advances in Signal Processing for Non Destructive Evaluation of Materials
Québec City (Canada),2-4 Aug. 2005. © X. Maldague ed., É. du CAO (2006), ISBN 2-9809199-0-X

Effective compression algorithms for pulsed thermography data

by S. Lugin and U. Netzelmann

Fraunhofer-Institute for Nondestructive Testing, University Bldg.37, 66123 Saarbrücken, Germany

Abstract

Two compression algorithms for the image sequences generated by pulsed-transient thermography for non-destructive testing were developed. The first algorithm allows to balance the quality of the original measurement data reproduction against the compression ratio. This algorithm comprises a dedicated space/time mapping (STM) method and an image compression algorithm (JPEG2000). The second algorithm provides lossless reproduction of the original measurement data. This algorithm is based on a particular transformation of dynamically changing data and a lossless compression algorithm (ZIP). Both algorithms were tested on typical experimental thermography data. In both cases, the achieved compression ratios were significantly higher than those of existing algorithms.

Keywords: thermography; data compression; space/time mapping

1. Introduction

Pulsed thermography (PT) has experienced wide application during the recent years due to its unique features like: contact-free operation, capability to inspect large areas simultaneously and fastness of inspection. The technical equipment and method have now reached a stage of maturity which allows them to be used for in-line full-time quality control of components. This kind of testing allows for the detection of subsurface defects, inclusions and delaminations as well as for materials characterization [1]. The inspection usually consists of three phases:

1) *Thermal excitation*
2) *Observation of the thermal response*
3) *Data analysis*

The problem we discuss in this work concerns an efficient compression of the recorded IR image sequence as obtained after phase 2. On one hand, the reconstructed measurement data should match the original data with high accuracy, on the other hand excessive data storage volume as it might occur in 24-hours process control applications has to be avoided.

As an introduction, the raw format in which the equipment stores information is briefly described. An IR camera that is used for PT may have a temperature range from -10 °C up to 100 °C. We use a focal-plane array mid-range IR camera (AIM). The IR signal is digitized with 14 bits accuracy (digital discretization). It is assumed that the camera array has an image size of 256 x 256 pixels and stores the data as words of 16 bits. According to these characteristics the size $Size_{seq}$ of the image sequence (comprising N images in raw format) is computed as:

$$Size_{seq} = N \cdot (256 \cdot 256) \cdot 2\,bytes$$

(for example, if N=200 then $Size_{seq}$= 25 MB)

Due to the signal characteristics of the image sequence, the raw data contain a significant amount of redundant information. As the thermal response of the inspected sample may be described as a diffusion process, some images after the flash occurred, the whole sequence will always change smoothly over time, i.e., the temperature of any chosen location will not change to significant amounts between adjacent frames. Based on this property, we propose two compression algorithms for pulsed thermography data. Note that, in general, there are two kinds of data compression: lossless compression and compression with slight losses (the latter being frequently used for the storage of images as well as audio and video data). Both paths are pursued in this work.

2. Raw measurement data format

Prior to the description of the algorithms, the raw data format as provided by the PT equipment is considered. During the measurement the IR camera generates the sequence of *S* images, each image having the same size (*M*x*N* pixels), where each image pixel is represented by a word of 16 bits. The whole sequence is stored on a media device as a single file. Further, in the paper we use the abbreviation $IM_{raw}(i,j,t)$ to address to the IR value in the point *i,j* ($1 \le i \le N, 1 \le j \le M$) at the time *t* ($1 \le t \le S$).

3. Lossy data compression

3.1 Time signal reconstruction (TSR) method

To estimate the efficiency of the algorithm proposed in the following, its results will be compared with the results of the thermographic signal reconstruction (TSR) method [2, 3] which was especially developed for pulsed thermography. The TSR method serves as a reference for the algorithm proposed in the following.

3.2 Space/time mapping JPEG-based data compression

A unique aspect of the algorithm we propose is to combine spatial and temporal information in the cooling image sequences. The main idea lies in a transformation of the recorded data $IM_{raw}(i,j,t)$ into a single image STM to be further compressed [4]. The algorithm consists of three steps:

1) *Extraction of the dynamically changing part of the data*
2) *Space/time mapping*
3) *JPEG compression*

The first and second steps were developed and described in detail in our previous work [4]. The novelty in this work is an application of the common JPEG2000 algorithm [5] for compression of the image STM. This image compression algorithm was chosen for its high compression ratio, high reconstruction quality and high speed. The algorithm accepts the *quality/size* ratio as an input parameter for the compression procedure. The range of this parameter and its influence depends on the particular implementation of the algorithm.

In our implementation of the algorithm we used the programming language LabVIEW to produce the image STM as a BMP file and the image processing package Corel PHOTO-PAINT to compress it. The package provides JPEG2000 compression as a function of exporting that requires an input parameter *CMP* in the range 1-100 for varying the *quality/size* ratio. The value 1 corresponds to high quality at low compression, and the value 100 corresponds to low quality at high compression.

In section 3.3 we present the results of pulsed thermography data compression for several values of *CMP*.

3.3 Compression results

In most the STM-JPEG algorithm compressed data tightly and provided high reproduction quality. In the following, results are presented of two example IR image sequences compressed with different values of *CMP*. The quality and file size are compared to those obtained by use of the TSR method. A parameter *RL* is used as a measure for reproduction loss estimation:

$$RL = \frac{1}{MN} \sum_{i}^{N} \sum_{j}^{M} RMSE_{ij} , \tag{1}$$

where *RMSE*$_{ij}$ is the root mean square error computed against the actual cooling curve at the point *i,j* of the image sequence:

$$RMSE_{ij} = \sqrt{\frac{1}{S}\sum_{t}^{s}\left(IM_{raw}(i,j,t) - IM_{rep}(i,j,t)\right)^2} \, , \qquad (2)$$

where *IM*$_{rep}$*(i,j,t)* is a pixel with the coordinates *i,j* in the image number *t* of the decompressed image sequence.

Remark: In order to suppress image noise which is inevitable under practical circumstances, a 3 x 3 median filtering was applied to the measured data before storing them in raw format.

In the first example, a polyvinylchloride (PVC) plate (thickness: 1 cm) containing two lengthy grooves simulation sub-surface defects (depths: 0.7 mm and 1 mm) was tested. Flash lamp heating was employed. The image sequence included 188 images with a size of 256 x 256 pixels. The data in the centre region of interest (128 x 128 pixels) were extracted and stored in raw format, which required 5.9 MB of storage space. Further the data were compressed by TSR and the STM-JPEG algorithm and their reproduction estimated by Eq. (1). Table 1 shown below summarizes the result of compression at different values *CMP* of the JPEG compression.

Table 1. Compression results on a PVC plate with buried grooves

Factor	TSR method	STM-JPEG algorithm				
		CMP=1	*CMP*=20	*CMP*=40	*CMP*=60	*CMP*=80
Size, KB	320	314	241	139	77	32
Reproduction losses (*RL*)	5.95	2.79	3.14	3.84	4.55	5.50

The STM-JPEG compressed file size can be one tenth of that of the TSR method, or to a total compression factor of 184 at *CMP*=80. At the same *CMP*, the reproduction losses are still at a lower level than for TSR. In a trade-off with file size, *RL* can be made significantly smaller. This may be helpful, if defect reconstruction algorithms will be applied on the IR data.

In the second example, the object under test was a sample made from polyethylene which contained 4 circular defects of different diameters in different depth (diameter/depth: 3/1.2 mm, 5/1 mm, 3/2 mm and 5/2 mm). The results are similar to that of the first example. Fig. 1(a) shows a thermographic image of the test specimen from the cooling sequence. The entire sequence contained 98 images with a frame spacing of 0.35 s.

Again, the centre region of 128 x 128 pixels was extracted and stored in raw data format (3 MB). The compression results are shown in Table 2.

Table 2. Compression results on a polyethylene sample with flat bottom holes

Factor	TSR method	STM-JPEG algorithm				
		CMP=1	*CMP*=20	*CMP*=40	*CMP*=60	*CMP*=70
Size, KB	320	226	183	134	82	59
Reproduction losses (*RL*)	5.02	2.49	2.81	3.29	4.03	4.55

The reproduction losses are shown in Fig. 1(b,c) for the reference algorithm (TSR) and the STM-JPEG algorithm proposed here. The grey values represent the *RMSE*$_{ij}$ as defined in Eq. (2). A perfect reconstruction would produce a black image. It is obvious that the proposed algorithm reduces the reproduction error significantly, in particular at the defect positions.

4. Lossless data compression

The problem of lossless data compression has been widely discussed. The most famous algorithms developed for solving this problem and using nowadays are Huffman coding, LZW coding and arithmetic coding [6].

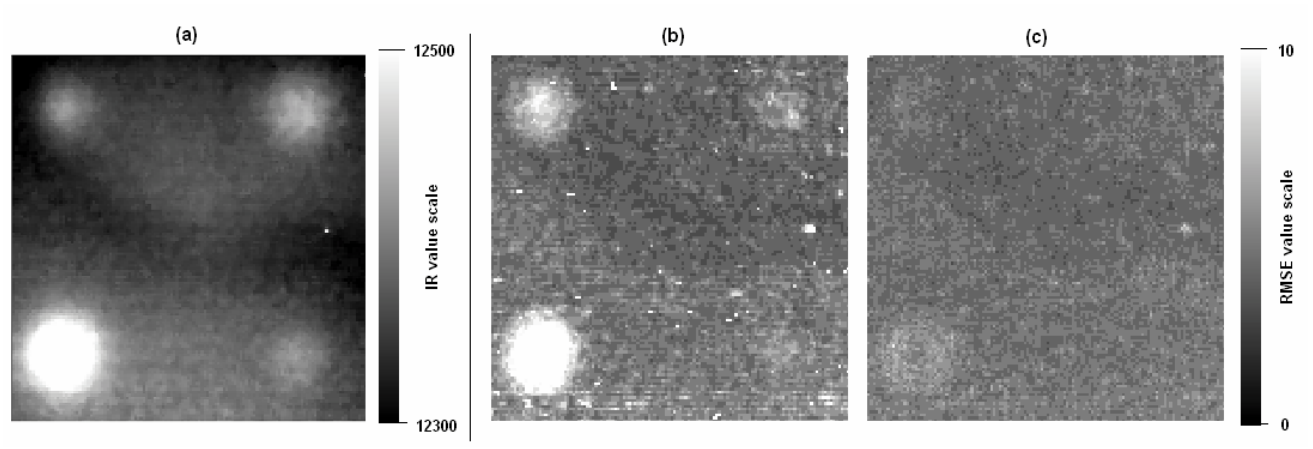

Fig. 1 (a) - Thermographic image (frame 40) from the cooling sequence. The four sub-surface defects are visible close to the corners (image size 37 mm x 37 mm). (b,c) Images of the reproduction loss obtained by both algorithms ((b) - TSR method, (c) - STM-JPEG compression at *CMP*=70), the grey values represent the $RMSE_{ij}$ as defined in Eq. (2).

These algorithms are based on different approaches but have in common that they exhibit a dependence of the compression ratio on the data to be compressed. A data block having a high number of repetitions can be compressed tighter than a data block having a low repetition number. In some cases the transformation of a source data block based on the information context allows to intentionally increase the number of repetitions and in consequence, to increase the compression ratio. Considering pulsed thermography data in the raw format one can note that there are few repetitions, so this does not allow to achieve a high compression ratio using existing algorithms. As an approach to this problem, it is necessary to transform the data into a form suitable for compression. In this work we propose an algorithm that consists of a dynamically changing data transformation (DCDT) of measurement data and a compression package (ZIP) chosen as a widespread lossless compression package.

To compress pulsed thermography data, the algorithm performs three steps:

1) *Transformation of dynamically changing data (DCDT)*

   The algorithm separates dynamical and statical information, stores the data in the form suitable for compression.

2) *Lossless compression*

   The transformed data is compressed by the ZIP algorithm.

4.1 Transformation of dynamically changing data

In this step the algorithm separates dynamical and statical information the data contains. As statical information, the average cooling process is considered. All image IR values $IM_{raw}(i,j,t)$ at $1 \le t \le S$ decrease with more or less similar speed that is explained by a transient cooling process after thermal excitation. To separate the cooling process and dynamical changes (dc), the algorithm computes average image temperatures $AV(t)$:

$$AV(t) = \left[\frac{\sum_i \sum_j IM_{raw}(i,j,t)}{MN} \right], \tag{3}$$

and subtracts the computed temperatures from source images:

$$IM_{dc}(i,j,t) = IM_{raw}(i,j,t) - AV(t). \tag{4}$$

The row *AV(t)* represents the average process of cooling while the values *$IM_{dc}(i,j,t)$* describe the thermal dynamic changes caused by internal thermal reflections. Further the algorithm extracts dynamic changes of single image pixels (sip) computing the difference of pixel values between adjacent images:

$$IM_{dc-sip}(i,j,t) = IM_{dc}(i,j,t) - IM_{dc}(i,j,t+1) \qquad (5)$$

Note that the difference *$IM_{dc-sip}(i,j,t)$* is computed for all pixel values except the last image *S*. The performed computations significantly increase numbers of repetitions in the data *$IM_{dc-sip}(i,j,t)$*. If the probability histogram of *$IM_{dc-sip}(i,j,t)$* is considered, the maximal probability will be close to 0 and rapidly decrease from 0 to $+\infty$ and from 0 to $-\infty$. In binary form it has a following feature that bits of low order contain most of changes while bits of high order almost always equal to zero. In order to effectively use this feature, the algorithm carries out a bit separation by bit order. Note, that we choose 16-bits integer form to store values *$IM_{dc-sip}(i,j,t)$* where the highest bit is used for a sign and 15 bits are used for the number coding. The algorithm forms 16 single data chains by the following equation:

$$DC(n) = \perp_t \perp_i \perp_j \left(IM_{dc-sip}(i,j,t)_{[n]}\right) \qquad \textit{for n=0…15} \qquad (6)$$

where the abbreviation $IM_{dc-sip}(i,j,t)_{[n]}$ denotes an extraction of the n-th bit from the value $IM_{dc-sip}(i,j,t)$ and the abbreviation $\perp_k a(k)$ denotes a bit concatenation of elements a in forward order: $a(1), a(2), a(3)...$. This transformation is schematically shown in Fig 2. To store data chains, they are converted in the byte form.

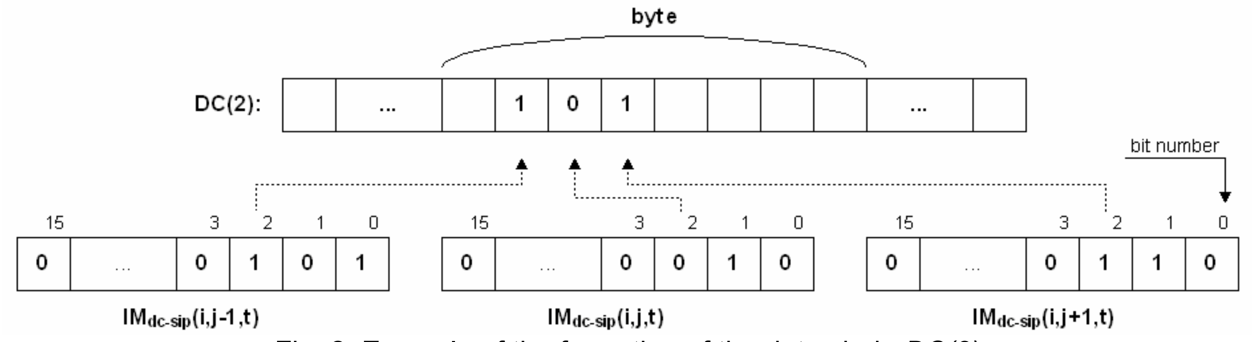

Fig. 2. Example of the formation of the data chain *DC(2)*

After these steps have been done, the algorithm stores a data structure that includes the raw *AV(t)*, the last image *$IM_{dc}(i,j,S)$*, and 16 data chains *DC(n)*. For simplicity of description we assume that they are stored as single files to be compressed: a file *av-t.raw* for raw *AV(t)*, a file *im-dc-s.img* for the last image *$IM_{dc}(i,j,S)$*, files *dc-0.dat*, *dc-2.dat* … *dc-15.dat* for data chains *DC(n)*, a file *descr.txt* that stores supplemental information – the number of images, image resolution and etc.

4.2 Lossless compression

There are a lot of software packages we can use for file compression. ZIP, TAR, RAR, UHA are the most famous of them. They differ from each other in internal compression algorithms and their implementations. To compress data obtained in previous step of our algorithm, we choose to apply the package ZIP due to its popularity, availability and high speed. Note that indeed any lossless compression package can be used instead of ZIP. The results of data compression are presented in Section 4.3. All obtained files *av-t.raw*, *im-dc-s.img*, *descr.txt*, *dc-0.dat* … *dc-15.dat* are compressed by the package ZIP to a single file. The decompression procedure is carried out in inverse order.

4.3 Compression results

The developed algorithm was also applied to pulsed thermography data measured by the testing system of Fraunhofer IZFP. The algorithm shows better compression ratio than pure ZIP compression of the data in raw format. Table 3 summarizes the result of compression of two image sequences described in Section 3.3 (PVC plate with buried grooves (sample 1), polyethylene sample with flat bottom holes (sample 2)). As seen, the algorithm, namely the DCDT transformation, allows significantly to increase the compression ratio.

Table 3. Compression results on PVC and polyethylene sample

Sample	file size, MB		
	data in raw format (uncompressed)	ZIP compression	DCDT-ZIP compression
Sample 1	5.88	3.15	1.55
Sample 2	3.06	1.37	0.84

To complete the algorithm description, we note that the transformation DCDT should be adjusted to the compression package we apply. It concerns the last step (Eq. 6) when the data chains *DC(n)* are formed from $IM_{dc\text{-}sip}(i,j,t)$. Our tests have shown that, if ZIP is used, the transformation (Eq. 6) is useful, as it increases the compression ratio. But some packages (e. g. RAR in maximal compression mode) compress the data $IM_{dc\text{-}sip}(i,j,t)$ and *DC(n)* with almost the same ratios. So, the transformation (Eq. 6) can be skipped. Nevertheless, we emphasize that the DCDT transformation in the form we described is the most suitable and universal for any data or text compressor.

5. Conclusions and future work

In the present work, the new lossy and lossless compression algorithms for pulsed thermography data were proposed. The STM-JPEG algorithm combines a priori knowledge of the diffusion phenomena, the specific space/time representation of dynamically changing data and a well established static image compression algorithm. This combination achieves a high compression ratio while preserving high reconstruction quality. The DCDT-ZIP algorithm involves the specific transformation of dynamically changing data and a well established lossless data compression algorithm. The developed algorithms have demonstrated good results when applied to various experimental data sets measured by flash thermography.

Concerning further improvements, one can suggest for the STM-JPEG algorithm using 16-bit grey scale JPEG2000 compression. It will allow to preserve initial camera sensitivity (avoiding 256-level discretization) while forming the STM image.

REFERENCES

[1] RÖSNER (H.), NETZELMANN (U.), HOFFMANN (J.), KARPEN (W.), KRAMB (V.), MEYENDORF (N.). *Thermographic Materials Characterization.* In: Meyendorf N, Nagy P and Rokhlin S (eds.), Springer Series in Materials Sciences (Springer Verlag, New-York 2004) 247-285.

[2] *U. S. Patent 6,516,084*

[3] SHEPARD (S.M.), Ahmed (T.), RUBADEUX (B.A.), WANG (D.) and LHOTA (J.R.). *Synthetic Processing of Pulsed Thermographic Data for Inspection of Turbine Components.* In: Insight, Vol. 43 No. 9, Sept 2001, British Inst. of NDT, pp 587-589

[4] LUGIN (S.) and NETZELMANN (U.). *An effective compression algorithm for pulsed thermography data.* In: NDT&E International, Vol. 38, 2005, pp 485-490

[5] TAUBMAN (D.S.) and Marcellin (M.W.). *JPEG2000: Fundamentals, Standards and Practice.* Boston: Kluwer Academic Publishers, 2002.

[6] NELSON (M.) and GAILLY (J.). *The Data Compression Book.* New York: M&T Books, 1995.