# Alignment of Noisy and Uniformly Scaled Time Series

Constanze Lipowsky, Egor Dranischnikow, Herbert Göttler, Thomas Gottron,
Mathias Kemeter, Elmar Schömer

Institut für Informatik, Johannes Gutenberg-Universität Mainz
55099 Mainz, Germany
{lipowsky, dranisch, goettler, gottron, schoemer}@informatik.uni-mainz.de,
kemeter@gmail.com

**Abstract.** The alignment of noisy and uniformly scaled time series is an important but difficult task. Given two time series, one of which is a uniformly stretched subsequence of the other, we want to determine the stretching factor and the offset of the second time series within the first one. We adapted and enhanced different methods to address this problem: classical FFT-based approaches to determine the offset combined with a naïve search for the stretching factor or its direct computation in the frequency domain, bounded dynamic time warping and a new approach called shotgun analysis, which is inspired by sequencing and reassembling of genomes in bioinformatics. We thoroughly examined the strengths and weaknesses of the different methods on synthetic and real data sets. The FFT-based approaches are very accurate on high quality data, the shotgun approach is especially suitable for data with outliers. Dynamic time warping is a candidate for non-linear stretching or compression. We successfully applied the presented methods to identify steel coils via their thickness profiles.

**Keywords:** time series, linear time warping, alignment, stretching factor, offset, FFT, bounded dynamic time warping, shotgun analysis, linear regression

## 1 Introduction

Given two time series $X$ and $Y$, where $Y$ is a subsequence of $X$ in the sense that, for a human observer, $Y$ looks similar to a part of $X$, we want to find an alignment of $X$ and $Y$. However, compared to $X$ the values of $Y$ are more or less distorted. What makes the problem worse is the fact that $Y$ is either stretched or compressed relative to its corresponding part of $X$. Now, our aim is to align the two data series automatically, that means to find the counterpart of $Y$ within $X$ in spite of distortion and stretching or compression. To achieve this, we need to determine the offset and the stretching or compression factor of $Y$ relative to $X$. Figure 1 illustrates this situation: The two plotted data series look similar but are not easy to compare by a computer because the thinner, brighter one is stretched relative to the thicker, darker one.

This problem is highly relevant in practice, e.g., in steel production where the problem was posed and our test data comes from. All our methods can also deal with overlapping data series. The assumption that the second time series is part of the first one is not necessary but is true for the following practical example.

In the production of steel, coils are important (semi-finished) goods. A coil is a flat wound up steel strip of a certain width and thickness and varying length from a few hundred up to several thousand meters. During the production process a coil passes through different machines and steps. It is repeatedly unwounded, lumbered, cut into pieces, welded together with parts of other coils and wound up again. In between the production steps the coils are stored in the company's interim storage facility. In order to control the whole process and to retrace the origin of a certain piece of steel, it is important to be able to follow a coil and/or pieces of it throughout the whole production process. Therefore, the company *iba AG* in Fürth, Germany came up with the idea to derive a unique identification of each coil via its thickness profiles, the so called "fingerprints" [1], similar to the fingerprints which are unique for humans. The thickness of each coil is measured in certain fixed time intervals before and after each production step (e.g. every 10 ms). The *varying* throughput speed during this process is measured through the varying velocity of the rollers. So, it is possible to convert the time based data to locations on the steel strip – an information which is more interesting in the context of quality control. Based on these measures the thickness of the coil at certain positions or in certain discrete distances (typically every ten centimeters) is calculated by linear interpolation. Two fingerprints of the same coil after one and before the next production step are always similar but not identical. Differences arise due to different measuring devices, measuring inaccuracies, failure of measuring devices and the generally difficult circumstances of the production process (dirt, steam, large temperature differences and changes in the material). Surprisingly, some particular production steps (e.g. galvanization) do not change the fingerprint too much, so it is still possible to recognize the coil afterwards with our techniques.

Inaccuracies in thickness measuring lead to vertical errors, inaccuracies in speed measurement cause horizontal displacements. Because of the latter ones, it is not possible to find the optimal starting position of the second fingerprint within the first one by simply minimizing the mean squared error between the values of the two data series. As can be seen in figure 1, it is necessary to stretch one of the two fingerprints like an elastic band before it is possible to calculate a good alignment of the two data series.

Our main aim was to develop algorithms to align two fingerprints of the same coil after one and before the following production step. Therefore, we have to deal with vertical and horizontal errors as described above. In order to be able to compare the two data series, we consider one fingerprint as fixed (we will refer to it as the "fixed coil") and transform the other coil (the "align coil") onto the same scale. The necessary stretching/compression of the align coil corresponds to a horizontal scaling. An additional difficulty is that in most production steps short pieces of the coil are cut off at the beginning and at the end, because they have been damaged or are inhomogeneous (head and tail scrap). Hence, we also have to find the starting position of the align coil within the fixed coil. This corresponds to a positive offset. So, the assumption holds that the second data series is an inner part of the first one.

Even though all described methods have been developed, adapted or/and chosen to solve the described problem for thickness profiles of steel coils, they can also be applied to other data sets. We successfully used them on width profiles of steel coils and made some promising experiments on sea shell data. Like trees, sea shells form annual "rings"[1] which can be compared in order to recognize particular environmental influences (e.g., extremely warm and cold years or the eruption of a volcano) or to reconstruct a chronology.

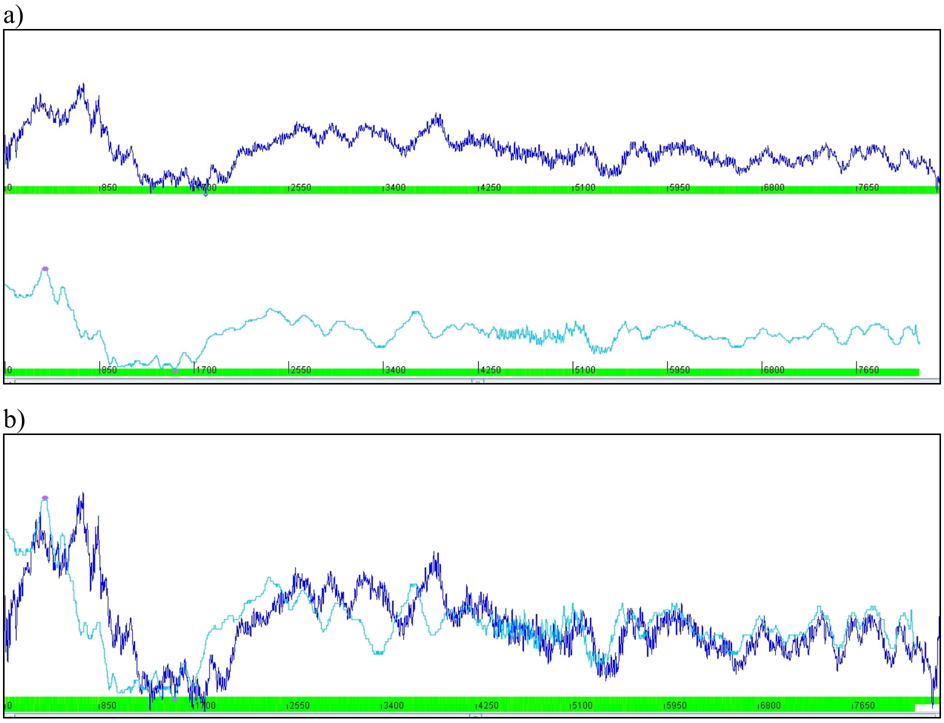

**Fig. 1.** Two fingerprints of the same coil after one and before the next production step: a) the two profiles below each other, b) the two profiles in the same window: Obviously, there are similarities but those similarities are not easy to detect automatically when comparing the two series straight away because of the horizontal stretching/compression.

In general, there are two different approaches: Either to calculate/estimate the horizontal scaling first and figure out the offset in a second step or to calculate these two parameters simultaneously. We developed methods for both variants, implemented them in Java and examined them on different real and synthetic data sets.

The rest of the paper is organized as follows: In section 2 we briefly deal with related work. In particular, we will explain some algorithms and ideas which motivated our approaches to solve the task of estimating a horizontal scaling and an offset in time series. We then formally describe the problem in section 3 and describe some aspects of the data we worked with. In section 4 we explain the different algorithms and approaches we used. The results and evaluation methodology of our

[1] In sea shells, these structures are not circular but differences in the annual rates of shell growth show up as lines with different distances and can be measured, too.

experiments are listed in section 5, before we conclude the paper in section 6 with a discussion of our findings and a look at future work.

## 2 Related Work

Our problem is a special form of time series analysis. There are similar problems in image processing, automatic speech recognition, dendrochronology and bioinformatics. We took the following approaches into account, modified and enhanced them where necessary and applied them to our data.

### 2.1 Calculation of Offset and Scaling Based on Fast Fourier Transform (FFT)

To find an alignment of two data series with the same scaling, a naïve approach is to calculate the mean squared error for each possible offset. More precisely, we place the align coil at the starting position of the fixed coil first and slide it point by point to the right afterwards. The mean squared error between the time series is calculated for each possible position. The best match comes with the least error and its starting position is the offset. The whole procedure can be accelerated by carrying out the necessary operations via FFT, which reduces the quadratic run time to $O(n \cdot \log n)$ for the calculation of the correlation (for details see section 4.1).

We can repeat this process for every possible scaling and finally pick the combination of scaling and offset which delivers the least mean squared error over all possible combinations (see section 3.1 and section 4.1).

Furthermore, it is even possible to determine the scaling factor directly by using FFT [2]. It shows up as an impulse in the frequency domain (see section 4.2).

### 2.2 Dynamic Time Warping (DTW)

Dynamic time warping [3,4] is used for pattern-recognition through comparison of two data series, e.g., for image retrieval, handwriting retrieval, speech recognition and to determine the age of a given piece of wood by comparing its annual rings' structure to the reconstructed dendrochronology of wood for thousands of years [5,6,7]. The order of the data points stays unchanged, but each value in the second sequence is associated with the "best fitting value" at an allowed position in the first one. Several points in the second data series can be mapped onto the same point within the first sequence and vice versa, as long as the order within each series remains unchanged. Another restriction is that each point of the second data series has to be mapped on a point of the first one. The entire process corresponds to finding a path in the matrix of all possible point assignments, which minimizes the squared error. This optimization problem can be solved via dynamic programming.

Run time and space of the so far described algorithm are quadratic. It is possible to adapt the Hirschberg algorithm [8] to this task so that space becomes linear by only doubling the run time. However, methods with quadratic run time are not suitable for long data series as in our steel coil example where the series can have up to 30,000 data points. Since we figured out that the horizontal scaling of our data series is always between 0.9 and 1.1, we can restrict the matrix to a corridor around the diagonal, which reduces the run time to $O(b \cdot n)$ where $b$ is the width of the corridor and thus, speed up the process a lot [5,9]. This variation of DTW is called Bounded Dynamic Time Warping (BDTW, see section 4.3).

### 2.3 Shotgun Alignment

Bioinformatics deals with alignment problems, too [10,11]. At first glance, these problems are quite different from our problem because there it is the aim to calculate an optimal or at least a good alignment of two DNA or protein sequences which are represented as strings over a finite alphabet. But, when a coil consists of parts which were welded together, our problem looks in a way similar to alignment-problems of pieces of DNA in bioinformatics. So, we had the idea to get inspiration from this field of research. Instead of calculating an alignment of two DNA or protein sequences, we have to calculate an alignment of two different but similar discrete data series. An important difference is that, in our case, we have no fixed character set but different discrete numbers/thicknesses. We can deal with this difference by comparing the interpolated thicknesses at certain positions itself instead of using weights for each possible pair of characters. A second difference is that gaps in the middle of a coil do not occur or are at least extremely rare.

An additional problem is to calculate the horizontal scaling, which can be addressed by the following idea: Venter et al. [12] were the first to use the shotgun sequencing method to figure out the DNA-sequence of the human genome. Their idea was to produce several copies of the human genome, to cut each copy of the huge human chromosomes randomly into very small pieces, to sequence these pieces and to reassemble them automatically by using the overlaps. This process was much faster than the techniques used in the public Human Genome Project and only in highly repetitive regions less accurate (see section 4.4).

## 3 The Data Series

### 3.1 Definition of the Alignment-Problem

Given two data series $X = x_0, x_1, x_2, \ldots, x_{n-1}$ and $Y = y_0, y_1, y_2, \ldots, y_{m-1}$ we want to minimize the mean squared error of the overlapping part.

The problem is that it is not possible to compare the given data points directly because of a different horizontal scaling. This scaling is assumed to be constant for each pair of data series. According to our studies, this delivers good results for the alignments of coil fingerprints. In other practical examples, we might have to cut the data series into smaller pieces and to calculate a scaling for each piece separately.

If we calculate the scaling $s$ first, keep one data series unchanged (the fixed coil) and adapt the other data series (the align coil) by linear interpolation

$$\widetilde{y}_i = y_{\lfloor s\cdot i\rfloor} + (s\cdot i - \lfloor s\cdot i\rfloor)\cdot(y_{\lfloor s\cdot i\rfloor+1} - y_{\lfloor s\cdot i\rfloor})$$

we can calculate the mean squared error for each possible starting position of the modified second data series $\widetilde{Y} = \widetilde{y}_0, \widetilde{y}_1, \widetilde{y}_2, \ldots, \widetilde{y}_{r-1}$ with $r = \lfloor s\cdot m\rfloor$ within the first data series $X$.

As explained before, in steel production a coil is at most trimmed between two production steps. So, the second fingerprint should be found completely inside the first one. Hence, we can use the following formula to calculate the mean squared error

$$MSE = \frac{1}{r}\cdot\sum_{i=0}^{r-1}(x_{t+i} - \widetilde{y}_i)^2$$

where $t \geq 0$ is the offset and $r \geq 0$ is the length of the corresponding part of $Y$ within $X$.

It is necessary to transform the second data series vertically first, so that the mean values of both data series are equal, because MSE as distance measure is influenced by different values.

### 3.2 Uniqueness of the Fingerprints

Before actually aligning thickness profiles, we wanted to make sure that such a profile is really characteristic for a certain coil/piece of steel in a sense that it satisfies the fingerprint idea. Therefore, we wanted to know, how long a piece has to be to determine its origin or – in other words – how many values we need to make a decision. Since we had only a limited set of real data, we additionally generated and examined synthetic data to get a more general idea. As we figured out, our example data series can be modeled as a damped random walk, in our case as a discrete Ornstein-Uhlenbeck-process [13]. The measuring point $x_{i+1}$ can be calculated as follows

$$x_{i+1} = \alpha\cdot x_i + z$$

where $\alpha$ is a damping factor and the random variable $z \overset{d}{=} N(0,\sigma^2)$ is normally distributed with mean 0 and standard deviation $\sigma$.

We estimated the parameters $\alpha$ and $\sigma$ of this damped random walk from our data sets to generate synthetic data series. Visually those artificial data series could not be distinguished from a real data series by experts from the *iba AG* (“Turing test”). Then, we generated random walk series with 1,000,000 values, copied pieces of a given length by chance, added some realistic noise to those pieces and tried to locate their starting position within the original data series. We figured out that the noise is normally distributed and, thus, can be modeled as white noise (with different intensities in decibel).

Our experiments showed that it is extremely unlikely to find two corresponding fingerprints by chance: Pieces of 1,500 points were located correctly in 99.8% of all test. Data series of 3,000 points or more can be considered as unique. As for the steel coils we usually have more than 8,000 data points, their thickness profiles can definitely be seen as fingerprints.

## 4 Algorithms

In this section we describe several algorithms we developed, adapted or simply applied to solve the task of aligning noisy and uniformly scaled time series.

### 4.1 Naïve Alignment (without and with FFT)

The simplest approach is to calculate the optimal position of the align coil within the fixed coil by directly minimizing the mean squared error. As we found out empirically, the horizontal scaling always varies only between 0.9 and 1.1, we try every possible scaling in between (the number is limited because our data sets are discrete, so it is sufficient to try each scaling that maps at least one point of the align coil onto a different one of the fixed coil). We then choose the parameters for scaling and offset that produced the smallest mean squared error. Obviously, this process is quadratic for each possible scaling and therefore quite time consuming, but it can be accelerated through the use of a fast Fourier transform as follows. The calculation of the mean squared error can be split into three sums

$$MSE = \frac{1}{r} \cdot \sum_{i=0}^{r-1} (x_{t+i} - \tilde{y}_i)^2 = \frac{1}{r} \cdot \left( \sum_{i=0}^{r-1} x_{t+i}^2 - 2 \cdot \sum_{i=0}^{r-1} x_{t+i} \cdot \tilde{y}_i + \sum_{i=0}^{r-1} \tilde{y}_i^{\,2} \right)$$

where the first one varies only by one value when the offset is moved one step further, the second one is the correlation and can be calculated simultaneously for all possible offsets in the frequency domain with a variation of the common FFT-based calculation of the convolution [14] and the third one is constant for all offsets.

### 4.2 Calculation of the Scaling in the Frequency Domain

It is also possible to calculate the scaling directly in the frequency domain. The idea comes from the field of image processing. We followed the course of action proposed in [2] and slightly optimized it for the one dimensional data.

Given a function $f_1(t)$ and its scaled and translated replica $f_2(t) = f_1(st + h)$, their corresponding Fourier transforms $F_1$ and $F_2$ will be related by

$$F_2(x) = \frac{1}{s} F_1(s^{-1} x) e^{-i2\pi h s^{-1} x}$$

Therefore the following relation for the magnitudes of $F_1$ and $F_2$ ($m_1$ and $m_2$ respectively) holds after converting the x-axis to the logarithmic scale

$$m_2(y) = \frac{1}{|s|} m_1(y - a)$$

where $y = \log x$ and $a = \log s$.

Thus the scaling is reduced to a translational movement and can be found by the phase correlation technique, which uses the cross-power spectrum

$$C(z) = \frac{M_1(z) \cdot M_2(z)^*}{\left|M_1(z) \cdot M_2(z)^*\right|} = e^{-i2\pi a z}$$

where $M_1$ denotes the Fourier transform of $m_1$ and $M_2^*$ denotes the complex conjugate of the Fourier transform of $m_2$.

By taking inverse Fourier transform on the cross-power spectrum, we will have an impulse, which is approximately zero everywhere except at the sought-after displacement a. After scaling the data, we can use the same phase correlation technique for finding the offset $h$.

Although the described theory can be applied to our problem, there are some subtleties which must be considered in order to get good results in practice. The problems arise because the assumption that one function is a replica of another is not entirely true due to noise and other errors. On the other hand, we use the discrete Fourier transform for concrete computation and this can be a source of further errors due to aliasing and other effects.

The following course of action seems to yield the best results:

1. Not the magnitude spectra $M_i$ but log-magnitude spectra $\log M_i$ should be used.
2. Because of the fact, that the Fourier spectrum is conjugate symmetric for real sequences, only one half of the spectrum should be used.
3. Only a small window in the middle of the logarithmic scale can be used for our purposes. This is due to the fact, that the first points on the logarithmic scale are calculated by means of the linear interpolation from only very few data points in the original lattice and thus contain not much information. On the other hand, it seems as if the higher frequencies were the consequence of the noise and do not comprise any useful information. Experiments have shown that the choice of the right window is the most crucial.

The complexity of this approach is dominated by the costs of getting the Fourier transforms and therefore is of time complexity $O(n \cdot \log n)$ when using the fast Fourier transform.

The sensitive spot of this technique is the determination of the right scaling. The noise in the data has a negative impact on the correctness of the result yielded by the algorithm. Using the FFT accelerated approach described in 4.1 to determine the horizontal scaling gives better results, but is also susceptible to extreme noise.

This technique is a good choice for time series with little noise, since the probability of an incorrect matching rises with the level of the noise.

### 4.3 Bound Dynamic Time Warping with Regression Analysis

As mentioned before, BDTW can be used to align two given coils, as well. Each point of the align coil is mapped on the best fitting point of the fixed coil while maintaining the order in both series. Then, each pair of mapped indices $(k_i \mid l_i)$, i.e., data point $x_{k_i}$ of the fixed coil has been mapped on data point $y_{l_i}$ of the align coil, is interpreted as a point in a two dimensional coordinate system. Then, the best linear approximation $f(k) = m \cdot k + b$ of all those points is calculated by regression analysis. Therefore, the squared error

$$SE = \sum_{i=0}^{p-1} (l_i - f(k_i))^2$$

is minimized which can be easily done by solving a system of linear equations.

Above, $m$ is an estimation for the horizontal scaling of the align coil, $b$ is an estimation for the offset. Figure 2 shows a mapping illustrating the alignment.

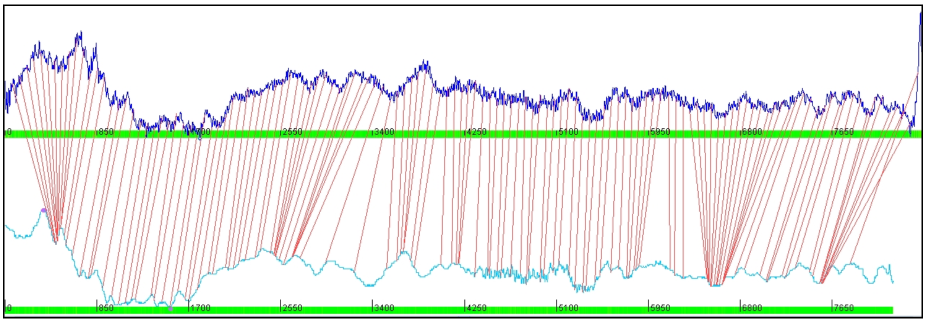

**Fig. 2.** BDTW shown at the example of two coils

### 4.4 Shotgun Analysis (without and with Regression Analysis)

The method of shotgun sequencing and reassembling the parts afterwards was invented by Celera Genomics in the Human Genome Project [12] and inspired us for another approach: The shorter the align coil, the less impact has the usually small horizontal scaling on the alignment. Hence, our idea was to cut the align coil into pieces of a certain length $m_s$ (e.g., of 50 meters = 500 values) and to locate these snippets within the fixed coil by a normal naïve alignment where no horizontal scaling is taken into account. Figure 3 gives an impression how the method works.

The median of the differences between the calculated starting positions of every pair of subsequent snippets divided by the length of the snippets can be used as horizontal scaling between the two data series. After an interpolation step as described in 3.1 the offset of the whole align coil can be calculated naïvely by minimizing the mean squared error.

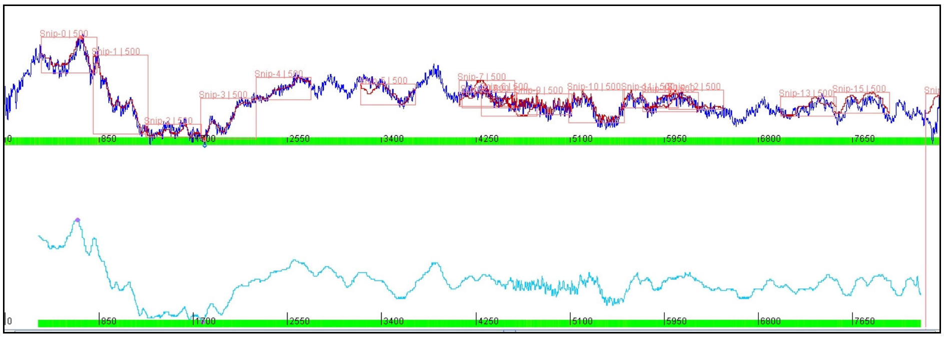

**Fig. 3.** The “best” positions of the different snippets of the second coil within the first one are shown by boxes. As you can see, the first four snippets are positioned extremely good, while the positions of some other snippets are inconsistent.

The naïve alignment at the end is not necessary. It is possible to calculate scaling and offset simultaneously by linear regression: The expected and the found starting position of each snippet is interpreted as a point in a two dimensional coordinate system ( $k_i$ is the expected starting position, $l_i$ is the found starting position). Again the best linear approximation $f(k) = m \cdot k + b$ of all those points is calculated, where $m$ is an estimation for the horizontal scaling of the align coil and $b$ is an estimation for the offset. The computation time is quadratic or more precisely $O(c \cdot m_s \cdot n)$ where $c$ is the number of the snippets and $m_s$ is the length of each snippet.

One problem is that the order of the pieces can be inconsistent, especially when a piece seems to fit well on different positions inside the fixed coil, what can be true for quite a number of them. To avoid this, we used only characteristic snippets to construct the regression line. A snippet is assumed to be characteristic, if the error at its best fitting position is considerably lower than at any other position.

Obviously, this is a heuristic approach. On high quality data, it might be less accurate than the algorithms described before (even though it still produces good results that cannot be distinguished from the other results visually in most cases). Its main advantage is that it is very tolerant to noise and local outliers. This occurs quite often in practice because of measurement failures and/or dirt on a coil. Extremely bad fitting snippets can simply be ignored.

### 4.5 Alternative Quality Criteria

Instead of the mean squared error two other quality criteria can be used, the (ordinary) correlation or the coefficient of parallel variation[2] [15]. The latter can be seen as a signum function on the differences between data points. An advantage of the correlation is that it implies a normalization of the data. It is tolerant to different means and to a vertical scaling of the data. The coefficient of parallel variation is even more tole-

[2] In related literature, we also encountered the German technical term „Gleichläufigkeitskoeffizient“ quite often.

rant to noise, as it merely considers the direction of a change what makes it especially helpful for the naïve approach on bad data. It can also be calculated via our FFT algorithm.

# 5 Experiments

To analyze the quality of the different algorithms which we described in section 4, we ran several experiments. They had different aims, provided insights in different aspects of the algorithms' performance and reflected different tasks in our application scenario of steel coil tracking.

The time complexity of all algorithms has been explained above. In practice, they all need 0.5 to 3 seconds to construct an alignment of two data series with 5,000 to 8,000 measuring points. It is hard to compare their run time exactly because it depends on length and structure of the data series.

## 5.1 Determination of the Horizontal Scaling

Since the horizontal scaling was unknown for all practical examples, we used synthetic data to show that the developed methods are able to determine it correctly. Therefore, we created "coil data" with head and tail scrap artificially: We first created 1,000 time series modeled by a damped random walk (the "fixed coils"). Then, we scaled and trimmed a copy of these data and added some white noise to produce corresponding "align coils". Since, for these data, we know the scaling (and the offset), they are perfectly suitable for our tests (a gold standard). The average difference between real and calculated scaling was less then $5 \cdot 10^{-4}$ for all our methods, which means that all methods have an average accuracy of more than three decimal places.

## 5.2 Recognition of Head and Tail Scrap

The intention of this experiment was to analyze the capability of the algorithms to estimate the parameters for offset and scaling. If head and tail scrap are calculated correctly, also the scaling must have been determined correctly. This is clear because our methods calculate only scaling and offset. The ending position is computed from these two parameters afterwards.

We used different data series for this test. On the one hand, we used real world data of coil thickness profiles recorded during steel production at the end of the first and before the second production step. On the other hand, we used again synthetic data.

### 5.2.1 Real World Data

As mentioned before, the real world data is noisy under several aspects: First of all, the devices measuring the thickness are based on different technologies and, therefore, have a different resolution and accuracy. Second, the speed of the steel strip while passing through the production plant is measured indirectly and at a different point than the thickness. This setup causes the observed differences in the horizontal scaling in the data. In the beginning of the research project the steel mill *voestalpine AG* in Linz, Austria provided us with information on 20 coils. We had two data series for each coil, one measured after the hot rolling mill and one right before the next production step, the cold rolling mill. In addition, we knew the real length of the coil along with the amount of steel cut off at the beginning and at the end of the coil before the second production step (head and tail scrap). This information should have been sufficient to use it as a gold standard to compare our calculated results – assumed the data were precise enough. Unfortunately, the production conditions did not allow a correct measurement of the parts that had been cut off, so the values given to us finally turned out to be only rough estimates. A visual analysis of the alignments revealed that our methods delivered far more accurate results. Such an alignment is shown in figure 4.

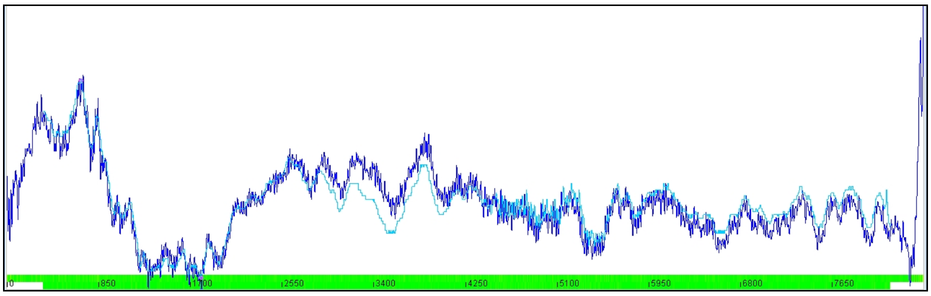

**Fig. 4.** An alignment of two fingerprints of the coil already shown in figure 1

### 5.2.2 Artificial Data

To run tests on a larger scale and to be able to evaluate the results, we again used the artificial data described in 5.1. Apart from white noise, scaling and trimming of a coil at the beginning and the end we observed other, more rare and particular noise in real world coil data. The devices to measure and record the thickness might temporarily fail. In this case the data series contains a long period (several hundred data points) of zero values. Another problem is dirt, pieces of steel or holes in the coil. They cause extremely short (1 to 5 data points) peaks in the time series. This kind of noise is difficult to handle.

To analyze how robust the algorithms are towards these phenomena we created further artificial data which contained also this kind of noise. The results of our test on synthetic data are shown in table 1 and 2.

**Table 1.** Accuracy of the different methods using MSE as quality criterion: Fraction of correctly aligned data series: An alignment was classified as correct when the real and the calculated starting and ending position of the second data series within the first one where no more than 5 points away from each other (equal to half a meter).

| method | normal | with 3 peaks | with 3 zero lines | with both |
|---|---|---|---|---|
| Alignment with FFT | 1.0 | 0.659 | 1.0 | 0.634 |
| BDTW with Regression | 1.0 | 0.117 | 0.546 | 0.107 |
| Shotgun Analysis | 0.952 | 0.953 | 0.96 | 0.945 |

Table 1 shows e.g., that in all "simple" cases the second data series was aligned correctly to the first one with the FFT-based approach described in section 4.1 and with the BDTW method described in 4.3, whereas the Shotgun Analysis described in 4.4 achieves comparably better results on data series with errors (e.g., still 94.5% correct alignments of all data series with zero lines and peaks).

**Table 2.** Accuracy of the different methods using MSE as quality criterion: Average difference (in data points) between real and found starting and ending position of the second data series within the first.

| Method | normal | with 3 peaks | with 3 zero lines | with both |
|---|---|---|---|---|
| Alignment with FFT | 0.51\|0.20 | 17.95\|17.89 | 0.51\|0.19 | 19.73\|19.65 |
| BDTW with Regression | 1.62\|1.15 | 119.85\|119.88 | 7.79\|7.72 | 115.20\|115.26 |
| Shotgun Analysis | 1.64\|1.55 | 1.72\|1.64 | 1.72\|1.61 | 6.60\|6.54 |

In table 2, you can see the average difference of "real" and calculated head and tail scrap. The first number stands for the average difference in head scrap, the second one for the average difference in tail scrap. Here, it becomes clear that the Shotgun Analysis is only slightly less accurate than for example BTDW with Regression on good data. The result in table 1 seems worse because there, an alignment is categorised as wrong if its starting or ending position differs 6 or 7 instead of the allowed maximal 5 points from the "real" starting or ending position. This difference can hardly be noticed by visual inspection.

### 5.3 Searching a Database for Fitting Pairs of Fingerprints

The last test comprised again the fingerprint idea. For this analysis, we were provided with data of about 1,000 hot rolled strips and 191 cold rolled strips. The latter ones were the values of 191 coils among the 1,000 whose thickness was measured again at the beginning of the next production step. Here, we had a gold standard from the steel producers database, providing unique numerical identifiers for the coils. Our task was to identify the 191 among the 1000. Even with the simplest approach described in 4.1, we achieved the very high accuracy of 98%, that means, given the fingerprint of a coil from the beginning of the second production step, we where able to identify the corresponding one from the end of the first production step in most cases.

To accelerate the process and as for this matching task we do not need highly precise alignments, it is possible to compress the data by a method called piecewise aggregate approximation (PAA). Here, e.g. 30 consecutive points are mapped onto one new point by calculating their mean value. This way one gets a compression factor of 30. This gives a considerable speed-up since many candidate coils can be excluded fast. On the compressed data, we got an even higher accuracy of 100% because small inaccuracies are smoothened through the averaging. In this way, it is possible to identify the corresponding hot rolled strip to a given cold rolled strip within 1,000 candidates in less than 5 seconds.

# 6 Conclusions and Future Work

As illustrated above, the different methods have different strengths and weaknesses: Exact methods are more precise on good data sets but less robust to noise and errors. Therefore, for the practical use, it would be a good idea to have an automated choice of the alignment method based on the peculiarities of the involved data series. Another idea is to manipulate the data in a way that extreme outliers are “smoothed” before the alignment. We already made some experiments and got promising results but have not yet tested them systematically.

In some (exceptional) case, there can be gaps within a data series (e.g., because an erroneous part has been cut out). It is already possible to deal with those cases by cutting such a data series into two pieces and then aligning each piece. The shotgun method is in principle able to ignore the missing parts and to calculate the scaling only for the segments that can be found in both data series. An additional task would be to detect such gaps automatically and to calculate two different offsets and scalings for the two parts.

So far, our methods are restricted to a linear/constant horizontal scaling, which might not be true in all practical examples. Some of our methods (especially the ideas described in 4.3 and 4.4) have the potential to overcome this limitation, e.g., by replacing the linear regression by more sophisticated methods.

An alternative in practice is to compute alignments based on width profiles instead of thickness profiles which is also possible with the described methods. Besides, the described methods can basically be used to align any two data series of numerical values. Another practical example we are dealing with is the alignment of annual sea shell growth measurements to obtain a chronology as described above (see section 1). Therefore, it would be good to allow the alignment of more than two data series.

**Acknowledgements**

We want to thank the *voestalpine AG* in Linz, Austria for the allocation of real world fingerprint data from steel coils and the *iba AG* in Fürth, Germany for good and interesting cooperation.

**Remarks**

A selection of the methods described in this paper has been integrated into the freely available tool iba-Analyzer which can be found at the website of the *iba AG* at http://www.iba-ag.com/.

## References

[1] Anhaus, H.: Verfahren und Vorrichtung zur Identifizierung eines Teilstücks eines Halbzeugs, Patentblatt DE102006006733B3 (23.08.2007)

[2] Reddy, B.S. and Chatterji, B.N.: An FFT-Based Technique for Translation, Rotation, and Scale-Invariant Image Registration, IEEE Transaction on Image Processing, Vol. 5, No. 8 (1987)

[3] Toyoda, M., Sakurai, Y. and Ichikawa, T.: Identifying Similar Subsequences in Data Streams, DEXA, LNCS 5181: 210–224 (2008)

[4] Chu, S., Keogh, E., Hart, D. and Pazzani, M.: Iterative Deepening Dynamic Time Warping for Time Series, Proceedings of the Second SIAN International Conference on Data Mining (2002)

[5] Ratanamahatana, C.A. and Keogh, E.: Everything you know about Dynamic Time Warping is Wrong, 3rd Workshop on Mining Temporal and Sequential Data, in conjunction with 10th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (KDD-2004), Seattle, WA (2004)

[6] Euachongprasit, W. and Ratanamahatana, C.A.: Efficient Multimedia Time Series Data Retrieval Under Uniform Scaling and Normalisation, ECIR, 506-513 (2008)

[7] Wenk, C.: Algorithmen für das Crossdating in der Dendrochronologie, diploma thesis, Freie Universität Berlin (1997)

[8] Hirschberg, D. S.: A Linear Space Algorithm for Computing Maximal Common Subsequences, Commun. ACM, 18(6): 341-343, ISSN 0001-0782 (1975)

[9] S. Salvador P. Chan: Fast DTW: Toward Accurate Dynamic Time Warping in Linear Time and Space, Intelligent Data Analysis, 11(5): 561-580 (2007)

[10] Needleman, S.B. and Wunsch, C.D.: A general method applicable to the search for similarities in the amino acid sequence of two proteins, Journal of Molecular Biology, 48:443–453 (1970)

[11] Smith, T. F. and Waterman, M.S.: Identification of common molecular subsequence, J. Mol. Biol, 147: 195-197 (1981)

[12] Venter, L.C. et al: The Sequence of the Human Genome, Science, 291: 1304-1351 (2001)

[13] Uhlenbeck, G.E. and Ornstein, L.S.: On the theory of Brownian Motion, Phys.Rev. 36:823–41 (1930)

[14] Vetterling, W.T., Teukolsky, S.A., Press, W.A., Flannery, B.P.: Numerical Recipes in C, Second Edition, Cambridge Univ. Press, Cambridge (1999)

[15] Kemeter, M.: Effizientes Alignment von Stahlband-Fingerprints, diploma thesis, Johannes Gutenberg-Universität Mainz (2008)