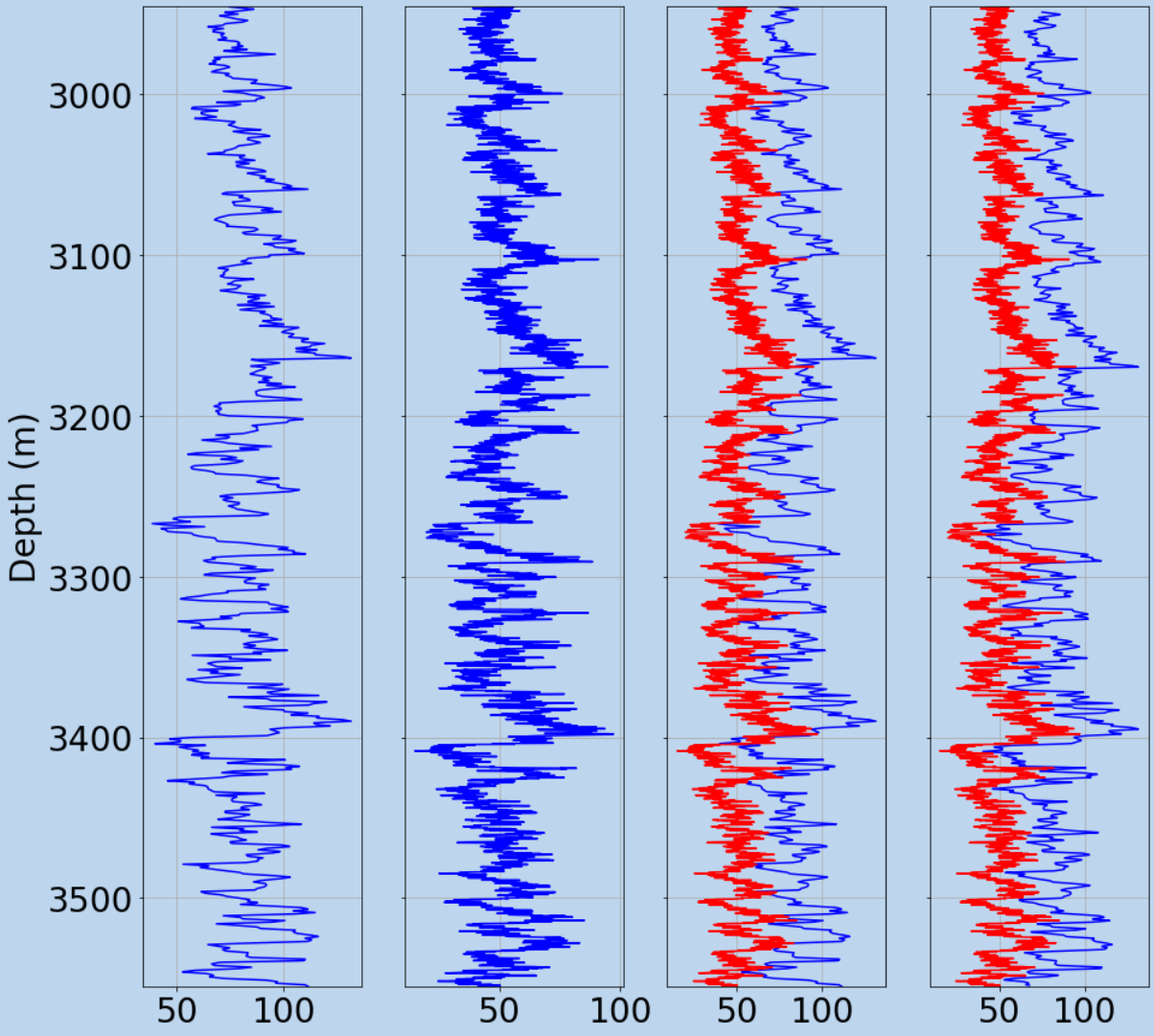
Automatic Depth Matching for Petrophysical Borehole Logs



Aitor García Manso

AUTOMATIC DEPTH MATCHING FOR PETROPHYSICAL BOREHOLE LOGS

THESIS

submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE

in

ELECTRICAL ENGINEERING

by

Aitor García Manso
born in Zumarraga, Spain

This work was performed in:

Circuits and Systems Group
Delft University of Technology
and
Shell Global Solutions International B.V.
Digitalisation department

Delft University of Technology

DELFT UNIVERSITY OF TECHNOLOGY
DEPARTMENT OF
MICROELECTRONICS

The undersigned hereby certify that they have read and recommend to the Faculty of Electrical Engineering, Mathematics and Computer Science for acceptance a thesis entitled **"Automatic Depth Matching for Petrophysical Borehole Logs"** by **Aitor García Manso** in partial fulfillment of the requirements for the degree of **Master of Science**.

Dated: 28th August 2020

Chairman: _____
prof. dr. ir. G. Leus

Advisors: _____
dr. J. Przybysz-Jarnut

_____
dr. W. Epping

Committee member: _____
dr. ir. E. Isufi

ABSTRACT

In the oil and gas industry a crucial step for detecting and developing natural resources is to drill wells and measure miscellaneous properties along the well depth. These measurements are used to understand the rock and hydrocarbon properties and support oil/gas field development. The measurements are done at multiple times and using different tools. This introduces multiple disturbances which are not related to physical properties of rocks or fluids themselves, and should be tackled before data is used to build subsurface models or take decisions. One important source of this disturbances is depth misalignment and in order to compare different measurements care must be taken to ensure that all measurements (log curves) are properly positioned in depth. This process is called depth matching. In spite of multiple attempts for automating this process it is still mostly done manually. This thesis addresses the automation problem and proposing a model based approach to solve it using Parametric Time Warping (PTW).

Based on the PTW, a parameterised warping function that warps one of the curves is assumed and its parameters are determined by solving an optimization problem maximizing the cross-correlation between the two curves. The warping function is assumed to have the parametric form of a piecewise linear function in order to accommodate the linear shifts that take place during the measurement process. This method, combined with preprocessing techniques such as an offset correction and low pass filtering, gives a robust solution and can correctly align the most commonly accruing examples. Furthermore, the methodology is extended to depth match logs with severe distortion by applying the technique in an iterative fashion. Several examples are given when developed algorithm is tested on real log data supplemented with the analysis of the computational complexity this method has and the scalability to larger data sets.

ACKNOWLEDGEMENTS

First, I would like to express my gratitude to all the people that have helped me and supported me during the time I have been working on this thesis under such adverse circumstances.

It was in September when Dr. Richard Heusdens offered me this thesis proposal in collaboration with Shell Global Solutions International B.V. I was excited by the idea and I immediately accepted.

The first person I want to thank is Dr. Justyna Przybysz-Jarnut who was my daily supervisor and helped me better understand the problem as well as to solve the doubts I had during the project. I would also want to thank Dr. Willem Epping whose experience in the field was crucial to evaluate the reliability of the designed methods.

Regarding TU Delft, I would like to thank the whole CAS department with whom I conducted the thesis and in particular Prof. Geert Leus for his acute comments which guided the thesis by showing the flaws of the algorithm and encouraging solutions and new research directions.

Finally, I would like to thank all the people I have met during these two years I have been in TU Delft. Their support has helped me endure in the worst moments and they have given me some of the best experiences in my lifetime so far. I will never forget my enriching experiences acquired during this time and I will take them from now onward.

Aitor García Manso
Delft, The Netherlands
28 August 2020

CONTENTS

LIST OF FIGURES

1 INTRODUCTION

The increase in and availability of the computing power, makes it now possible to solve complex problems in a relatively short time and started a new trend seen across industries, where repetitive processes are automated, making them more efficient, consistent and less error prone. Innovations such as machine learning or cloud computing are examples of the attempts to use and further develop Artificial Intelligence to solve multivariate and complex problems.

Digitalisation is also a very important theme in the oil and gas industry, where companies put a lot of focus to deploy artificial intelligence technologies across their businesses to eliminate laborious and repetitive tasks, to improve efficiency and consistency and accelerate decision making. The benefits of digitalisation can be realized in the petrophysics domain where most of the current data handling and interpretation workflows are still done manually. One of such potential applications is depth matching of borehole logs. Automating that task is a subject of research for a number of years [1; 2; 3; 4].

When recording data along a drilled borehole, several problems may manifest themselves resulting in depth mismatches between data recorded at different times [5]. In the process of acquiring petrophysical data, an instrument is introduced underground and that measures different responses per depth resulting in a one dimensional curve per measured response. However, this process is in general non-repeatable. If a different instrument is used or the measurement is taken at another time, depth shits may occur, and need to be addressed first before recorded response can be attributed to a certain depth.

There have been several attempts to tackle this problem. The most basic was using correlation coefficients [2]. Another attempt was to apply the Dynamic Time Warping (DTW) technique [3; 4]. However, both methods have their drawbacks, and depth matching of borehole logs remains an active research field. Recently, a novel technique using machine learning was used with promising results [6].

In essence the depth matching of borehole logs can be abstracted as a warping problem of a one dimensional curve [2]. This has been tried in a wide field of applications (data mining, biology, speech recognition, etc.) [7; 8] and several developed methods could be adapted to the particular problem of depth matching. Currently, the most widely used is the DTW [9] which has several variations depending on the properties of the curves to warp. Another method gaining interest is the Parametric Time Warping (PTW) [10], where a parametric warping function, usually a polynomial, is defined and its values are estimated by solving an optimization problem.

The main goal of this thesis is to investigate the applicability of existing warping methods to solve depth matching problem. The project is in collaboration with Shell Global Solutions International B.V who provided data to judge develop and test the methodology.

1.1 RESEARCH STATEMENT

The research done for this thesis addresses the following question:

- Can two borehole log measurements which have been obtained using different tools at different times be automatically depth matched?

1.2 THESIS OUTLINE

This thesis is structured as follows. In chapter 2 an introduction to petrophysics is given as well as a description of the depth matching process. Afterwards in chapter 3 a general overview of available warping techniques is presented. Chapter 4 addresses how the Parametric Time Warping (PTW) can be adapted to the current case, a piecewise linear parametrization is introduced and the properties of such technique are investigated. In chapter 5 some preprocessing techniques are introduced considering the properties of the logs to improve the optimisation results. Chapter 6 introduces an iterative version of the PTW and tests its performance with data where severe distortion is present. Finally, chapter 7 gives the conclusions and suggests future research directions.

2 PROBLEM STATEMENT

2.1 PETROPHYSICS

One of the crucial elements for resources development in the oil and gas industry is the detection of hydrocarbon deposits and its evaluation in terms of potential exploitation. Multiple techniques are nowadays available one of them being the borehole log analysis. In search for underground hydrocarbon resources, first a seismic measurement is carried out using wavefield imaging either with electromagnetic or sound waves [11]. The resolution of the measurement is on the order of 10 meters, so it is hard to guarantee with certainty the presence of any hydrocarbon. Therefore, this technique can just give a rough idea whether oil can be present at the location but it is by no means conclusive.

If the seismic shows a high chance of finding accumulation of oil or gas, an exploration well is drilled to confirm the presence of hydrocarbons [12]. If the prospect is economic, additional wells are drilled to produce hydrocarbons. Each drilled well is logged (logging is a process of measuring certain property, such as gamma ray or neutron response at different depths to obtain a one dimensional curve that maps the property versus depth). The most basic way is to insert a probe on the bottom of the hole and steadily raise it while taking measurements at different depths.

There are a wide variety of measurements that can be recorded while logging, the most basic is the gamma ray measurement as it can clearly differentiate between sands (potentially containing hydrocarbon) and shales by measuring natural gamma radiation [13]. There are other measurements taken in the same manner such as the density, resistivity or neutron-porosity measurements [12]. Furthermore, once the borehole is drilled, some physical samples can be taken to analyse them in the laboratory. Combining all the data an assessment can be made whether or not oil or gas are present in the location and if it is economical to develop given resource.

The borehole is drilled in several steps [14]. First a section is drilled with a given diameter. Afterwards the measurements are taken and later the hole is secured by putting a cement casing around it and filling the hole with drilling mud (a liquid dense enough to keep the pressure balanced, usually water). Once the section is secured the next section of the hole can be drilled using a smaller diameter than in the previous step. This procedure is repeated a couple of times, each drilled section is called a run, so the whole hole consists of a number of runs. From each run a series of measurements are obtained and plotted as a function of the depth, called logs.

However, since the real position of the probe is not known with full accuracy some parts of the run might overlap with the previous or following runs. As some sections are cased, the values may differ, and thus the runs cannot be blindly concatenated to form the whole borehole log [2].

Furthermore, during the measurement multiple disturbances take place which affect the outcome [5]. The probe is connected with a metal cable and pulled up using a machine [15]. Therefore, the probe is not raised with constant speed but with sudden moves. Additionally, since the size of the probe is not much smaller than the hole itself, often, mainly in the deepest runs, the probe can touch the wall and be momentarily stuck or suffer from friction. On the other hand, the mud is also a source of friction and all these disturbances create a distortion in the measurement [5]. The measurements themselves might be still correct but the recorded depth is stretched or squeezed as compared to the actual depth and needs to be corrected.

Another technique to extract information from the boreholes is the Logging While Drilling (LWD) [16]. As the name suggests, the LWD is a technique where the driller is equipped with some measuring tools, that probe the sediments while the drilling is taking place. This process has some advantages and disadvantages. It is more time efficient as some preliminary conclusions can be obtained directly after drilling. Besides, as the drill bit is in touch with the sediments the measurements can be more stable than when the probe is hanging. However, due to the vibrations other errors can appear resulting in noisier measurements. Furthermore, there is a trade-off between the drilling speed and the measurement quality, because the optimal drilling would be as fast as possible, while the measurements need to take samples at certain rate. Therefore, the sampling rate in the LWD is not as flexible as in the wireline logging and it is usually lower.

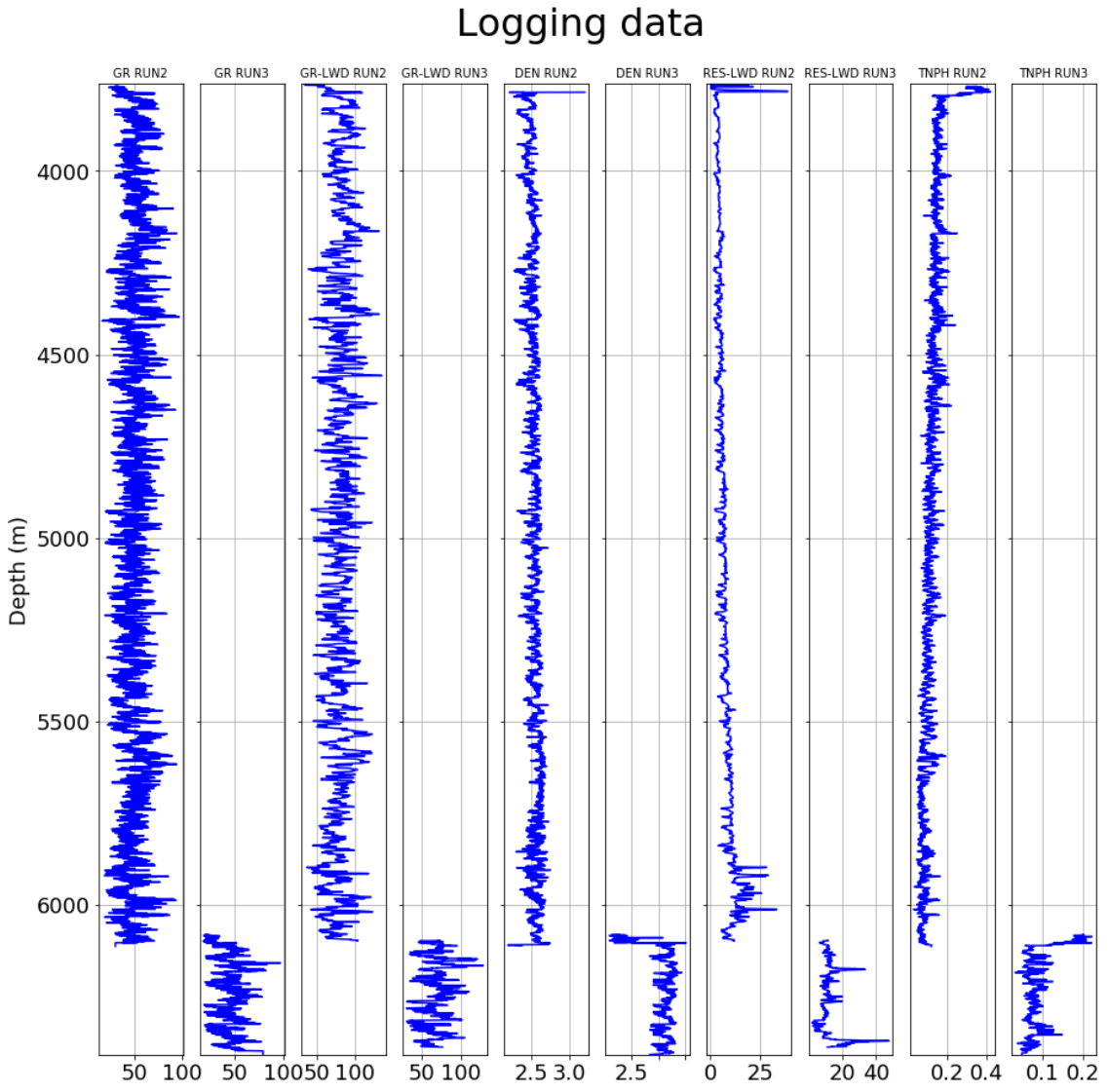


Figure 2.1: Example of the data obtained from a well log. In the vertical axis the depth at which the measurement was taken is shown and in each horizontal axis measurement values are displayed. From left to right the gamma ray (GR RUN2, GR RUN3), the gamma ray using LWD (GR-LWD RUN2, GR-LWD RUN3), the density (DEN RUN2, DEN RUN3), the resistivity (RES-LWD RUN2, RES-LWD RUN3) and the neutron measurements (TNPH RUN2, TNPH RUN3) are shown.

In figure 2.1 an example of the data obtained from a well is presented. The vertical axis is the depth (increasing from shallow to deep) and the horizontal axis is the measurement value. There are multiple measurements that can be performed at the same time, in this case gamma ray, density, resistivity and neutron-porosity measurements are shown.

Gamma ray (GR-LWD) and the resistivity (RES-LWD) measurements were done using the LWD technique and both were taken simultaneously, while the others were also taken simultaneously but using the wireline logging. Two runs are shown in this example highlighting the main issues, for instance, the LWD measurements have a smaller sampling rate and the different tools have different offsets, so the length and starting point of the logs can be different.

The petrophysicists can extract information from the logs, but in order to do it properly they need to align the curves first, so they must ensure that the depths of the wireline logging and the LWD are the same [13]. This process is called depth matching: the two gamma rays recorded with different tools are aligned and this result is used for adjustments in other curves. The alignment is done by finding and matching similar patterns and features.

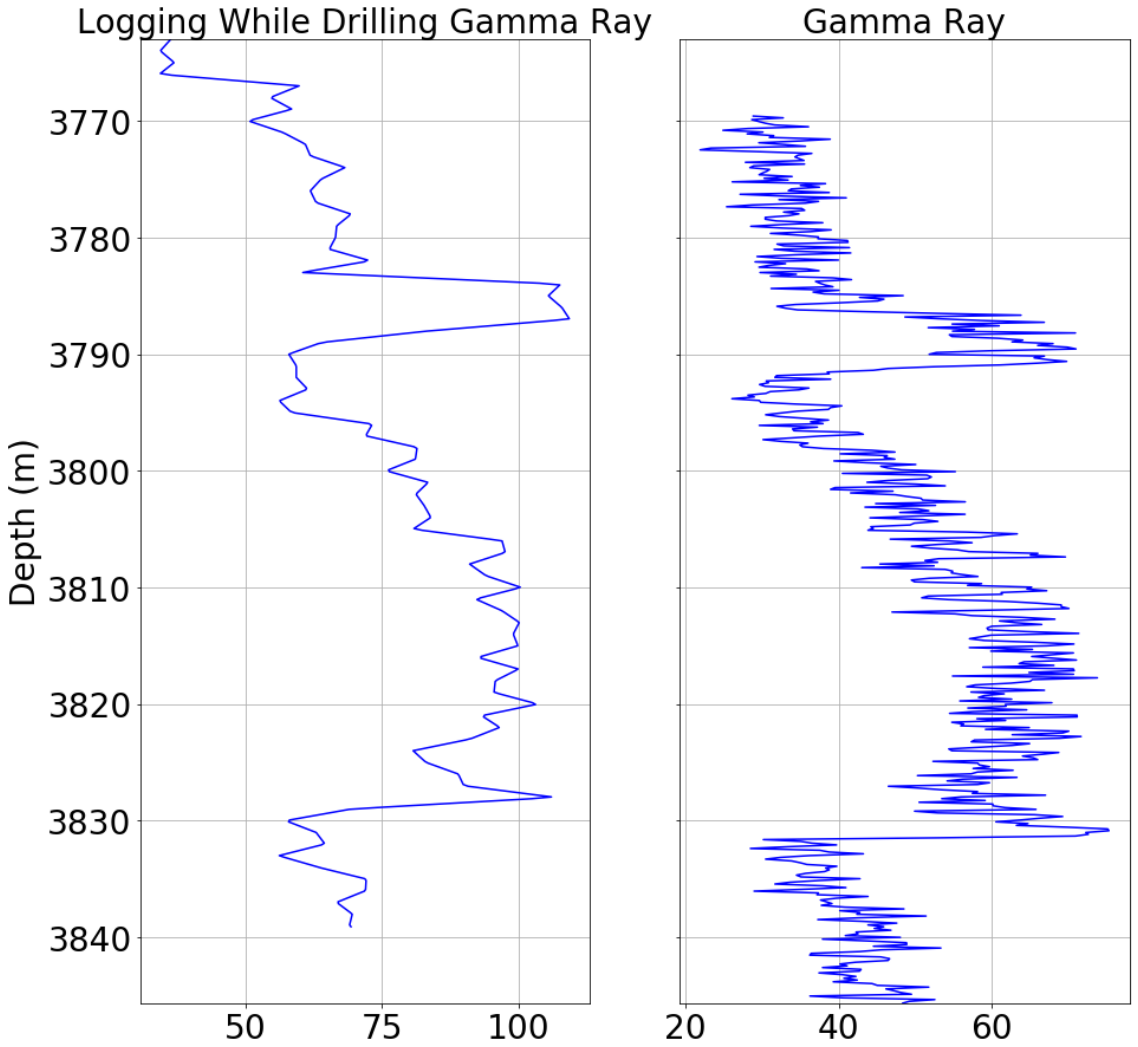


Figure 2.2: Example of two gamma ray logs, the the left one was obtained using Logging While Drilling (LWD) and the right one was obtained with a wireline logging.

2.2 DEPTH MATCHING PROBLEM

In figure 2.2 an example of two gamma ray logs are shown. These logs have a similar shape, but they are not aligned at the same depth. Also, it is important to notice the difference in their amplitude, which may be due to differences in the tools used, calibration, etc. A suitable method for alignment should focus on the shapes and patterns rather than amplitudes.

In spite of the numerous attempts to automate the process of depth alignment between logs, it is still mostly done manually, with a petrophysicist stretching and squeezing one dimensional curve against the other to attempt the match as shown in figure 2.3. The method is relatively simple, only one curve, the query, usually that with the lowest sampling rate, is stretched and squeezed until it matches the other, the reference. A petrophysicist selects a number of points in the original query and they are displaced to the depth where they appear in the other curve [2]. This is usually done using an existing petrophysical software, one of the most common being Techlog, where each handpicked point will represent the boundary of an interval and when a point is displaced the plot is adjusted by stretching or squeezing the neighbouring intervals.

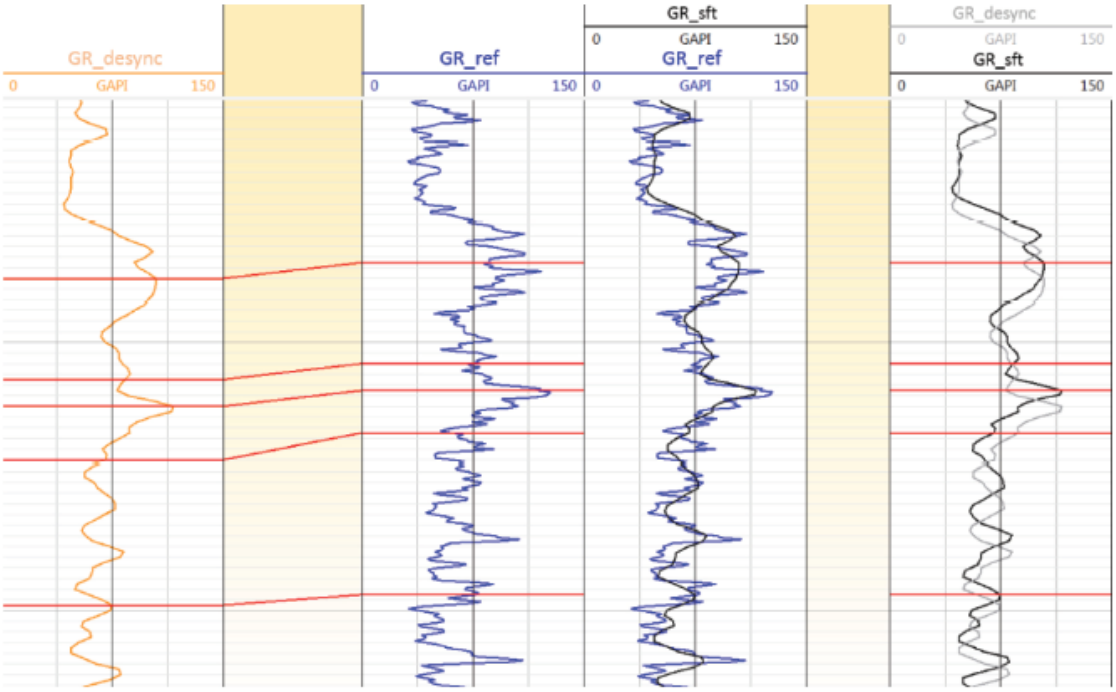


Figure 2.3: Manual depth matching example taken from [6]. From left to right first the gamma ray measurement obtained with Logging While Drilling and secondly the gamma ray measurement using the wireline technique are plotted. Thirdly the depth matched result is shown and finally the alignment without depth matching the logs.

2.3 MATHEMATICAL DESCRIPTION OF THE PROBLEM

In order to tackle the depth matching problem, the input data can be expressed mathematically. The query and the reference can be considered a continuous function $s(t)$ and $r(t)$ respectively which depend on the variable depth, t. Nonetheless, both curves are misaligned, which can be expressed using the warping function, $w(t)$. The simplest misalignment is an offset, for instance, of 10 meters; therefore, the warping function would be $w(t) = t - 10$ and the relation between

the query and the reference would be $r(t) = s(w(t)) = s(t-10)$. In the real application; however, the warping function is not known so it must be estimated.

In petrophysical applications, both query and reference are discretised into a set of noisy samples, and continuous functions are not available. Each can be expressed using a vector, $\mathbf{s}$ whichdenotes the data points from the query while $\mathbf{r}$ refers to the reference. The measurements are taken at certain depth, t_i. In order to avoid misunderstandings, the depth instances for the query will be denoted as $t_{s,i}$ while those for the reference will be denoted as $t_{r,i}$. Therefore, the input reference and query would be expressed respectively as

$$\mathbf{r} = r(\mathbf{t}_r) \equiv \begin{pmatrix} r(t_{r,1}) \\ \vdots \\ r(t_{r,N}) \end{pmatrix} \quad \text{and} \quad \mathbf{s} = s(\mathbf{t}_s) \equiv \begin{pmatrix} s(t_{s,1}) \\ \vdots \\ s(t_{s,M}) \end{pmatrix}. \tag{2.1}$$

In consequence, the problem can be defined as finding or estimating the warping function, $w(t)$, given a discrete number of data points.

3 OVERVIEW OF WARPING METHODS

In this chapter some of the existing methods used for curve alignment are discussed. It is important to note that there is a wide variety of warping methods, each with its own advantages and disadvantages. The suitability of those methods depends on the problem and the properties of the data.

3.1 MANUAL APPROACH

Currently most of the depth matching is still done manually, by visual inspection, as shown in figure 2.3. One of the curves, usually the one with the lowest sampling rate is used as a query and the other curve is used as a reference. Petrophysicist select a number of anchor points on the reference and the corresponding anchor points on the query. The query signal between anchor points are then squeezed or stretched to match the signal between anchor points in the reference. This can be explained in mathematical terms as follows: two set of depths of n samples are chosen

$$\mathbf{t}_{r,Manual} \equiv \begin{pmatrix} t_{r,Manual,1} \\ \vdots \\ t_{r,Manual,n} \end{pmatrix} \quad \text{and} \quad \mathbf{t}_{s,Manual} \equiv \begin{pmatrix} t_{s,Manual,1} \\ \vdots \\ t_{s,Manual,n} \end{pmatrix}, \tag{3.1}$$

where $t_{r,Manual,i} \in [t_{r,1}, t_{r,N}]$ and $t_{s,Manual,i} \in [t_{s,1}, t_{s,N}]$. The warped query is built using this information, so for the depth indexes $\mathbf{t}_{r,Manual}$ the warped query has the value $s\left(\mathbf{t}_{s,Manual}\right)$. However, only n points are directly assigned, any other point of the curve must be interpolated. The simplest is to use a linear interpolation, resulting in the stretch or squeeze of the intervals the manual points delimit. Although there are more advanced interpolation techniques that try to preserve the peaks and valleys, for instance using Fourier transformations, the most commonly used methods are still based on linear interpolations [17].

3.2 CORRELATION COEFFICIENTS

The first attempts to automate depth matching were done using correlation coefficients [2]. There are several ways of measuring the correlation between two data sets, being the most common the Pearson correlation coefficient [18]. The way to compute it between a random pair of variables (X, Y) with mean (μ_X, μ_Y) is

$$\rho_{X,Y} = \frac{cov(X,Y)}{\sigma_X \sigma_Y} = \frac{E[(X-\mu_X)(Y-\mu_Y)]}{\sigma_X \sigma_Y}. \tag{3.2}$$

If these values are not known but n sample points or realisations of each variable are available the correlation can be computed with

$$\rho_{X,Y} = \frac{cov(X,Y)}{\sigma_X \sigma_Y} = \frac{\sum_{i=1}^{n}(x_i-\mu_X)(y_i-\mu_Y)}{\sqrt{\sum_{i=1}^{n}(x_i-\mu_X)^2}\sqrt{\sum_{i=1}^{n}(y_i-\mu_Y)^2}}. \tag{3.3}$$

Thus, the algorithms are based on selecting an n sample window around a central point and calculating the correlation or certain cost function in the window for that specific point. Afterwards, some variations are applied to this initial window, for instance, adding an offset shift or scaling the window and the combination that gives the maximum correlation is chosen. Moreover, this must be done for all the points of the curve in order to obtain the optimum warping. However, considering numerous possibilities requires a big number of windows, which makes the implementation tricky. If all of them were tried and the best option were chosen the computing time would be prohibitively high. The proposed solution was to use dynamic time warping to compute it efficiently as in [19; 20] and obtain the optimum warping.

3.3 DYNAMIC TIME WARPING (DTW)

Most of the attempts to automate depth matching were based on dynamic programming, so it is worth investigating Dynamic Time Warping (DTW) [21], which is the most widely applied algorithm for comparing different time series. It was originally created for speech recognition [22] but it was extended to multiple applications for time series data [23; 24].

Suppose that two time series one, the query, s, with M elements and the other, the reference, r, with N elements are to be compared between each other. The DTW will modify the query in order to match it as good as possible to the reference, which does not undergo any changes. The algorithm matches each point from the query with another from the reference while preserving monotonicity [25]. In order to solve the problem using dynamic programming a local cost measure between two points from the query and the reference is defined, $c(s_i, r_j)$, which is small when both points are similar. This value is usually the Manhattan distance between both points although other measures can be used [26]. All these cost measures can be assembled together into an $M \times N$ matrix where each element is given by $C(i,j) = c(s_i, r_j)$. This matrix is illustrated in figure 3.1a. The DTW computes the path by matching points from the query to the reference [9], which can be expressed as an $L \times 2$ matrix $\mathbf{L}$, where each line gives the matched pair of points, so

$$\mathbf{L} \equiv \begin{pmatrix} L_1 \\ \vdots \\ L_L \end{pmatrix} \equiv \begin{pmatrix} \mathbf{s}_L & \mathbf{r}_L \end{pmatrix} \equiv \begin{pmatrix} s_{L,1} & r_{L,1} \\ \vdots & \vdots \\ s_{L,L} & r_{L,L} \end{pmatrix} \equiv \begin{pmatrix} s_1 & r_1 \\ \vdots & \vdots \\ s_M & r_N \end{pmatrix}. \tag{3.4}$$

The DTW tries to select the path, $\mathbf{L}$, that minimises the total or accumulated cost

$$\sum_{l=1}^{L} c(s_{L,l}, r_{L,l}). \tag{3.5}$$

This path is calculated using dynamic programming, so first the accumulated cost matrix D is defined, where each element is computed as

$$D(i,j) = min\{D(i-1,j-1), D(i-1,j), D(i,j-1)\} + c(s_i, r_j). \tag{3.6}$$

The recursion is initialised with $D(i,0) = \infty \; \forall \; n \in [1,M]$, $D(0,j) = \infty \; \forall \; n \in [1,N]$ and $D(0,0) = 0$. In figure 3.1b an example is shown and it can be seen how it is related to the cost matrix.

Once this matrix is computed a backtracking algorithm is run to calculate the optimum path. It is initialised at $l = L$ and in the location on the matrix (M,N) and each element is computed as

$$L_{l-1} = argmin\{D(i-1,j-1), D(i-1,j), D(i,j-1)\}, \tag{3.7}$$

where (i, j) represent the indexes on the matrix. So, the algorithm moves around the accumulated cost matrix from the top right until reaching the bottom left. In figure 3.1 the cost matrix and the accumulated matrix are shown together with the path the DTW computes. It is important to notice that the first point of each time series will be matched to each other and the same will happen with the last value, so it is a closed beginning and end problem.

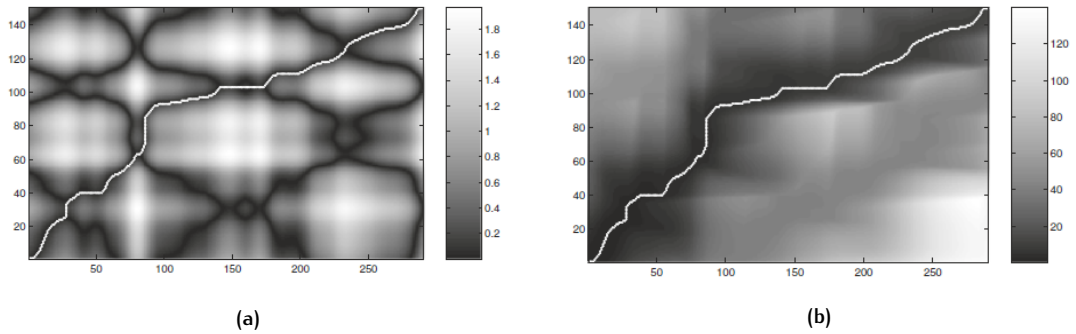


Figure 3.1: a) Cost matrix with the optimum path. b) accumulated cost matrix with the optimum path taken from [9].

This technique requires constructing the accumulated cost matrix, which has a $\mathcal{O}(NM)$ complexity, which can become a problem for big data sets [9]. One solution would be to reduce the amount of data by windowing the input, so that N and M are relatively small. However, this is not a sustainable solution, as it just constrains the input size.

Another solution is based on the fact that the optimum path is unlikely to go through the edges, so those points do not need to be computed. Therefore, a window is created and only the points that are located inside are computed, this method is called global path constrain. There are several windows and their suitability will finally depend on the application and the properties of the matching. In figure 3.2 the two commonly used windows are shown, the Itakura parallelogram on the left and the Sakoe-Chiba band on the right[21]. This reduces the computation time and the smaller the window is the faster it will be to compute.

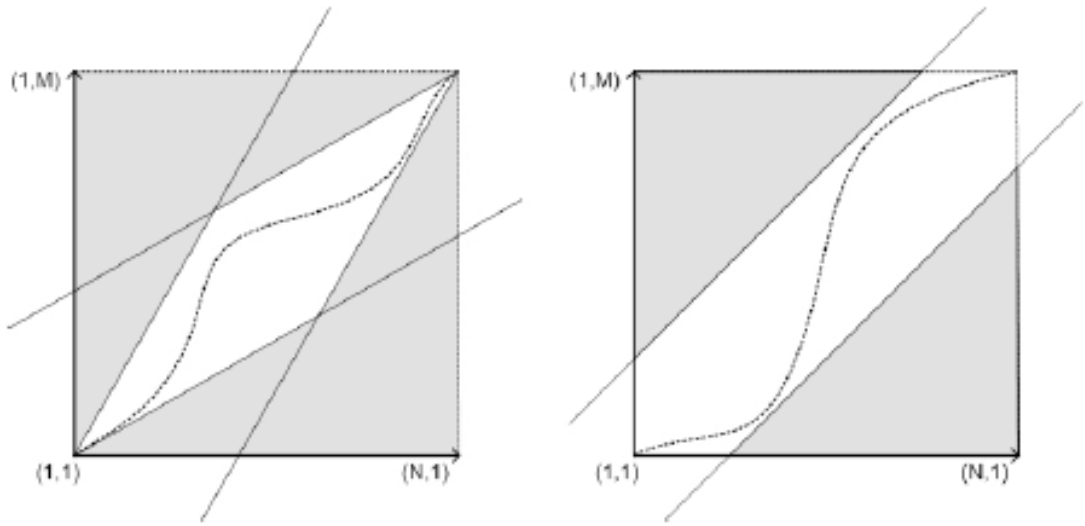


Figure 3.2: Left: the Itakura parallelogram, Right: Sakoe-Chiba band taken from [21].

However, there are some drawbacks to this method, the most relevant related to the fact that it will provide an unaltered result as long as the optimal path is inside the window [27]. So, if

the window is too narrow, the output path will differ from the optimum. Thus, it is important to understand theundergoing warping of the data sets. If no big variation is expected a narrow window may suffice, otherwise a wider window will still be required at a higher computation cost in order to obtain the proper warping.

Another approach is to use approximations instead of using the whole data. The data size can be reduced using numerous transforms, for instance using a low pass filter or downsampling. Afterwards, the downsampled problem can be solved, warped and converted back to its original resolution. This method can suffer from several problems as it is based in downsampling and approximations; therefore, it may give a suboptimal solution. These kinds of procedures are called abstraction techniques and they try to operate on a representation of the data [27].

The main critic to DTW is that it usually overfits the warping because it is very flexible [28]. It is common that certain point from the query is assigned to multiple from the reference, so the warping can destroy peaks and valleys as it is shown in figure 3.3. So, the algorithm tries to fix any difference in the value between curves by warping it, even though its origin could be another, such as noise or an interesting different feature between curves. This can create unwanted artefacts which can be crucial in certain applications including the automated depth matching.

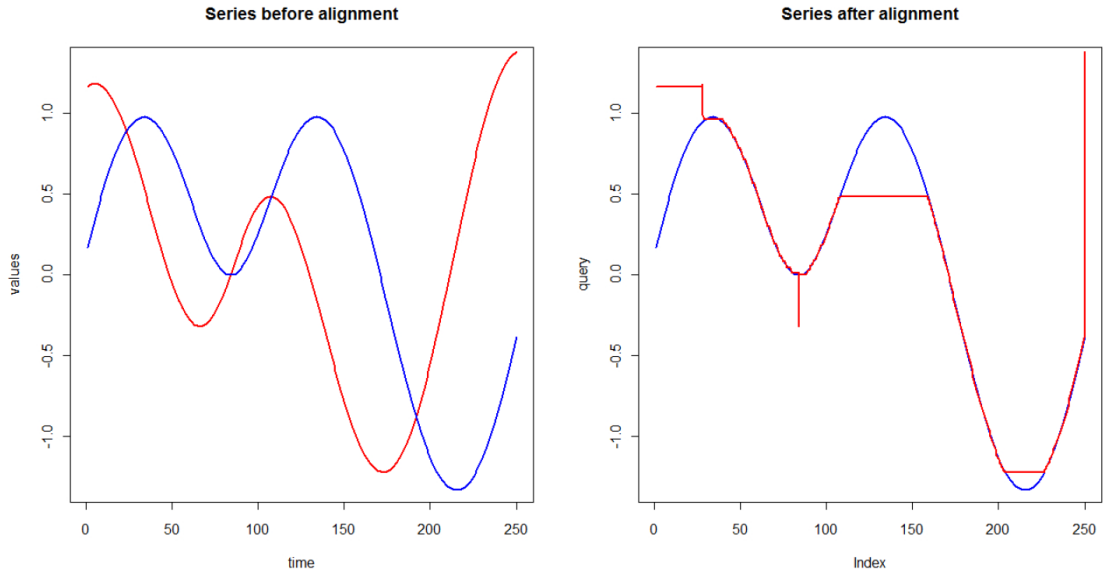


Figure 3.3: On the left the reference (blue) and the original query (red) are shown and on the right the result after the warping. The image has been taken from [21].

Several methods have been proposed for reducing the overfitting [26; 25; 29]. For example, considering how the accumulated cost function is built, the recursion can be changed to penalise skipping a value [9]. Some weighting coefficients, w_d, w_h and w_v, are added to obtain the recursion

$$D(i,j) = min\{w_d \cdot D(i-1,j-1), w_h \cdot D(i-1,j), w_v \cdot D(i,j-1)\} + c(s_i, r_j). \tag{3.8}$$

This technique is called slope weighting.

Another modification to reduce matching multiple points together is the Derivative Dynamic Time Warping (DDTW) [30]. In this case the cost function is changed, instead of using the

Manhattan distance a cost value related to the derivative is introduced. There are multiple ways to estimate the first derivative, the simplest one is given by

$$D_x(s_i) = \frac{s_i - s_{i-1} + \frac{s_{i+1} - s_{i-1}}{2}}{2}, \tag{3.9}$$

and the cost function would be given by

$$c(s_i, r_j) = \left(D_x(s_i) - D_x(r_j)\right)^2. \tag{3.10}$$

There are more modifications available but all of them employ similar techniques, either weighting the recursion or changing the cost matrix.

3.4 CORRELATION OPTIMISED WARPING (COW)

One popular variation of the DTW for reducing its flexibility is the Correlation Optimised Warping (COW) [26]. DTW performs the warping in a pointwise fashion, and this can have a big impact on the peaks and valleys of the signal. In the COW the signals are segmented first and the warping is done using the segments while allowing some flexibility for each segment; therefore, maintaining the overall shape of the curve.

First the data or the query is segmented into smaller groups. These segments are flexible, so they can be adapted to match the reference by linear means. It is an open ended problem, so the first point of the reference to match is known but the length of the segment is variable. In order to have an efficient calculation a maximum deformation is set, defined by "the slack". The cost measure is the normalised cross-correlation between the different segments and it is stored in the cost matrix. However, since the length of the segments is also variable, the different configurations must be considered, which is done by defining a secondary matrix per segment. The problem is solved using dynamic programming and backtracking the optimum path, similar to the DTW.

This method has been extensively used in chromatogram alignment with great success [31]. The main critic is that it is very dependent on its parameters, namely the length of the segments and the slack which determines the maximum stretch or squeeze of the warped segments. The computational time will also depend on these variables; the bigger the segments and the lower the slack the faster the computation would be; therefore, the more flexibility allowed the more configurations that need to be tested.

3.5 MACHINE LEARNING

One of the most recent attempts to automate the depth matching process was based on machine learning utilizing neural networks [6]. Depth matching can be thought as a pattern recognition problem, so it is sensible to tackle it with machine learning techniques. The main advantage is that it does not require to tune the algorithm and adjust it based on the different data types or properties because the program will do it by itself. Although some good results are achieved in the cited study, the main disadvantage is that it requires many samples for the training phase and the performance will depend on these values.

3.6 PARAMETRIC TIME WARPING (PTW)

The Parametric Time Warping (PTW) was introduced in the paper [10]. In this method the warping function has a pre-defined functional form. While the DTW and similar methods try to

find the most similar points and match them individually, the PTW tries to estimate the warping function directly.

In this approach, the warping function is assumed to have certain parametric form, the most basic being a K order polynomial

$$w(t) = \sum_{k=0}^{K} a_k t^k. \tag{3.11}$$

The warping function is then fully characterised by the set of polynomial coefficients **a**. Thus, the problem of estimating the warping function, $w(t)$, is reduced to estimating the parameters **a**. This is done by solving an optimisation problem. A cost function is defined that measures the similarity between the reference and the warped query and the optimum parameters are obtained by maximising the similarity in an iterative manner. In the original paper [10] the cost function, $\mathfrak{C}$, was defined as the least squares function between the reference and the warped query, while in a more recent paper [32] it is suggested to use a cost function related to the normalised cross-correlation. The optimization problem can always be written as a minimization problem,

$$\arg\min_{\mathbf{a}} \mathfrak{C}(r(t), s(w(t))), \tag{3.12}$$

which gives the optimum **a** parameters and; in consequence, the optimum warping function given the parametrization.

PTW successfully aligns chromatograms [28] due to its moderate flexibility, which allows warping the curve without introducing distortions. Furthermore, it has also been proposed to use a B-spline function for parametrisation with certain number of knots equidistant with each other instead of a K order polynomial [28] to improve the warping. Although there has not been much research regarding this approach, it seems a well suited model for the depth matching problem as it is similar to how the matching is done manually, so it was decided to focus the research in this direction.

4 PARAMETRIC DEPTH MATCHING

In this thesis the Parametric Time Warping [10] was chosen to depth match borehole logs. The methods described in this chapter have been implemented using Python programming language and tested on real data to analyse their performance.

4.1 K DEGREE POLYNOMIAL

In the early implementation of PTW a K degree polynomial was used as the warping function [10]. The known implementations in the field of biology used a second-degree polynomial to define the warping function, to account for a constant offset, a linear stretching or squeezing and a high order disturbance [33]. In theory using a higher degree polynomial might be beneficial, because, as stated in Taylor's theorem [34], any function can be expressed as an infinite sum of weighted polynomials and the higher the degree the smaller the approximation error. However, increasing the polynomial degree also increases the number of optimization parameters and its complexity and it can prompt errors such as Runge's phenomenon [35; 36] . Although there exists a high order polynomial that describes the function properly, estimating it is not simple, and the convergence to a proper result is not guaranteed.

For borehole matching the majority of misalignments seem to be caused by offsets and linear stretchings and squeezings [5]. Some higher order problems also appear due to the friction with the borehole walls and the mud, but they have a smaller influence. Due to this fact, a rigid warping was preferred and a first degree polynomial was chosen, considering

$$w(t) = \sum_{k=0}^{1} a_k t^k = a_0 + a_1 t \tag{4.1}$$

the warping function to assess results the PTW could provide. The a_0 parameter account for constant offset, while the a_1 parameter defines the degree to which a signal is squeezed or stretched.

4.2 SIMILARITY MEASURE

To assess the warping results a suitable robust similarity measure needs to be defined. The selection will depend on the problem to solve. For continuous reference $r(t)$ and warped function $s(w(t))$, the most basic is based on the Euclidean distance also known as least-squares and defined as:

$$\sqrt{\int_{t\in\mathcal{D}} |r(t) - s(w(t))|^2 \, dt}, \tag{4.2}$$

where $\mathcal{D}$ is the domain were both functions are defined. The least squares is one of the most common cost functions and it is commonly used when the noise type is not known as it was shown to give good results [37]. It penalises big errors heavier than smaller ones and it is optimum with white Gaussian noise. The main drawback is that it only works if both curves

have the same magnitude. If for example the query is twice the magnitude of the reference, $s(t) = 2r(t)$, the least-squares will not give proper results because the similarity will be purely measured by the magnitude, disregarding the curve shape. This results in unrealistic solutions when Euclidean distance based metrics are used for depth matching without any preprocessing. Therefore, in those cases the functions must be normalised, by removing the mean and dividing by standard deviation.

In real applications we are dealing with discretised representations of the signals. The points were the values of the functions are evaluated can be expressed in an array $\mathbf{t} \equiv (t_1, \cdots, t_N)$ composed of N samples which are equidistant, as certain sampling rate is used . Thus, the integral in equation 4.2 is converted to a finite sum

$$\frac{\sum_{i=1}^{N} |r(t_i) - s(w(t_i))|^2}{N} = \frac{||r(\mathbf{t}) - s(w(\mathbf{t}))||_2^2}{N}. \tag{4.3}$$

Another useful measure of the similarity is the cross-correlation. Several authors suggested to use cost functions related to the cross-correlation as this is a better measure of similarity between two curves[32]. In general, for two continuous functions, the cross-correlation is defined as

$$\rho_{rs}(\tau) = \int_{t \in \mathcal{D}} r(t)^* s(w(t) + \tau)\, dt, \tag{4.4}$$

where τ is a delay. Cross-correlation is a function of the delay parameter τ, which defines a shift in the function, and not a scalar value. This is the most often used similarity measure for computing the delay between two functions [38]. However, this is not the best measure if noise is present in the data. The positive aspect of the cross-correlation is that it does not require previous normalization, as it is not affected by different scaling factors. However, in order to have the optimum behaviour, the functions must be zero-mean. For depth matching, the comparison is done with the original curves, so no extra delay is added and the expression can be simplified to

$$\rho_{rs}(0) = \int_{t \in \mathcal{D}} r(t)^* s(w(t))\, dt. \tag{4.5}$$

For discretised signals, the cross-correlation is expressed as a finite sum

$$\phi_{rs}(n) = \sum_{m=1}^{N} r(t_m)^* s(w(t_{m+n})) \Rightarrow \phi_{rs}(0) = \sum_{m=1}^{N} r(t_m)^* s(w(t_m)) = r(\mathbf{t})^H \cdot s(w(\mathbf{t})). \tag{4.6}$$

The cost function can be expressed as the cross product between the Hermitian of the reference $r(t)$ and the warped query$s(w(t))$. The bigger this value is the more similar the two signals are. However, this value is not bounded, so usually the normalised cross-correlation is used which is obtained by dividing the cross-correlation value by the variance of the signals,

$$\hat{\phi}_{rs}(0) = \frac{r(\mathbf{t})^H \cdot s(w(\mathbf{t}))}{\sigma_r \sigma_s}. \tag{4.7}$$

If both curves are perfectly correlated the cross-correlation value will be 1. On the other hand, if no correlation is present the normalised cross-correlation will be 0, and if the signals are perfectly negatively correlated the value will be -1. Therefore, the goal is to obtain the highest value possible close to 1.

The cross-correlation is valid if the functions are zero mean. The simplest solution to achieve this condition is to subtract the mean from both $r(\mathbf{t})$ and $s(w(\mathbf{t}))$.

Finally, the normalised cross-correlation problems are defined as a maximisation problem. This is equivalent to the minimisation of the cost function defined below:

$$\mathfrak{C} = 1 - \hat{\phi}_{rs}(0) = 1 - \frac{r(\mathbf{t})^H \cdot s(w(\mathbf{t}))}{\sigma_r \sigma_s}, \tag{4.8}$$

which will not affect the performance. This expression was used in this project as the cost function for the optimisation.

In the paper [32], the normalised cross-correlation is proposed but a matrix W is introduced to smooth the result, so the normalised cross-correlation is

$$\hat{\phi}_{rs,W}(0) = \frac{r(\mathbf{t})^H \cdot W \cdot s(w(\mathbf{t}))}{\sigma_r \sigma_s}. \tag{4.9}$$

This matrix is a triangular pulse with ones in the diagonal and decreasing values on the sides. Thus, it can be seen as an averaging operation. However, after some initial attempts with this definition of the cost function, no noticeable difference on the warping result was observed.

Furthermore, this extra multiplication adds complexity to the computation of the cost function, and the smoothing can turn out to be very costly. If the curves suffer from high noise, which would justify the use of the smoothing parameter, it is more sensible and efficient to preprocess and smooth the signal and later feed the smoothed curves to the algorithm. In consequence, the similarity measurement used during this whole project was the simple normalised cross-correlation.

Finally, it is important to determine the depth instances where the cost function is evaluated, $\mathbf{t}$. There are a couple of options, the first would be to define it as the depth indexes where either the reference or the query are known, so $\mathbf{t} = \mathbf{t}_s \cup \mathbf{t}_r$. However, this increases the number of samples and some might not be important as the part from one might not be on the other. For instance, if the query takes a wider depth range than the reference, a part of it will not appear in the reference, so using it will not add relevant information. Furthermore, in spite of using more elements, they do not necessary add relevant information; for example, if two points are very close to each there will not be a significant difference between using both or just one.

Another option is to consider the depth indexes from the reference, $\mathbf{t} = \mathbf{t}_r$. This approach is more sensible as the idea is to align the query to the reference; therefore, only the points of the reference are important. This approach was used throughout this project.

4.3 APPROXIMATION FOR DISCRETE FUNCTIONS

The theory presented above works under the assumption of continuous functions. Unfortunately, in depth matching borehole logs only discrete instances at depth are available, which poses a problem for computing $s(w(t))$ as $w(t)$ can have any value in $\mathbb{R}$, so it is continuous. Thus, during the optimization it can happen that $s(w(t)) \notin \mathbf{s}$.

This can be solved by applying an interpolation. In that way from the available discrete data points, $\mathbf{s}$, we can obtain the expanded continuous function, $s(t)$. In the original paper [10], a linear interpolation is used and we follow the same approach. Therefore, a value at a depth $t \in [t_{s,j}, t_{s,j+1}]$ can be calculated as

$$s(t) = \mathbf{s}(t_{s,j}) + (t - t_{s,j}) \frac{\mathbf{s}(t_{s,j+1}) - \mathbf{s}(t_{s,j})}{t_{s,j+1} - t_{s,j}}. \tag{4.10}$$

Although more advanced interpolations could be employed, such as cubic splines [17], this step is not crucial to warping and should have a minor influence on the result.

4.3.1 Values outside the domain

The interpolations are only valid for values inside the known domain, so for $t \in [t_{s,1}, t_{s,M}]$. For depths outside that domain the interpolation is not defined and this can occur during the optimization process as $w(t)$ can provide any value in $\mathbb{R}$. The most common case is when $\mathbf{t}_r$ is longer than $\mathbf{t}_s$, so when the length of the measurement of the reference is longer than the measurement of the query. In those cases there may be sections where $w(t_{r,i}) \notin [t_{s,1}, t_{s,M}]$. Fortunately, several solutions are given in the literature for similar problems:

- Give the closest neighbouring value:
  $t < t_{s,1} \Rightarrow s(t) = \mathbf{s}(t_{s,1})$
  $t > t_{s,M} \Rightarrow s(t) = \mathbf{s}(t_{s,M})$
- Mirroring the function, so the function is assumed to be periodic with a period of $T = t_{s,M} - t_{s,1} + t_{slope}$, where the last variable is an extra value to avoid having an step between the first and last values of $\mathbf{s}$ and have a smoother transition.
- Fixing the first and last values, so
  $s(w(t_{R,1})) = \mathbf{s}(t_{s,1})$
  $s(w(t_{R,N})) = \mathbf{s}(t_{s,M})$
  This is the case for the DTW [25].
- Computing only the values for $t \in [t_{s,1}, t_{s,M}]$ and disregarding values outside the interval. If $w(t)$ projects a value outside the depth where the reference is defined that value will not be used for computing the cost function.

Fixing the initial and final values can be a good idea if both curves cover a similar region on depth, but often one of the curves is significantly larger than the other, so in this problem it cannot be assumed that both curves will start and finish at the same depth. Therefore, forcing this matching will introduce distortions on the warped query and will make the matching harder.

Regarding the other methods, although all of them gave similar results the last one was chosen. The elements that are outside the "known" domain will not disturb the cost function. When the depth matching is done manually only the known data is considered so it seems sensible to do the same for the cost function.

The main drawback is that this can cause problems with the optimisation as dropping elements does not carry any penalisation. A way to solve this is to introduce a drop cost so that only the minimum required elements are removed [2]. Nonetheless, during the project no problem caused by this fact was observed, so no drop cost was introduced.

4.4 PIECEWISE LINEAR TIME WARPING

In depth matching related to borehole logs, the squeezing is not uniform through the whole well depth, in contrast with other applications such as chromatogram alignment [10]. For logs the squeezing can take place between different sections, so it might not be uniform. This behaviour and the solution obtained using the manual warping can be expressed with a piecewise linear function also known in the literature as broken lines [39].

The strategy is to implement the PTW assuming a piecewise linear function as the warping function; thus, imitating the current manual depth matching procedure. This was previously suggested in [28] where a B spline was used as the warping function. In this particular case, a first order B spline or a piecewise linear function is used. Higher order B splines could also be used, for instance, in curve approximation it is usual to employ third order B splines as they are optimum when Gaussian noise is present [40]. Nonetheless, the current problem is different as the B spline is not meant to fit a cloud of points with a function, but to fit the warped query, $s(w(t))$, to the reference, $r(t)$.

Higher order B splines introduce smoothness to the curve which is very convenient for curve fitting but its benefits for the current case are unclear. Although it may provide a more realistic warping function, it comes at the expense of setting conditions to the derivatives at the knots and losing some degrees of freedom because of the higher B spline order. Therefore, it was decided to use a simple piecewise linear function in order to have full control on the warping function and to only use linear shifts as it is done in the manual depth matching.

Another way to view this problem is that a piecewise linear function is a first order approximation to model an unknown warping function. Even in cases where the warping function is more complex or nonlinear shifts are involved, a first order approximation will be used. For borehole depth matching this approach proved adequate.

The mathematical formulation for piecewise linear time warping is as follows. First a set of points $\mathbf{p} \equiv (\mathbf{p}_1, \cdots, \mathbf{p}_n)$ is defined formed by n points or knots, where each has two components

$$\mathbf{p}_i = \begin{pmatrix} t_{knot,i} \\ a_i \end{pmatrix}. \tag{4.11}$$

The set of $\mathbf{t}_{knot}$ are the depth instances from the reference and the set of $\mathbf{a}$ are the values the warping function gets on each of those depth instances. The rest of the function is calculated in a similar fashion as the first order approximation is done. For a $t \in [t_{knot,j}, t_{knot,j+1}]$ the value of the function is given by

$$f_{BL}(t) = a_j + \left(t - t_{knot,j}\right) \frac{a_{j+1} - a_j}{t_{knot,j+1} - t_{knot,j}}. \tag{4.12}$$

The set $\mathbf{a}$ is the projection onto the depth of the query for each knot. Despite being a measure of depth, this name was used in order to keep the same nomenclature in the optimization problem.

In figure 4.1 an example of this kind of function is shown where the knot points are marked. It is a continuous function but it is not differentiable in the knots.

A similar problem as before arises for values of $t \notin [t_{knots,1}, t_{knots,n}]$. Similar methods can be used to approximate the value, such as giving the nearest value or assuming periodicity. In addition to those, the trend of the function can be used to predict the value using as slope the slope of the nearest interval, so the linear function can be continued outside the domain. Fortunately, as the depth knots are related to the reference depth, the first and last will be bounded by the reference, so

$$t_{knot,1} = t_{r,1} \quad and \quad t_{knot,n} = t_{r,N}. \tag{4.13}$$

Since the cost function is only computed inside $\mathbf{t}_r$, the piecewise linear function will never be evaluated outside the previously defined domain. All the information with respect to how it is built is given in Appendix A.

Therefore, the warping function will be assumed to be a piecewise linear function, $w(t) = f_{BL}(t)$. This was suggested in the paper [41] where the Semi-parametric Time Warping (STW) is introduced as a general formulation for warping functions based on B splines. Since in this work a piecewise linear function is used the method will be denoted as Piecewise Linear Time Warping (PLTW).

Thus, the problem can be expressed as

$$\arg\min_{\mathbf{a}} \mathfrak{C}(r(t), s(w(t))) = \arg\min_{\mathbf{a}} \mathfrak{C}(r(t), s(f_{BL}(t))), \tag{4.14}$$

where the knot locations, $\mathbf{t}_{knot}$, are a priori defined.

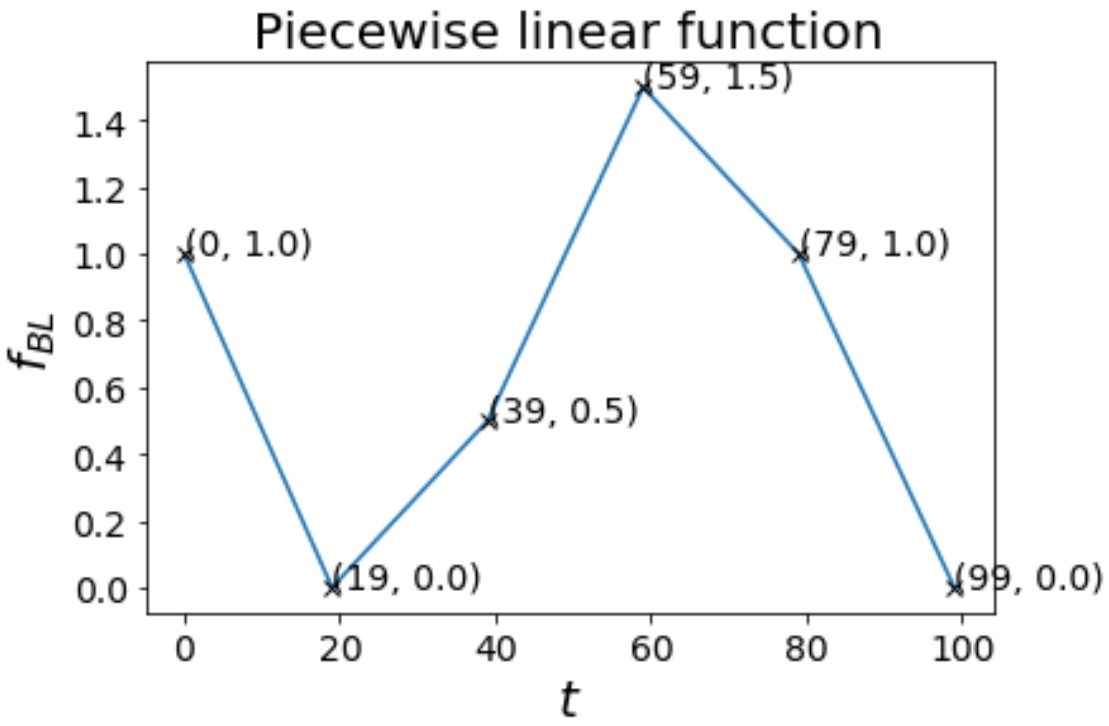


Figure 4.1: Example of a piecewise linear function or broken lines.

4.4.1 Optimization

As an analytic solution to the minimization problem we are trying to solve does not exist, numeric optimization methods are used instead. One of the widely used is the gradient descent method because in general it provides good results [10; 42]. Gradient descent method is initialized with a set of parameters, $\mathbf{a}_0$. Subsequently the warping function is computed using those parameters fixed, making it a function of t only. Afterwards, the cost function can be evaluated and the update, $\mathbf{a}_1$, is computed. The process is iterated until a suitable match between reference and warped signal is found.

The updating process depends on the method used. The simplest implementations compute the gradient of the cost function with respect to the parameters $\mathbf{a}$ and the following parameter set $\mathbf{a}_{i+1}$ is calculated based on that gradient value. In [10] the update is computed analytically, in this work; however, the gradient is computed using numeric methods [43]. So, the gradient at $\mathbf{a}_i$ will be computed by calculating the cost function at $\mathbf{a}_i$, $\mathfrak{C}(r(t), s(w(\mathbf{a}_i), t))$, and the value of the cost function close to $\mathbf{a}_i$, a very small number away from it. Once the different cost functions are known the gradient can be calculated.This enables freedom on the choice of the parametric warping function as well as on the use of any existing optimization technique to solve the problem, for instance using the second order momentum or the Hessian to converge faster.

The iterations continue until the specified convergence criteria are met. Those can be either a maximum number of iterations have been reached or the difference between the current and next parameters, the update, is considered low enough.

Several methods were tested to solve the problem, for instance, the Broyden Fletcher Goldfarb Shannon (BFGS) algorithm [44] or the truncated Newton (TNC) algorithm [45]. Having tested numerous algorithms the best result was obtained using Sequential Quadratic Programming (SQP) [46]. A further development was to add some constraints to the maximum and minimum values the projections, $\mathbf{a}$, can have. So, the minimization problem can be rewritten as

$$\arg\min_{\mathbf{a}} \quad \mathfrak{C}(r(t), s(f_{BL}(\mathbf{a}, t))) \tag{4.15a}$$

$$\text{subject to} \quad \mathbf{a} \geq \mathbf{lb}, \tag{4.15b}$$

$$-\mathbf{a} \geq -\mathbf{ub}, \tag{4.15c}$$

where both **lb** and **ub** are respectively the lower and upper bound of the projections. This limit is set so that the final projections are a maximum of 60 meters away from the initial, $\mathbf{a}_0$; therefore being $\mathbf{lb} = \mathbf{a}_0 - 60$ and $\mathbf{ub} = \mathbf{a}_0 + 60$.

Not only do these constraints guarantee monotonicity and that the result will be reasonable but it also helps to converge faster. When solving this problem using SQP the result is obtained 3 times faster than the unconstrained minimisation.

The Piecewise Linear Time Warping (PLTW) is described by the algorithm 4.1. The only input parameter is the number of knots and the warped query is obtained by solving a constrained minimization problem.

Algorithm 4.1: PLTW

Result: Warped query

1 define n;

2 construct $\mathbf{t}_{knot}$;

3 create the initial parameters $\mathbf{a}_0$;

4 $\mathbf{a}^* \leftarrow \arg\min\left(1 - \frac{r(\mathbf{t})^H \cdot s(w(\mathbf{t},\mathbf{a}))}{\sigma_r \sigma_s}\right)$ with constraints;

5 $\mathbf{s}_W = s\left(w\left(\mathbf{t}, \mathbf{a}^*\right)\right)$

4.5 PIECEWISE LINEAR TIME WARPING WITH KNOT OPTIMIZATION

A further modification would be to optimise the knot locations as well as their projections. These kinds of procedures are already well studied for B splines applied to fitting data clouds [47]. The splines try to minimise a least squares problem, which can be linearly solved using QR decomposition if the knot location is fixed [39]. However, it was seen that the knot location impacts the result [48], thereafter its selection has been a field of study. There are several methods in the literature which handle this problem, most of them are based on dividing the problem in two independent problems. First, the knot location problem, which is nonlinear, is solved and afterwards the computed knots are used to obtain the spline [49]. There are different ways of solving the nonlinear problem, but most of them try to avoid heavy computations by, for instance, computing the location using the second derivative of the data cloud.

In the basic PLTW a fixed knot location is used distributing certain number of knots uniformly, so this technique can also be applied for this case. An intuitive way to improve their locations is to run a similar optimization scheme as for the projections. This can be expressed as

$$\arg\min_{\mathbf{t_{knot}}} \mathfrak{C}(r(t), s(w(t))) = \arg\min_{\mathbf{t_{knot}}} \mathfrak{C}(r(t), s(f_{BL}(t))). \tag{4.16}$$

However, the cost function also depends on the parametrization and its values, **a**. So if the selected **a** is suboptimal there is no guarantee of the effectiveness of the knot location. In splines this problem is tried to solve by using the second derivatives of the data and solving an l_1 minimisation problem. However, this cannot be blindly applied in the parametric warping, because the two data sets are not directly comparable as there is a warping function between them. A different approach which is also present in spline theory is to solve a global minimisa-

tion problem where the knots and the spline are computed simultaneously [50]. Therefore, the best solution is to optimise both the knots and the parametrization parameters at the same time

$$\arg\min_{\mathbf{a},\mathbf{t_{knot}}} \mathfrak{C}(r(t), s(w(t))) = \arg\min_{\mathbf{a},\mathbf{t_{knot}}} \mathfrak{C}(r(t), s(f_{BL}(t))). \tag{4.17}$$

There are multiple techniques to solve these kinds of optimisations. For instance, iteratively, first the gradient of the cost function with respect to $\mathbf{a}$ can be computed keeping the knots fixed and an iteration can be made. Subsequently the same is done with the gradient with respect to $\mathbf{t_{knot}}$ while keeping $\mathbf{a}$ fixed and the process can be repeated using $\mathbf{a}$ until certain convergence criteria is achieved [43]. Nonetheless, for the current project a simpler approach was used. Both variables are combined into one larger vector, $\mathbf{v}$,

$$\mathbf{v} \equiv \begin{pmatrix} \mathbf{a} \\ \mathbf{t}_{knot} \end{pmatrix} \equiv \begin{pmatrix} a_1 \\ \vdots \\ a_n \\ t_{knot,1} \\ \vdots \\ t_{knot,n} \end{pmatrix} \tag{4.18}$$

which contains all the variables to optimise. Furthermore, since all are related to the distance or depth the same constrains as in equation 4.15 can be set and the problem can be rewritten as

$$\begin{aligned} \arg\min_{\mathbf{v}} \quad & \mathfrak{C}(r(t), s(f_{BL}(t))) && \text{(4.19a)} \\ \text{subject to} \quad & \mathbf{v} \geq \mathbf{lb}, && \text{(4.19b)} \\ & -\mathbf{v} \geq -\mathbf{ub}, && \text{(4.19c)} \end{aligned}$$

where $\mathbf{lb} = \mathbf{v}_0 - 60$ and $\mathbf{ub} = \mathbf{v}_0 + 60$. This formulation enables to use Sequential Quadratic Programming (SQP) [46] to solve the optimization problem as in equation 4.15. The explained method which optimises the knots as well as their projections is summarised in algorithm 4.2.

Algorithm 4.2: Optimised PLTW

Result: Warped query

1 define n;
2 construct $\mathbf{t}_{knot}$;
3 create the initial parameters $\mathbf{a}_0$;
4 construct $\mathbf{v}$;
5 $\mathbf{v}^* \leftarrow \arg\min\left(1 - \frac{r(\mathbf{t})^H \cdot s(w(\mathbf{t},\mathbf{v}))}{\sigma_r \sigma_s}\right)$ with constraints;
6 $\mathbf{s}_W = s\left(w\left(\mathbf{t}, \mathbf{v}^*\right)\right)$;

4.6 EXPERIMENTAL RESULTS

The methods presented in the previous section have been tested using real data of borehole logs to understand their robustness and drawbacks.

4.6.1 PTW and Piecewise function comparison

First the experiments with basic PTW of first degree were performed to evaluate the method and establish a baseline to compare with. As shown in chapter 2 the data consist of different gamma

ray runs which can go from a few hundred meters to a few thousand. In order to analyse the performance only a section of the whole run was used, because the PTW cannot handle multiple linear shifts. In figure 4.2 just the first 500 values were used.

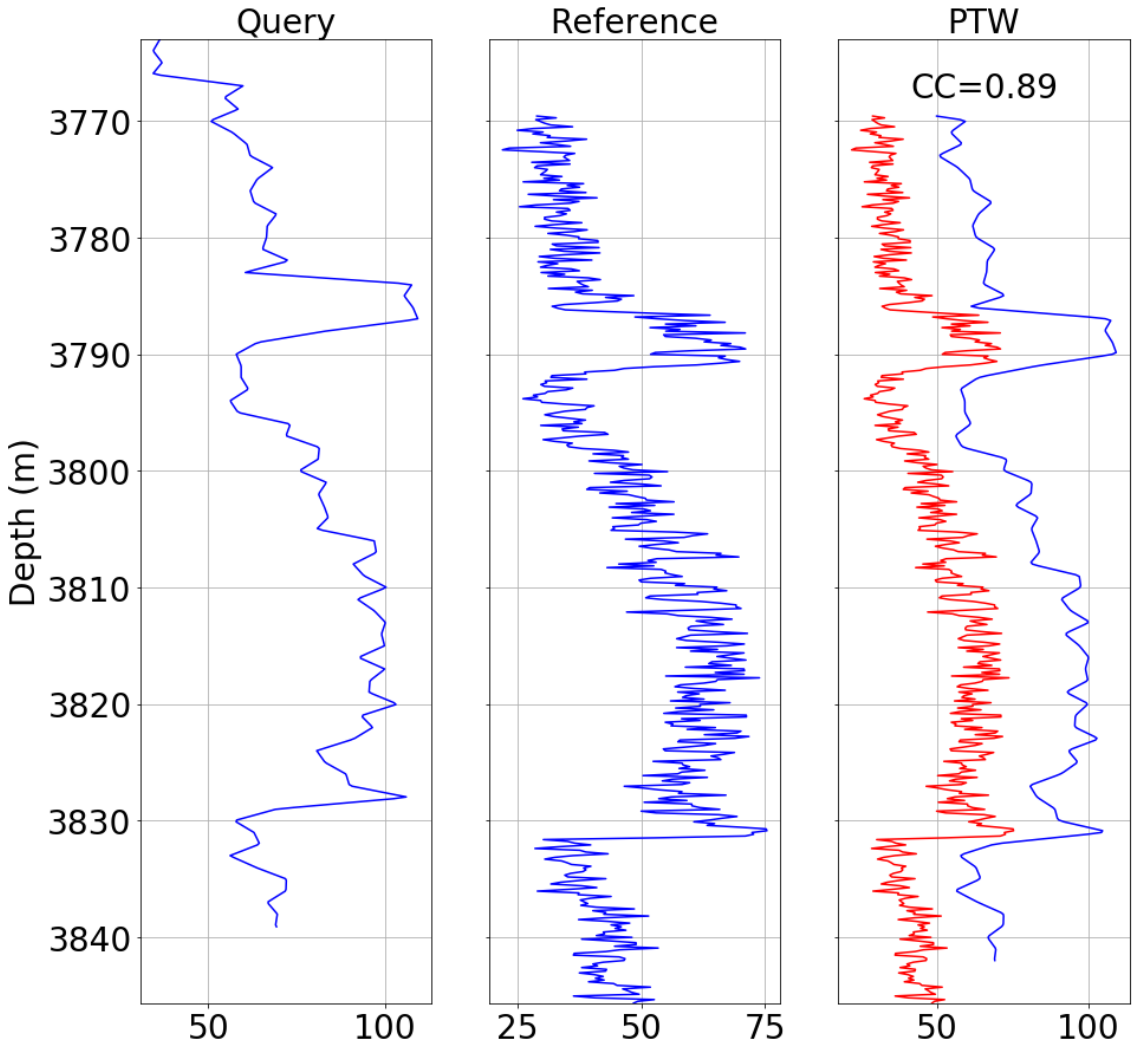


Figure 4.2: PTW solution on an extract of the raw data. In the left the original query with no warping is plotted, in the middle the reference is shown and in the right the reference in red and the warped query are plotted together. In the top right the total cross-correlation of the warped query is shown.

The optimization starts with an initial alignment and tries to improve the result with each iteration. The initial alignment will be the initial query at the common depth instances without any scaling or offset. Since a first degree polynomial was used, only an offset and a linear squeezing or stretching were considered.

The initial results are promising as the resulting alignment has a high total cross-correlation and the main peaks and significant features are aligned. Very good results obtained here also seem to incline that the majority of misalignment is caused by one unique linear shift. If there are multiple squeezes and stretches the ground assumption of the PTW will not be correct and the alignment will be incorrect as well. Therefore, for this case, the direct implementation is only valid for small sections of the signals and not for the whole log run.

Although the PTW provided a good result, for some of the real data tested this will not be the case for more complicated examples where multiple shifts are present. Therefore, the Piecewise Linear Time Warping (PLTW) algorithm was applied using the same log section. The

algorithm requires to choose the knot locations first. In general, the more knots used by the algorithm the more flexible the warping will be as more linear shifts are allowed.

The optimum case would be to have as many knots as linear shifts in the data and locate them in the appropriate depth instances, as it is done in the manual depth matching. Unfortunately, neither of those properties are a priory known. Following [41] a predefined number of knots, 4 in this case, were uniformly distributed, so the distance from one knot to the next is the same for all of them. In figure 4.3 the comparison between the PTW and the PLTW is shown. The results of the PLTW are virtually the same as those for the PTW and the main features are well aligned.

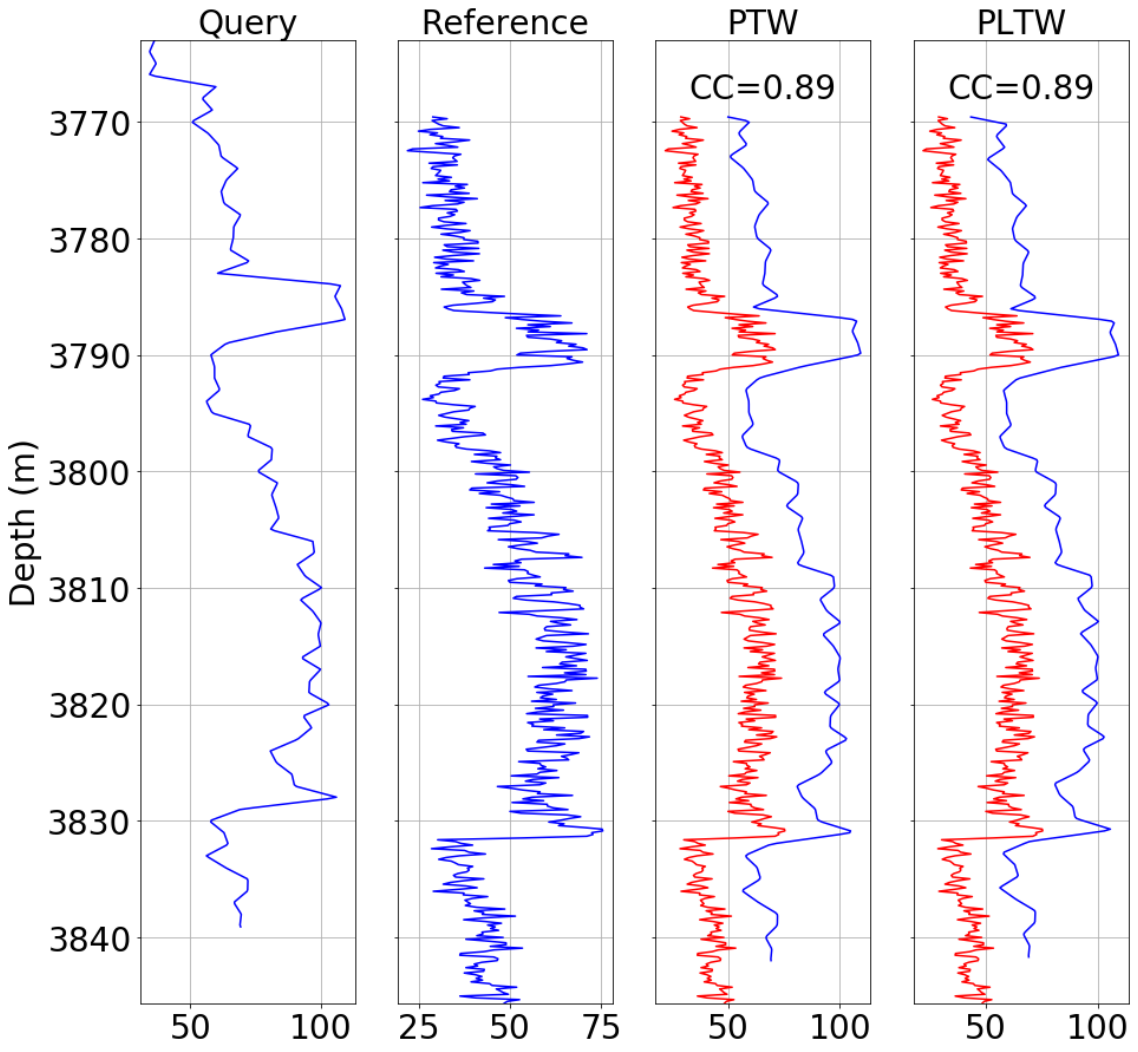


Figure 4.3: From left to right: First the query, secondly the reference and thirdly the warping obtained with the first order Parametric Time Warping. Finally, the warping obtained with the piecewise linear parametrization. In red the reference is plotted. In the top the total cross-correlation between the curves is shown.

In consequence, the PLTW is also successful coping with simple misalignment cases. In order to better understand the applicability of the PLTW to borehole depth matching different data sets were used and modifications to those introduced.

4.6.2 Initial parameter dependence

The first test was to check how the result would change if different initial parameters were used. Since the solution is obtained with an optimization problem, the initial parameters might be relevant, so it is sensible to understand the relation. In figure 4.4 multiple initial parameters were used to see whether there was any influence in the result. The number of knots was kept constant and fixed to 4 in order to have a fair comparison. Another section is used for this test which also has 750 data points. It is a more challenging set because more peaks and valleys are present. A known offset, t_{offset}, is added to this data and that is used as the initial parameters, so $\mathbf{a}_0 = \mathbf{t}_{knot} + t_{offset}$, and the result of the warping is plotted next to it.

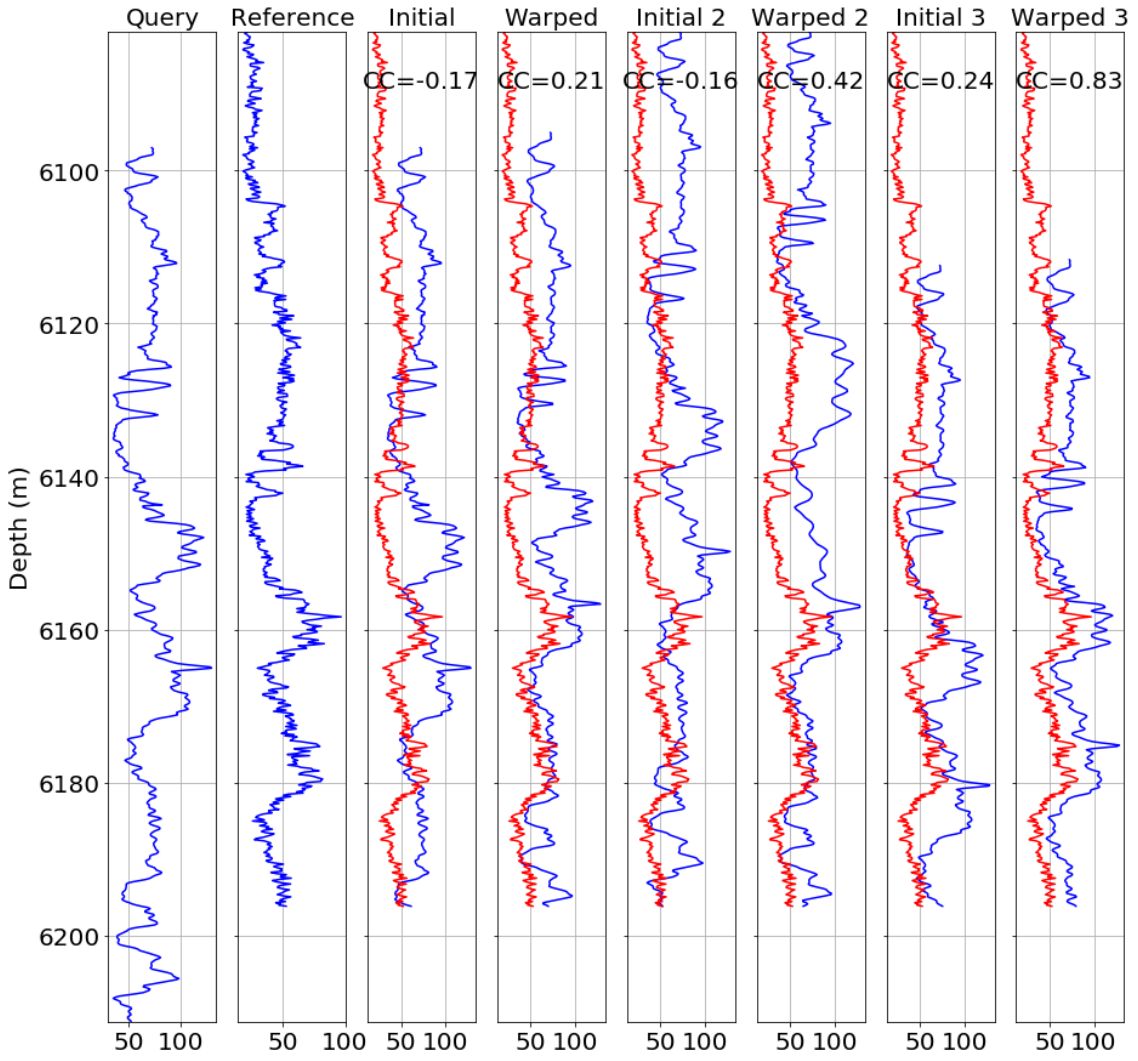


Figure 4.4: In the left the query and the reference signal. Afterwards three pairs of initial parameters and the warping result using a piecewise linear parametrization are shown. In red the reference and in the top the total cross-correlation are shown.

The first set is the original, where no offset is added to the query. The main two peaks are misaligned, and the last one is assigned to the penultimate; therefore, the cross-correlation has a very low value. The second set has an offset so that the initial curve is more misaligned, so the offset acts in the wrong direction, displacing it even more. The result for the last section is identical while the first is different, and it also provides a low total cross-correlation. Finally, another offset was tried but in this case the first iteration is more similar to the correct alignment, so the offset acts in the right direction. In this occasion the resulting warping is more accurate.

In spite of still showing some misalignment traces, it could be considered to be correct and it gives a relatively high total cross-correlation.

It can be concluded that in the three cases considered the warped curve is similar to the initial curve that was used for the iterations. So, the warping is not very flexible, it will just adapt the first iteration to align it as good as possible, converging to the local optimum. This can be better understood considering that the technique is based on a gradient descent method, so the local optimum will be found by iterating until the set of parameters is found. Thus, if a peak from the query is initially located on another peak from the reference, the algorithm will try to adjust the warping so that the peak adjusts to the other peak, regardless of whether there is another peak which suits it better.

Therefore, the current algorithm has a good performance if the initial parameters feed to it are close to the real warping. However, if an inappropriate starting parameter set is chosen, the algorithm will not produce the correct warping. On the other hand, the low flexibility level ensures that the warping distortions will be minimal so only minor adjustments will be performed. Thus, the main features will not be destroyed as in more flexible techniques, such as the DTW.

4.6.3 Knot number dependence

Another relevant parameter are the knots. For instance, it is clear that their locations will affect the result; if fewer knots than real shifts present in the distorted signal are used, it will not be possible to retrieve the optimal warping and only an approximation will be obtained. On the other hand, if the knots are located exactly where the linear shift takes place it is likely that the result would be better than if the knots where located in between.

In order to draw conclusions and determine how the knots affect the result a number of tests were performed. As explained before the simplest solution is to distribute a number of equidistant knots on the curve. In figure 4.5 the results of the warping using 4, 10 and 20 equidistant knots are shown. It is clear that increasing the number of knots increases the flexibility of the warp, since there are more segments and; thus, more possibilities for the matching. The figure ascertains the hypothesis, with 4 knots the changes are minimal and the warping just tries to adjust by making minor changes. On the other hand, with 10 knots some features are differently warped but the overall result is similar, although one of the valleys is incorrectly warped. This shows that the extra flexibility is not always positive as it may create artificial alignments. Furthermore, the last curve uses 20 knots and the result differs significantly from the others. The warping can be more severe and can give more flexible results, but this does not always mean the result will be better.

The more knots there are, the more flexibility the warping will have. This is an interesting results, since the flexibility of the algorithm can be controlled with the knot number and their locations. Although the algorithm maximises the total cross-correlation, the changes are done using the edges of each interval defined by knot positions. If the interval is big the warping would not dramatically deform the interval. But for a small interval the warping could cause heavy deformations. This is not something per se negative, but rather something to be aware of and consider when applying the technique.

Furthermore, for the examples tested here the knot location does not seem to be crucial, although it is obvious that locating them in certain spots might provide a better outcome. Nonetheless, for more complicated examples, it is very likely that a smart knot location may be beneficial and would reduce the necessary minimum number of knots to deploy and; therefore, decreasing the computational complexity. One option would be to locate the knots in the main features. However, this requires a preprocessing step of detecting peaks, valleys or other relevant features.

In spite of the advantages having a knot location preprocessing step could bring, it was decided to use the simple equidistant distribution of knots as it already provided satisfactory results. Investigating the optimum positioning of knots can be a part of future work to improve the presented algorithm.

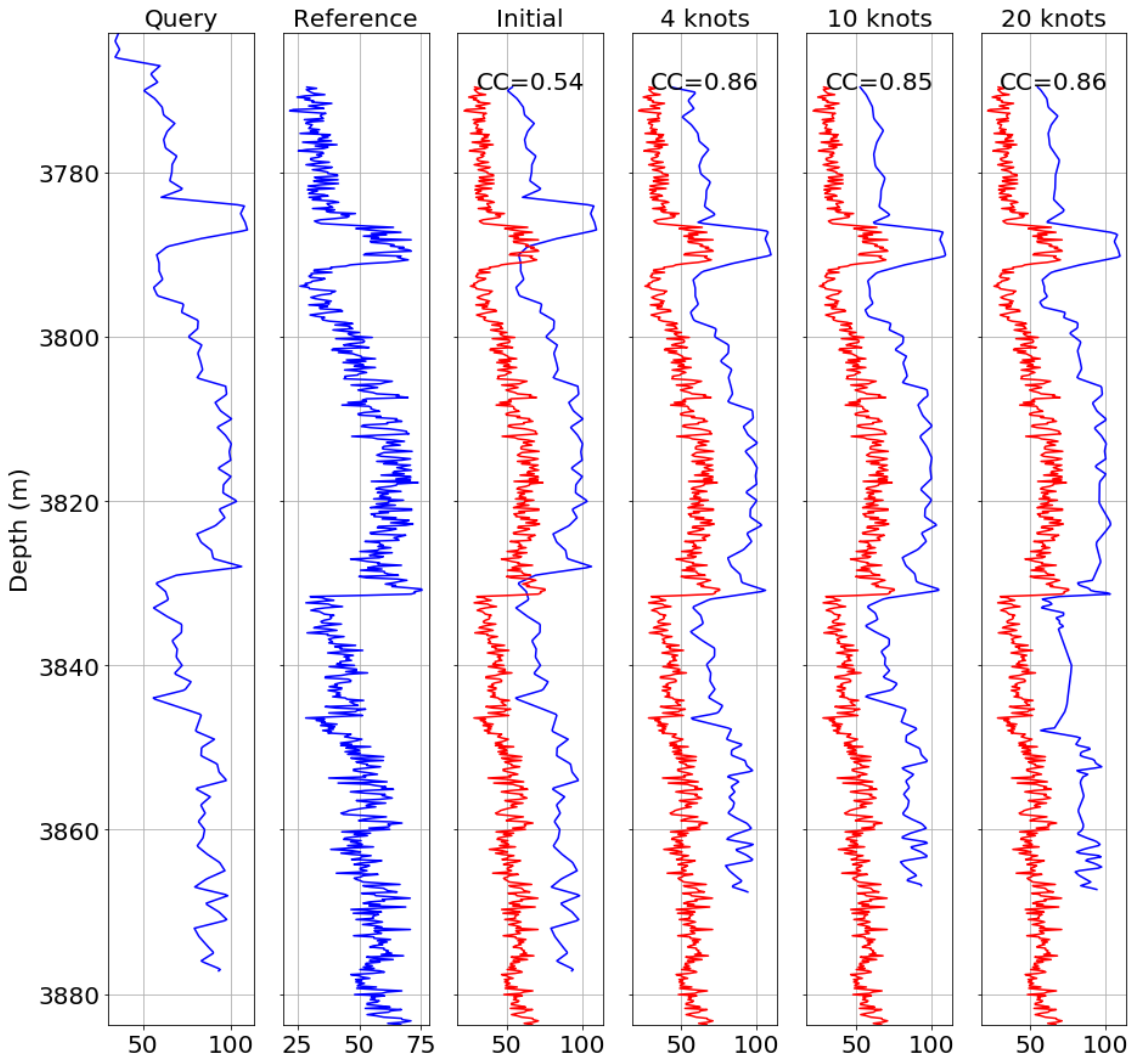


Figure 4.5: From left to right: first the query and secondly the reference. Thirdly the initial warping, fourthly the warping obtained with 4 knots using PLTW, fifthly the warping obtained with 10 knots using PLTW and finally the warping obtained with 20 knots using PLTW. In red the reference and in the top the total cross-correlation are shown.

In consequence, the flexibility requirement of the algorithm depends on the application. For the current use, a small degree of flexibility is preferred as, otherwise, the distortion might be too big. For the current example 4 knots seem to be enough. A standard solution is to define the number of knots based on the length of the signal. After some initial testing, the number of knots was calculated as

$$n = \frac{t_{r,N} - t_{r,1}}{100} + 1. \tag{4.20}$$

The denominator can be adapted if some prior knowledge about the signal is known, but this gave good results for the cases considered. In general, it is better to use fewer than excessive knots, so it is suggested to be conservative with this value.

4.6.4 Knot optimisation

Finally, the PLTW with knot optimisation described in section 4.5 was tested. In figure 4.6 the result obtained with the regular PLTW and with the PLTW with knot optimisation are compared. It can be seen that the latter gives a slightly better result mainly caused by the warping at the first and last intervals.

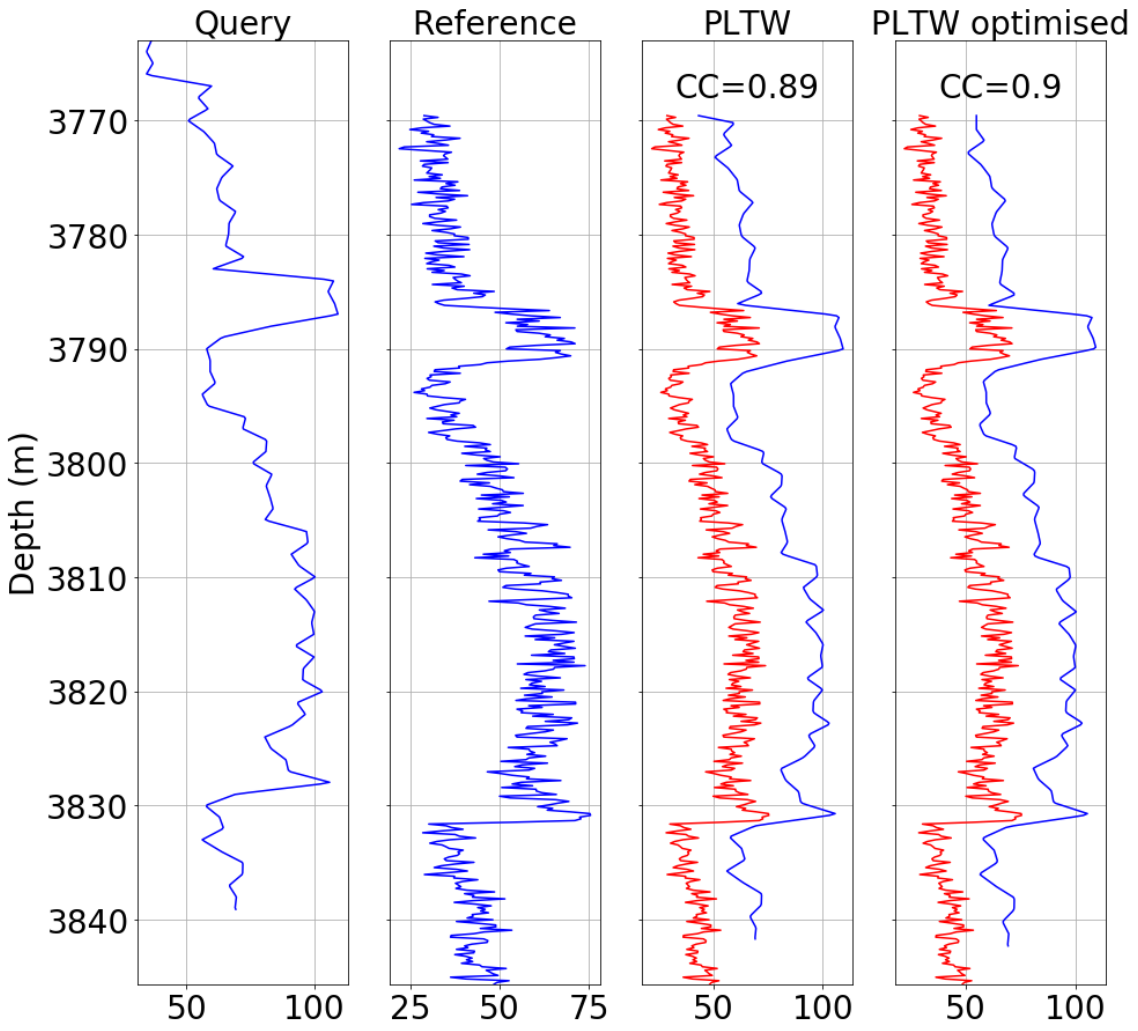


Figure 4.6: From left to right: firstly the query and secondly the reference. Thirdly the PLTW and finally the PLTW with knot optimisation. In red the reference and in the top the total cross-correlation are shown.

Having tested different logs no significant improvement was seen between both methods and the same resulting warping was achieved. This is due to the effectiveness of the regular PLTW, which achieves an almost perfect depth match, so there is no room for obtaining a better result. Thus, both methods give a correct warping and either of them can be used for the automated depth matching process.

In order to compare methods, both were applied on the same logs. In figure 4.7a miscellaneous logs with different number of samples were tested and the resulting final total cross-correlations of both methods on the same logs are shown, as a function of number of samples in the logs. As in the previous example the result is virtually identical although optimising the knots might have provided a minor improvement.

The other important variable to consider is the computational time, the time it takes to the algorithm to give the warped result. In figure 4.7b the computational time needed for each

algorithm is shown as a function of the signal length. Both methods seem to have a similar performance and achieve it in a comparable computational time.

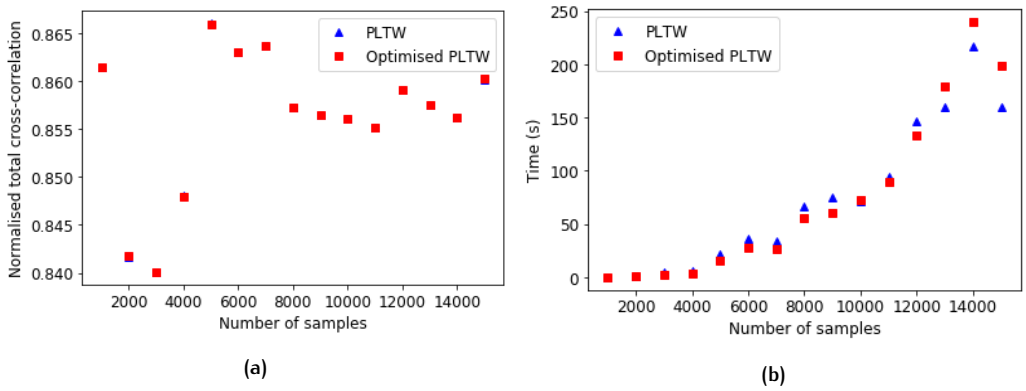


Figure 4.7: (a) Resulting normalised cross correlation for the PLTW and the PLTW with knot optimisation for logs with a different number of samples. (b) Time needed to compute the PLTW and the PLTW with knot optimisation for logs with a different number of samples.

It is remarkable that, in spite of using two times more parameters, the computing time for the knot optimisation PLTW is very similar to the regular PLTW. This is in all likelihood because the extra freedom on the knot location helps to reach the optimum sooner. So, in each iteration the knot locations and projections are updated simultaneously which enhances the speed; therefore, the convergence is reached in fewer iterations than in the regular PLTW.

Although no algorithm is clearly superior, it was decided to use the regular PLTW for the remaining of the project because it is the least flexible warping between both. Optimising the knots simultaneously can make the warping more flexible, so it was preferred to use the simplest version as its behaviour is a priori more predictable.

5 PREPROCESSING TECHNIQUES

In the previous chapter it was shown that the direct implementation of the Piecewise Linear Time Warping (PLTW) suffers from few problems and it is not robust enough to deal with complex situations. This chapter is devoted to introduce some preprocessing techniques in order to enhance the performance of the methodology.

5.1 OFFSET ESTIMATION

One of the first problems that arose was the dependence of the method to the choice of initial parameters; in consequence, making the selection of the initial parameters crucial. The first idea, which was used for the previous calculations, was to assume that the time index from the query was equal to the time index from the reference. Therefore, the warping function was equal to the depth, $w(t) = t$, and the initial parameter set was $\mathbf{a}_0 = \mathbf{t}_{knots}$. However, in the example shown in figure 4.4 this does not give a good initial alignment and is even hard to see the correlation using the naked eye. This misalignment is mainly caused by a constant offset.

Different tools have different reference depths so the 0 meter depth might not be the same; therefore, there might exist a relative offset between both signals. As mentioned before, there are two main misalignment sources: the offset and the linear shifts. If this offset was known, it could be corrected and the only problem left to solve would be to estimate the linear shifts. Furthermore, the offset is usually the most common problem, so fixing it would solve most of the alignment problems, in the simplest depth matching cases.

Offset estimation is a common problem in multiple domains regarding time series and there exist several methods to estimate it [51; 52]. For instance, certain window of the query could be selected and using pattern matching the location of that window in the reference could be found [53]. Another technique can be to use a DTW or another algorithm to align the main features of the curve so that the initial iteration is close to the real warping [25]. Basically, a preliminary warping can be computed to later serve as the initial guess for the algorithm.

In our case, in order to continue using the PTW theory [10], it can be adapted to estimate just the offset. For instance, a small window could be selected where there is presumably only one linear shift and a first degree PTW could be run to estimate the offset properly. If the result is correct, the offset would be precise, as the linear shift has also been considered. However, it can be hard to tell without any previous information how to select this window, so it can prompt some errors.

Another option, the one implemented in this project, is to select a large set of data, even the whole signal, and to apply a zero order PTW. The parametrization in this case is given by $w(t) = t + a_0$ and the method is described by algorithm 5.1. In consequence only one parameter is estimated, so a quick result is obtained with a low computational complexity. Furthermore,

as the whole signal is used the computed offset will maximise the cross-correlation of the whole signal, which is beneficial and helpful for the consecutive steps.

Algorithm 5.1: Offset estimation

Result: Estimated offset

1 initialise a;

2 $a^* \leftarrow \arg\min \left(1 - \frac{r(\mathbf{t})^H \cdot s(\mathbf{t}+a)}{\sigma_r \sigma_s}\right)$;

Once the offset is computed this information is passed forward to the Piecewiese Linear Time Warping (PLTW) so that the total warping is calculated. The initial parameters will simply be $\mathbf{a}_0 = \mathbf{t}_{knots} + t_{Offset}$. The same sample as in figure 4.4 is shown in figure 5.1 but first a pre-alignment is done using the described technique. It can be seen that the result has significantly improved and the main features are well aligned increasing the overall cross-correlation.

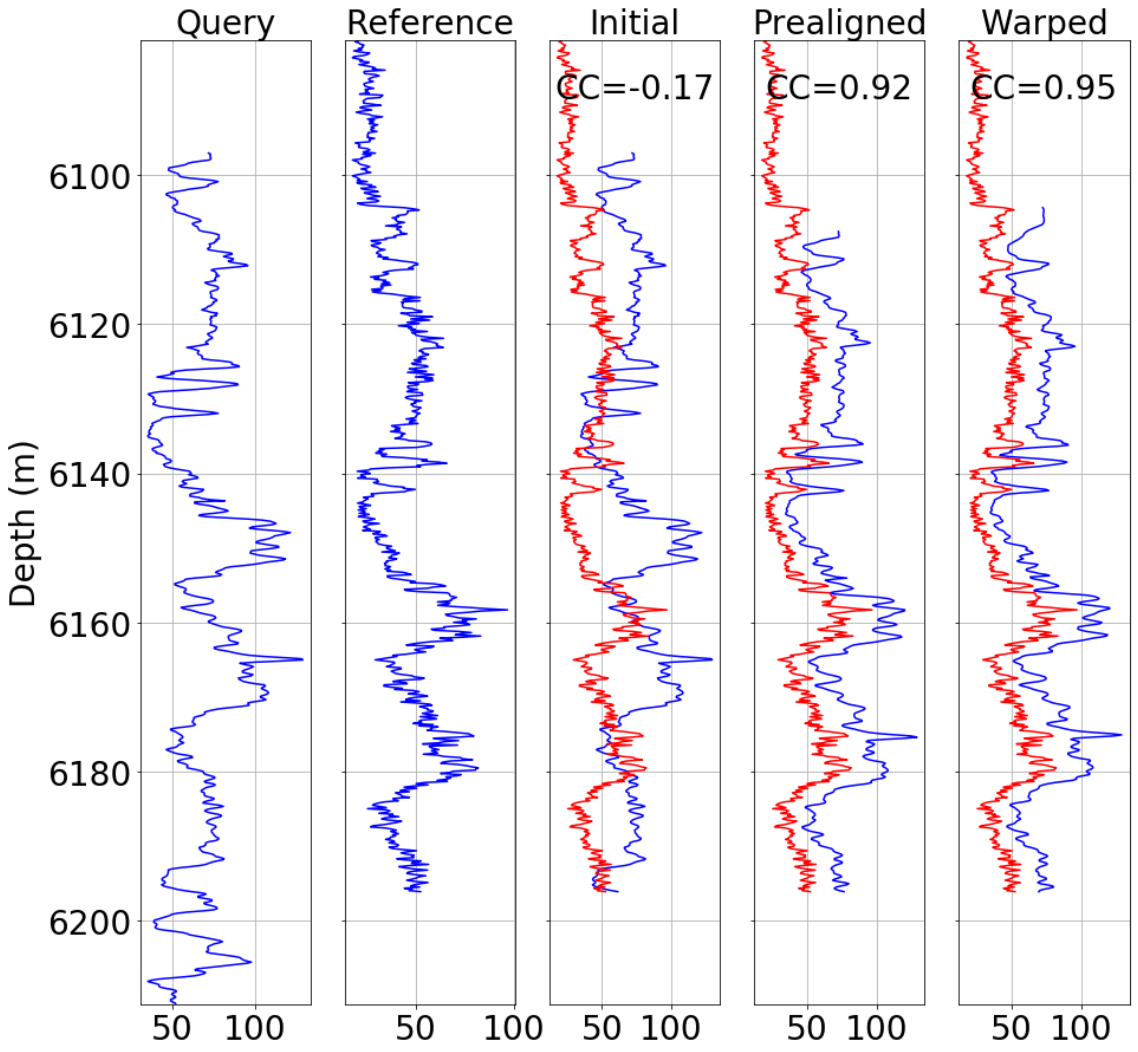


Figure 5.1: From left to right: firstly the query and secondly the reference. Thirdly the warping before the prealignment, fourthly the warping after the prealignment and finally the warping obtained with the Piecewiese Linear Time Warping. In red the reference and in the top the total cross-correlation are shown.

In this case the curves are already well aligned from the prealignment and only minor changes are needed as it can be concluded by the small improvements the PLTW has on the cross-correlation. For all the logs tested the results were good, the offset was estimated and the

improvement from prealignment was significant. However, the problem of convergence to a local optimum is still an issue to consider because the solution is obtained by iterative minimization.

5.2 FREQUENCY ANALYSIS

Another property to consider for preprocessing is the signal frequency. When aligning the data from Logging While Drilling (LWD) and the wireline logging it is clear that the latter has a higher sampling rate, so it contains higher frequencies. Furthermore, the features to align usually have lower frequencies than the maximum contained in the curve. The high frequencies are important and must be preserved. Nonetheless, during the search for the warping function these frequencies can cause issues and complicate the process both in the manual warping as well as in the PLTW. In the latter, this high frequency data can create additional local optimums resulting in convergence to a suboptimal solution. Therefore, it makes sense to only use the low frequencies for the warping and to add the extra information at the end. The simplest way to filter out high frequencies is to apply a low pass filter on the original data.

The first step was to represent logs in the frequency domain. The original signal was segmented in windows and each of them was transformed to the frequency domain using a DFT (Discrete Fourier Transform). This is a broadly used technique, for instance in audio processing [54]. In figure 5.2 the original curve and its representation in the frequency domain are shown. In the latter the absolute value for each frequency at each window is shown in dBs and an overlapping of 2 as well as zero padding are used in order to have a clearer result. Each window consists of 20 samples without any windowing function and the sampling frequency of the curve was of 6.6 $\left[m^{-1}\right]$.

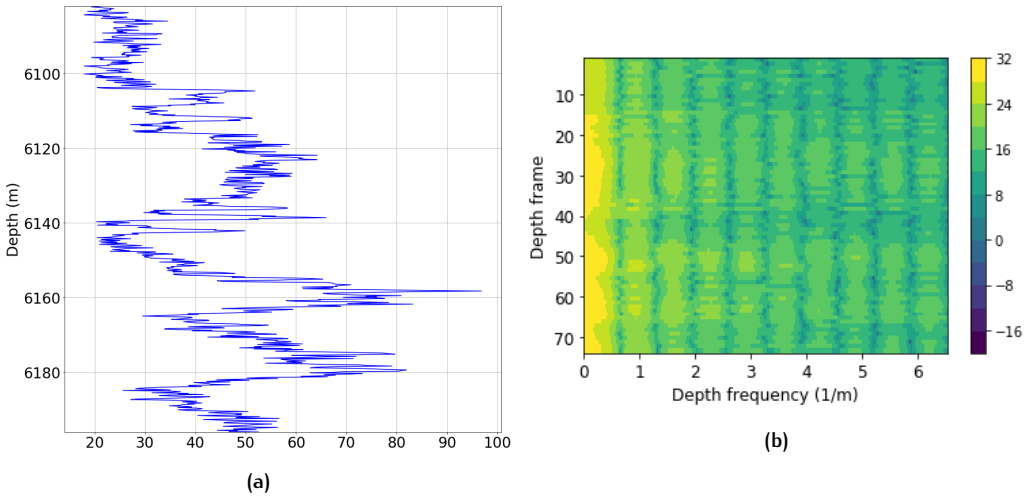


Figure 5.2: (a) Sample of a log on the (depth) time domain (b) Sample of a log on the frequency domain where the absolute value on dBs is shown.

It is clear that the curve mainly consists of low frequencies. Frequencies higher than $0.5m^{-1}$ seem to provide little important information for the alignment. Thus, these frequencies can be removed or suppressed to ease the alignment process. There are multiple ways for removing those frequencies, the simplest is to apply a low pass filter. In figure 5.3 a Butterworth filter has been applied using different cutoff frequencies to see how they affect the curve.

In the most left the original signal is shown followed by the filtered results using the specified cut-off frequency. The highest the cut off frequency clearly reduces the 'noise' but still much extra information is present. The limit is around $0.5m^{-1}$, with that cut off frequency for the low pass filter only significant features for the alignment are left. However, if the cutoff frequency is

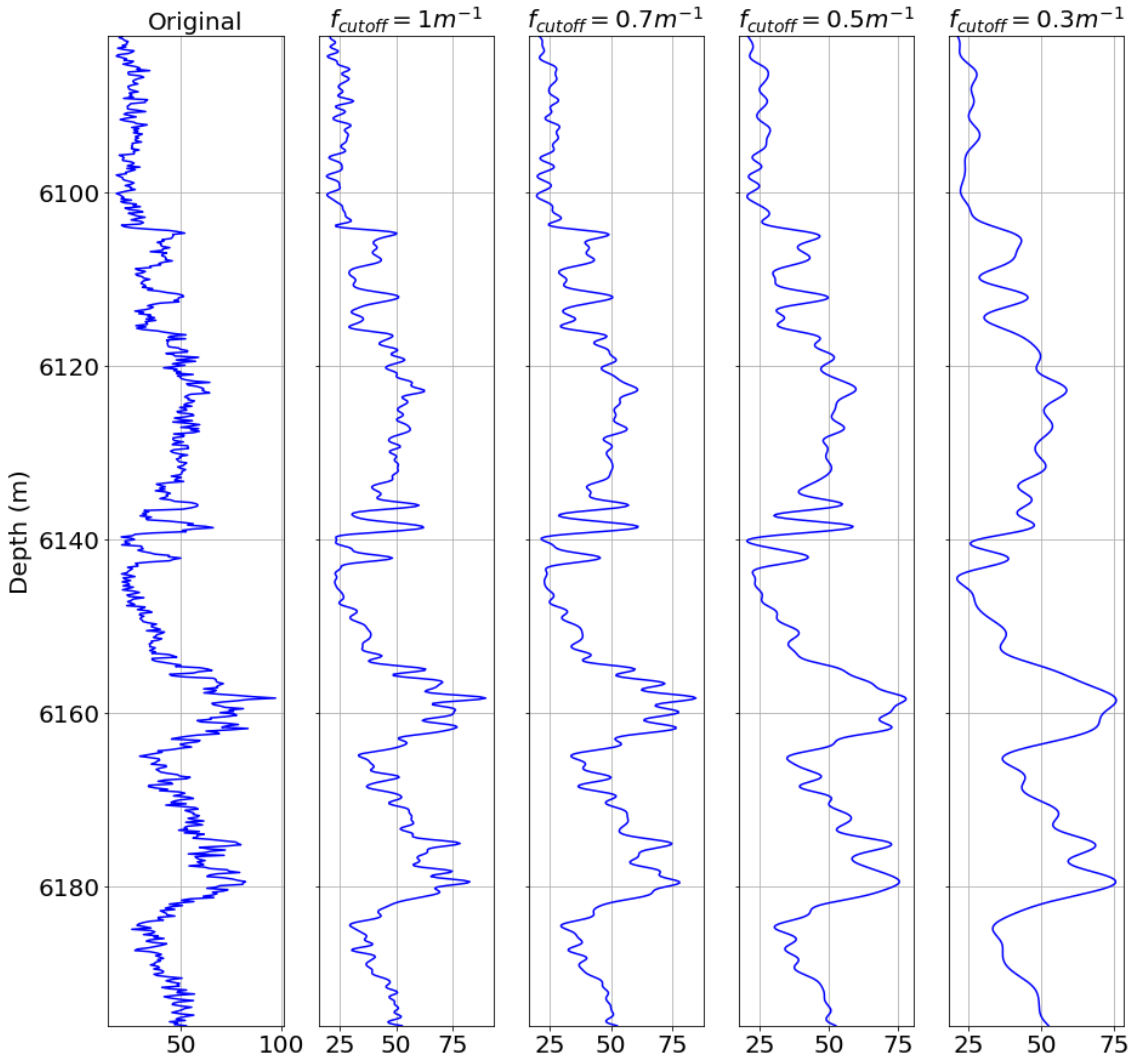


Figure 5.3: In the left the original log is plotted. Subsequently the results of applying a Butterworth low pass filter with different cutoff frequencies are shown.

lower than that, some features present in the signal might suffer significant changes. Based on that observation two cases can be considered.

In the first case, there is one frequency, $f_{preserve}$, that denotes the lowest cut-off frequency that preserves the main features. Cut off frequencies higher than this value will decrease the number of local optimums in comparison to the original signal and will not affect the major feature, being the optimum to use $f_{preserve}$ as the cut-off frequency.

The other situation is when lower frequencies that the ideal threshold are removed. If this is the case, the main features will start to merge, so the cost function will have even fewer local optimums. Thus, the warping will be able to move between major peaks allowing more flexibility. Nonetheless, this merging also means that the curves have been significantly deformed, so it must be used carefully because with strong deformations the results will not be reliable.

5.2.1 Filtering effect in the cross-correlation

It is important to understand how the low pass filtering affects the algorithm. This was checked by first filtering both the query and the reference signals using certain low pass filter and com-

puting the normalised cross-correlation at different delay values, which are shown in figure 5.4. This function was calculated using

$$\hat{\phi}_{rs}(\tau) = \int_{t\in\mathcal{D}} \frac{r_{filtered}(t)^* s_{filtered}(t+\tau)}{\sigma_{r_{filtered}}\sigma_{s_{filtered}}} dt = \frac{r_{filtered}(\mathbf{t}_r)^H s_{filtered}(\mathbf{t}_r+\tau)}{\sigma_{r_{filtered}}\sigma_{s_{filtered}}}, \tag{5.1}$$

so it can be seen as the cost function used in the zero order PTW used for estimating the offset. Thus, the delay, τ, that maximises the function will be the estimated offset. Nonetheless, it is necessary to consider that gradient based methods are used to solve the problem and the initial point is a 0 meter delay.

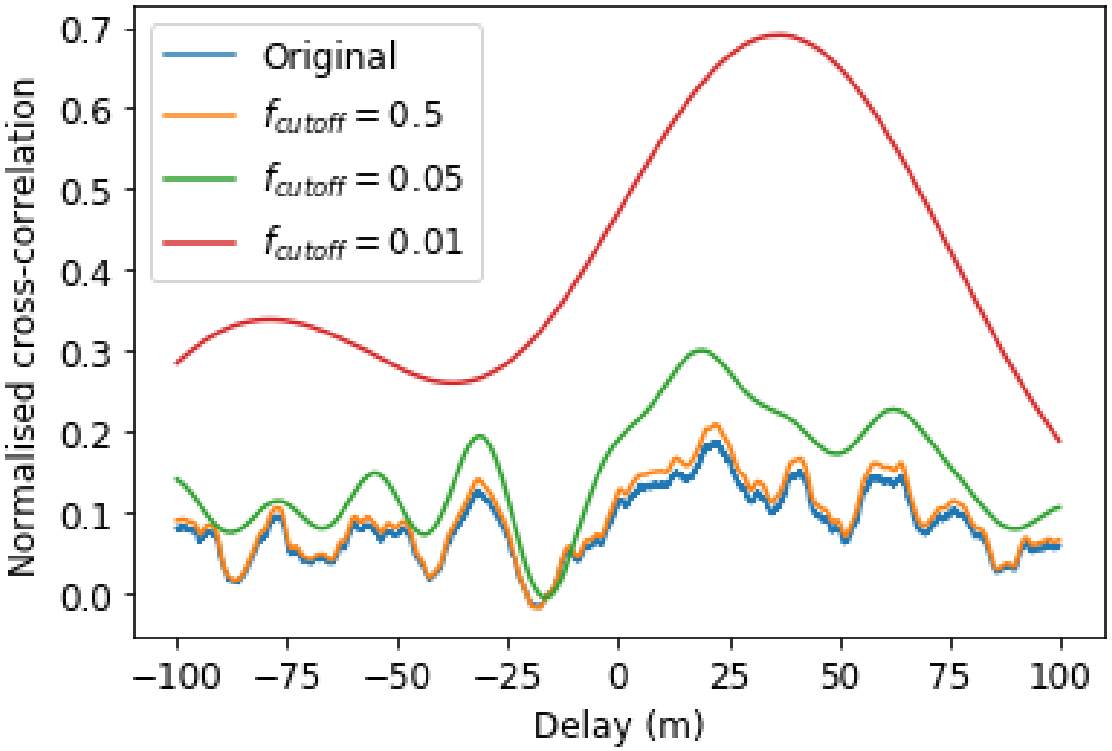


Figure 5.4: The normalised cross-correlation is calculated with different delay values. Four functions are shown, the first without any filtering and the others where both the query and the reference were previously filtered.

The original cross-correlation function has plenty of irregularities, so small peaks and valleys, which are the mentioned local optimums the problem can converge to. This is caused by the small features coming from both curves; therefore, if the query and reference are slightly smoothed these small irregularities will disappear but the function is almost identically preserved, as it is seen when a cut-off frequency of 0.5 m^{-1} is used.

Furthermore, when the low pass filter is applied the overall cross-correlation seems to increase. This is because the low pass filter takes neighbouring values into account an provides a result for each point, so it performs a weighted average. Therefore, the result will be more correlated because the adjacent values have been used, so the overall cross-correlation will indeed grow.

Heavier smoothing alters the cost function and the contribution from the different main features are combined together giving a different curve. This can be seen with the cut-off frequencies of 0.05 m^{-1} and 0.01 m^{-1}. In the above the optimum is close to the right delay while with the latter the optimum is significantly displaced. In conclusion, heavy smoothing allows to

avoid converging into a local optimums but the results might vary from the true offset, due to original signal deformation.

5.2.2 Implementation

Considering the effect low pass filtering provides, it was added to both the offset estimation and the PLTW.

The offset correction designed in section 5.1 is meant to give the optimum offset, that is the most beneficial for all the different warping sections. Thus, based on figure 5.4, it was decided to apply a low pass filter to both the query and the reference with a cut-off frequency of 0.05 m^{-1}. The outcome will be feed to the PLTW and will ease the computation of warping function and reduce the error.

However, it is obvious that the warped result cannot be directly given as the outcome because it only contains the low frequencies. To obtain the final warping the estimated $\mathbf{a}^*$ parameters are used to compute the estimated warping function, $w(t)$. Afterwards, the original query is warped according to that function and interpolating the points. The whole process is described in algorithm 5.2.

Algorithm 5.2: Filtered PLTW

Result: Filtered PLTW

initialise $a_{offset,0}$ and $f_{cutoff,offset}$;

low pass filter $\mathbf{s}$ and $\mathbf{r}$ using $f_{cutoff,offset}$;

$a^*_{offset} \leftarrow \arg\min \left(1 - \frac{\mathbf{r}_{filtered}(\mathbf{t})^H \cdot \mathbf{s}_{filtered}(\mathbf{t}+a_{offset})}{\sigma_r \sigma_s}\right)$;

construct $\mathbf{t}_{knot}$;

$\mathbf{a}_0 = \mathbf{t}_{knot} + a^*_{offset}$;

initialise $f_{cutoff,PLTW}$;

low pass filter $\mathbf{s}$ and $\mathbf{r}$ using $f_{cutoff,PLTW}$;

$\mathbf{a}^* \leftarrow \arg\min \left(1 - \frac{\mathbf{r}_{filtered}(\mathbf{t})^H \cdot \mathbf{s}_{filtered}(w(\mathbf{t},\mathbf{a}))}{\sigma_r \sigma_s}\right)$ with constraints;

$\mathbf{s}_W = \mathbf{s}\left(w\left(\mathbf{t}, \mathbf{a}^*\right)\right)$;

In figure 5.5 the results are shown for one of the logs from the dataset. Again the improvement the offset correction provides is significant and it solves most of the misalignment as the resulting cross-correlation is of almost 0.9. Afterwards, the PLTW just corrects some local misalignment and slightly improves the cross-correlation. The resulting warping is comparable to that an experienced petrophysicists would provide.

In conclusion, it is clear that this method can solve misalignment issues where the main source is a constant offset and the higher order shifts are minor. This is the case for the majority of depth matching situations. Nonetheless, it is relevant to understand how robust the PLTW is to solve problems where also linear shifts are also present. This is shown in figure 5.6 where some artificial linear shifts have been introduced to the query to test the approach. The first peak was widened and adapted to the reference properly and the last valley was moved matching the reference. The cross-correlation obtained using this method was of 0.88 while the cross-correlation of the aligned signal without artificial shifts was of 0.9, so the algorithm was able to give a correct warping despite the linear shifts.

Performed experiments show that the PLTW can retrieve signals with some level of linear shift. In these cases the offset correction will just help to get the features closer; therefore, avoiding converging to local optimums. Nonetheless, it is clear that there is a limit to the capacity to estimate the warping function correctly. If the samples have suffered strong distortion the solution might be incorrect.

Different cut-off frequency values were tested to see how they influenced the result. Small variations to the default values where made and no significant changes where observed. This is

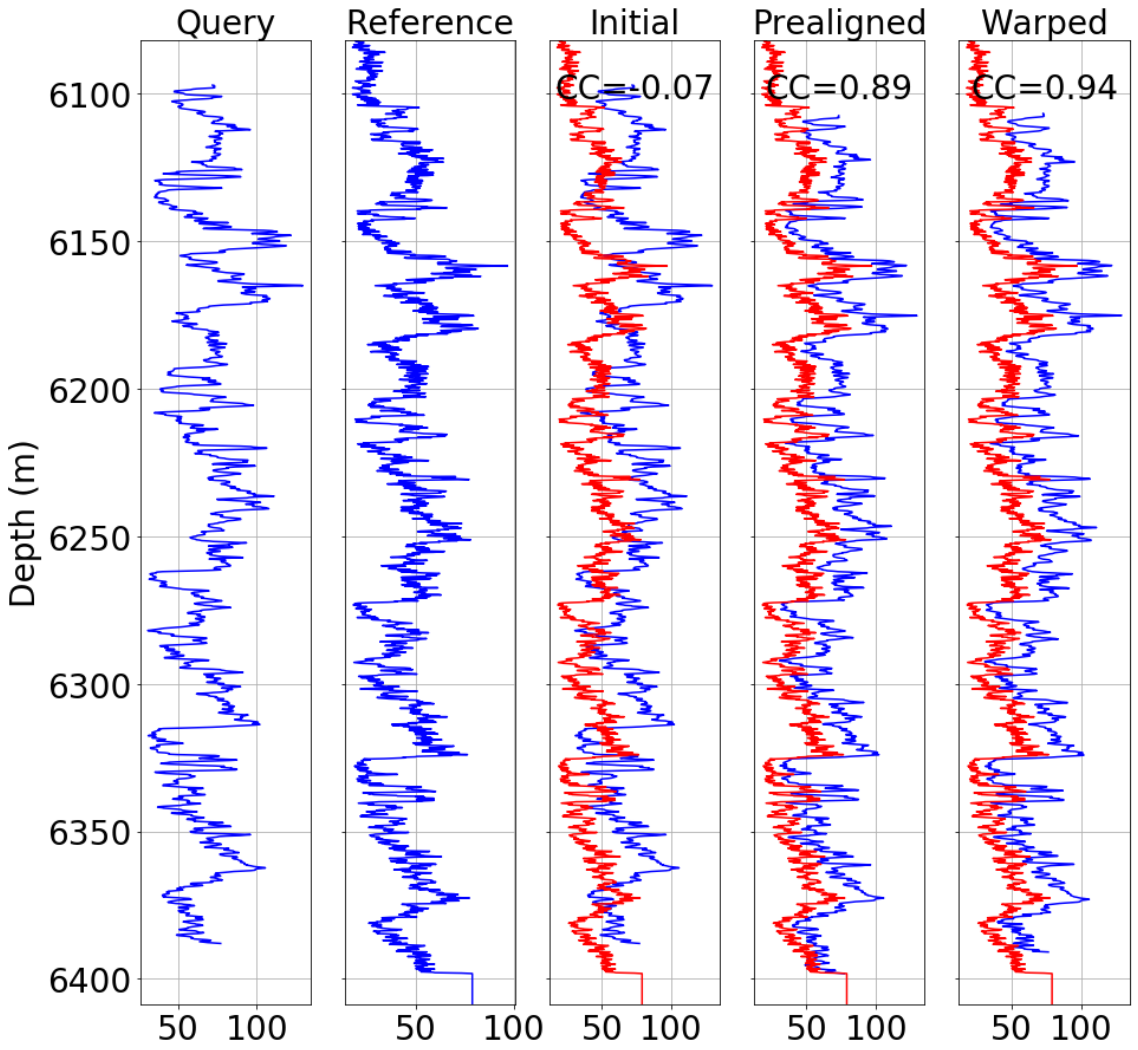


Figure 5.5: From left to right: firstly the query and secondly the reference. Thirdly the warping before the prealignment, fourthly the warping after the prealignment and finally the warping obtained with the Piecewiese Linear Time Warping. In red the reference and in the top the total cross-correlation are shown.

because these values are defined in the frequency domain and they are related to the properties of the main features to align, so they do not depend on factors such as the sampling rate or the measurement length.

Finally, this method only requires one input, the number of knots. In the previous chapter its value and properties have been discussed as well as providing a default value to use. Thanks to the offset correction, the initial alignment will be better so a higher number of knots can be used without the risk of creating artifacts. Nonetheless, in exchange for providing a more accurate warp, the computational complexity will also increase. Thus, it is suggested to use the same value as defined in chapter 4.

5.3 COMPUTATIONAL EFFICIENCY

Having seen that the PLTW is an effective method, some analysis regarding its computational efficiency were performed. The goal was to check how the algorithm would perform with logs

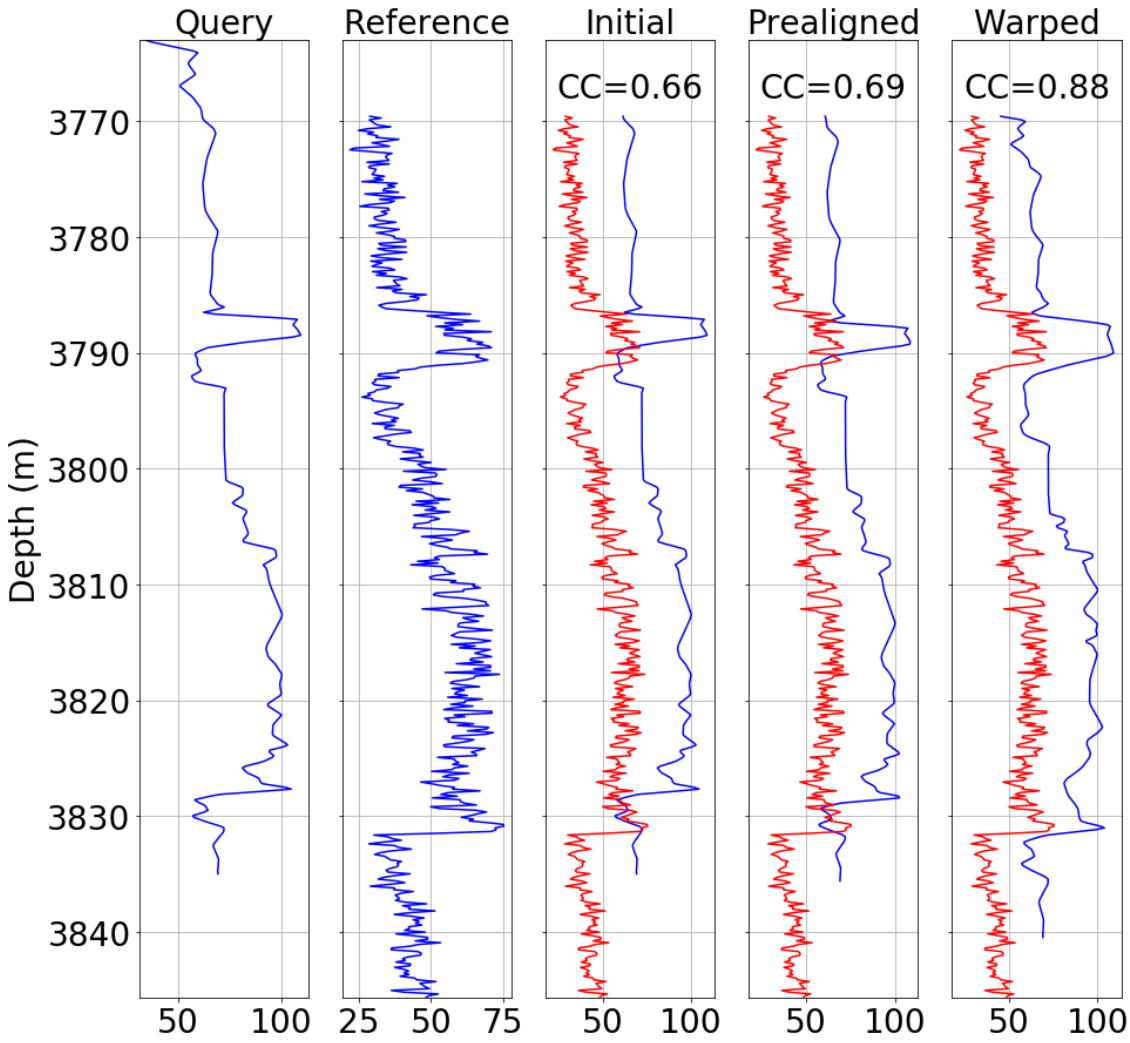


Figure 5.6: From left to right: firstly the query and secondly the reference. Thirdly the warping before the prealignment, fourthly the warping after the prealignment and finally the warping obtained with the Piecewise Linear Time Warping. In red the reference and in the top the total cross-correlation are shown.

of different sizes in order to analyse its scalability and applicability. The default strategy is to increase the number of knots when the total depth the signal covers increases as shown in equation 4.20. However, since both the number of samples and the number of knots will increase it is not clear which would be the cause; therefore, in figure 5.7 the default PLTW and a PLTW with 10 fixed knots regardless of the size of the signal are shown. The sampling period of the signal was 0.15 m, so the total depth of the signal can be computed using it. Hence, when the number of samples in figure 5.7 increases, so does the total depth and a new section is added and concatenated to it. The adjusted knots curve represents the time it took to converge using the number of knots given by equation 4.20 while the fixed number of knots represents the computational time for a PLTW using 10 knots.

When the knots are fixed the time consumption grows linearly, $\mathcal{O}(n)$. On the other hand, with the default procedure the time consumption does not grow linearly, but it rather shows a polynomial growth, $\mathcal{O}(n^{\alpha})$. This fact carries a scalability problem, but it is important to define a total threshold time, so the maximum time the algorithm should take to give the result. The idea is to implement such algorithms in the user tool, that can be executed before analyzing measurements in the well. In consequence, it is important to have the result quickly so that

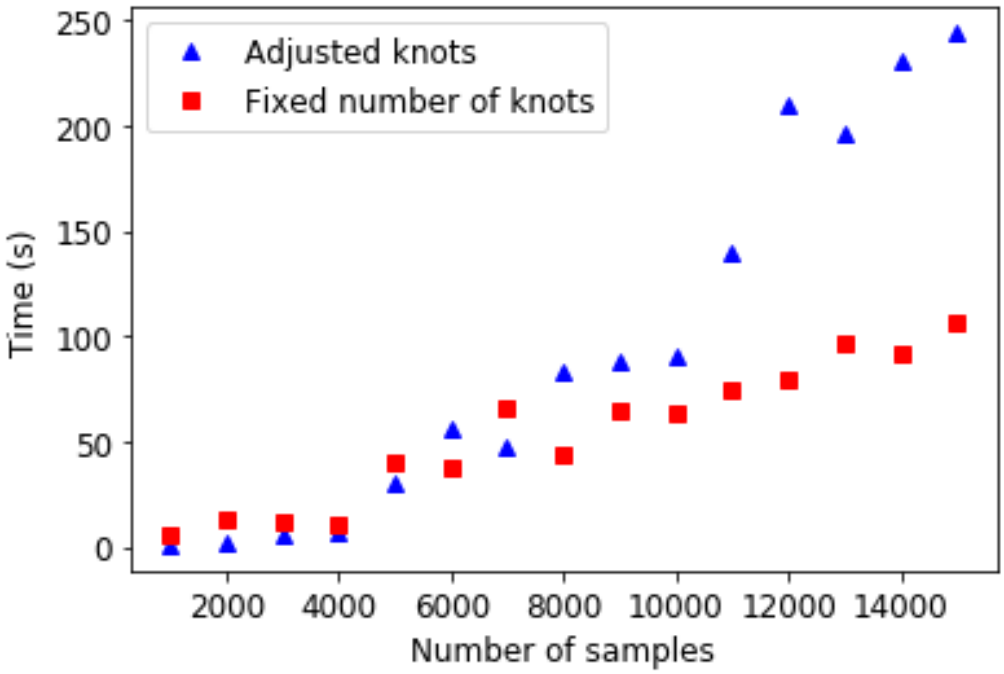


Figure 5.7: Computational time using the regular PLTW "Adjusted knots" and using a fixed number of 10 knots "Fixed number of knots" regardless of the size of the signal.

the user does not need to wait for too long, establishing the ideal maximum threshold at the order of seconds. Thus, the current algorithm has a reasonable performance for a low number of samples, below 5000 elements, but a higher number of samples may increase the computational time beyond desired.

One solution to tackle the problem in this particular case is to consider the frequency analysis done previously. It was seen that using low pass filters is beneficial if done correctly. It was seen that only low frequencies are important and due to this fact the use of low pass filters was encouraged. A further step is to undersample the curves, so to only pick some samples from the whole signal. This will obviously bring a loss of information but it might not be crucial. Another option is to apply a low pass filter to the raw signal and undersample the result. In this case the filtering would act as an average so that the points from the curve are more correlated and the potential impact undersampling can have is reduced. Low pass filtering and undersampling the data is a common procedure for data matching [55]. The PLTW will be applied to the undersampled signals to obtain the warping function, $w(t)$, and the warped query will be obtained using the raw data, $s(w(t))$.

In figure 5.8 the outcome cross-correlation of the mentioned cases is shown with different undersampling levels. The result obtained using no undersampling is also given for comparison. The sampling reduction shown in the horizontal axis is related to the undersampling degree, so the employed sampling will be the initial sampling frequency multiplied by the sampling rate reduction, so a sampling rate reduction of 2 means that only half of the total data points are used.

The first conclusion is that filtering does not seem to significantly improve the performance of the undesampling. However, in general, filtering does give the best result for the previously mentioned reasons, so since it is a virtually costless operation it is worth preserving it. In this case it was decided to use a low pass filter but using a moving average might provide a similar result [55], so either of them can be used.

The other conclusion is that the performance results can be divided into two trends. Up to a sampling rate of 5 small to none decrease in the performance is seen while after that threshold

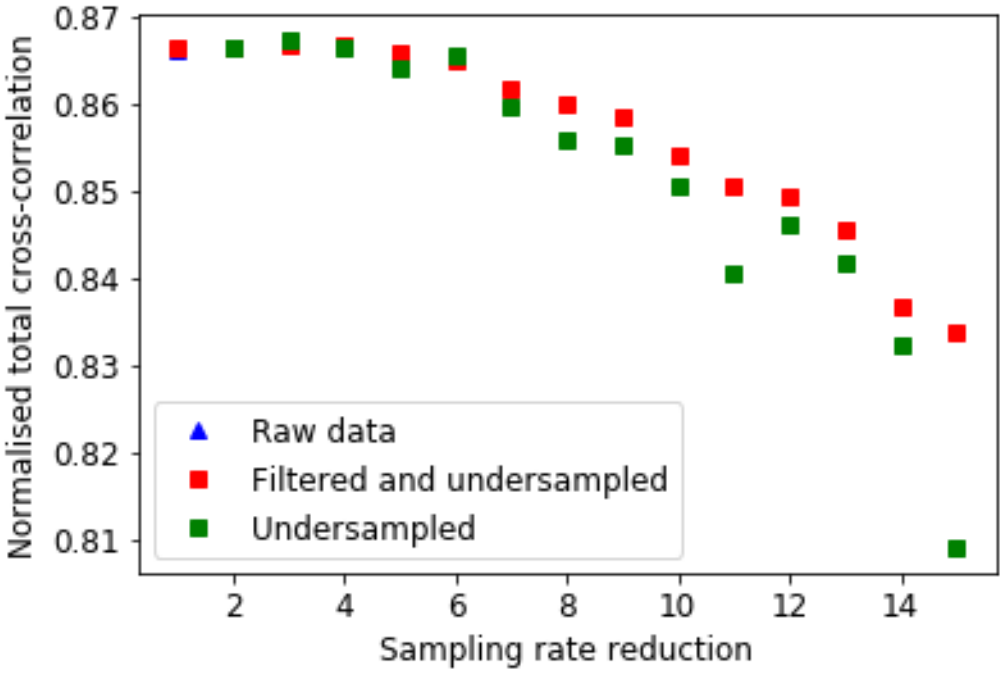


Figure 5.8: The blue triangle represents the resulting normalised cross-correlation obtained using the regular PLTW to the whole sample. The red square represents the resulting normalised cross-correlation if after filtering the data this is undersampled at different rates and the green square the result if no filtering is done.

a linear decrease can be observed. Therefore, for the dataset under consideration it can be stated that a low undersampling will not affect the result and it can be used in order to speed up the algorithm. This value will obviously depend on the sampling rate at which the original measurements were taken. If the original value is f_s its value after the undersampling will be $f_{undersampled} = \frac{f_s}{u_{rate}}$ where u_{rate} is the sampling rate reduction. The limit will be given by the new sapling rate $f_{undersampled}$. The threshold can be obtained from the frequency analysis at section 5.2. The minimum $f_{undersampled}$ can be set at $1m^{-1}$ so the maximum sampling reduction will be $\frac{f_s}{f_{undersampled,min}} = f_s$. For the used data the sampling period was of $0.15m$, so the sampling rate was of about $6.6m^{-1}$; therefore, the maximum sampling rate reduction is close to 6.

The other observation is regarding higher sampling rate reductions. In figure 5.8 a linear decrease trend is seen, so, as it could be expected, dropping too many samples will affect the precision and correctness of the result. Nonetheless, judging by the effect on the normalised cross-correlation, the result is not catastrophic and analysing the result by visual inspection it can be seen that the main features are relatively well align and there are just slightly misaligned. Therefore, although the result would be recognisable, small misalignments would be present in the result, and this error would increase with the strength of the undersampling used.

In figure 5.9 the computational time for warping signals with a different number of samples are compared to see the advantage undersampling provides. Up to 4000 samples the time is very similar, but for bigger samples this technique is extremely helpful, and it enables the use of the PLTW with long log sections without affecting the accuracy of the result.

In conclusion, the PLTW is based on an optimization problem which, in the default performance, will show a non-linear complexity. Since the curves are oversampled for alignment purposes, the number of samples can be reduced. This decreases the computational time of the automatic depth matching and allows to use it for any pair of logs.

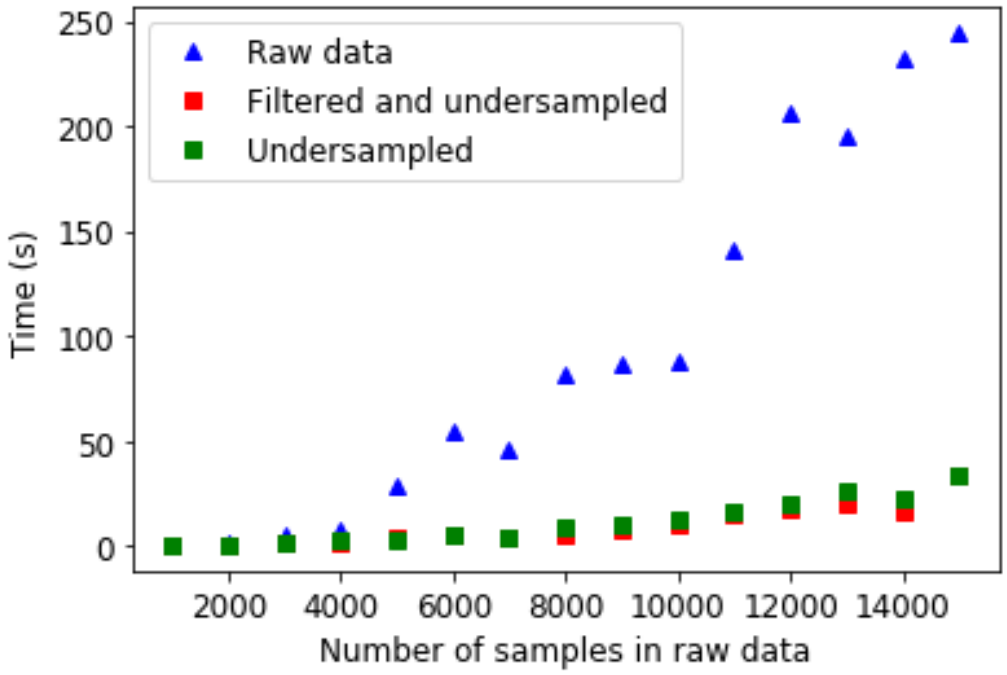


Figure 5.9: Computational time of warping signals of different lengths for the raw data, undersampled and filtered and undersampled versions of the raw data. A sampling rate reduction of 5 was used, so only a fifth of the data is used.

The final method, which incorporates all the mentioned techniques, is expressed in algorithm 5.3.

Algorithm 5.3: Improved PLTW

Result: Warped query

initialise $a_{offset,0}$ and $f_{cutoff,offset}$;

low pass filter $\mathbf{s}$ and $\mathbf{r}$ using $f_{cutoff,offset}$;

$a^*_{offset} \leftarrow \arg\min \left(1 - \frac{\mathbf{r}_{filtered}(\mathbf{t})^H \cdot \mathbf{s}_{filtered}(\mathbf{t}+a_{offset})}{\sigma_r \sigma_s}\right)$;

construct $\mathbf{t}_{knot}$;

$\mathbf{a}_0 = \mathbf{t}_{knot} + a^*_{offset}$;

initialise $f_{cutoff,PLTW}$;

low pass filter $\mathbf{s}$ and $\mathbf{r}$ using $f_{cutoff,PLTW}$;

undersample $\mathbf{t}$ to $\mathbf{t}_{und}$;

$\mathbf{a}^* \leftarrow \arg\min \left(1 - \frac{\mathbf{r}_{filtered}(\mathbf{t}_{und})^H \cdot \mathbf{s}_{filtered}(w(\mathbf{t}_{und},\mathbf{a}))}{\sigma_r \sigma_s}\right)$ with constraints;

$\mathbf{s}_W = \mathbf{s}\left(w\left(\mathbf{t}, \mathbf{a}^*\right)\right)$;

5.4 COMPARISON WITH MANUAL DEPTH MATCHING

Finally, the PLTW with the mentioned additions was tested using a real example and was compared with the warping a petrophysicist would do manually. This comparison is shown in figure 5.10. After a close inspection of the warping results by an experienced petrophysicist it was concluded that both results, manual and with the PLTW, were virtually identical.

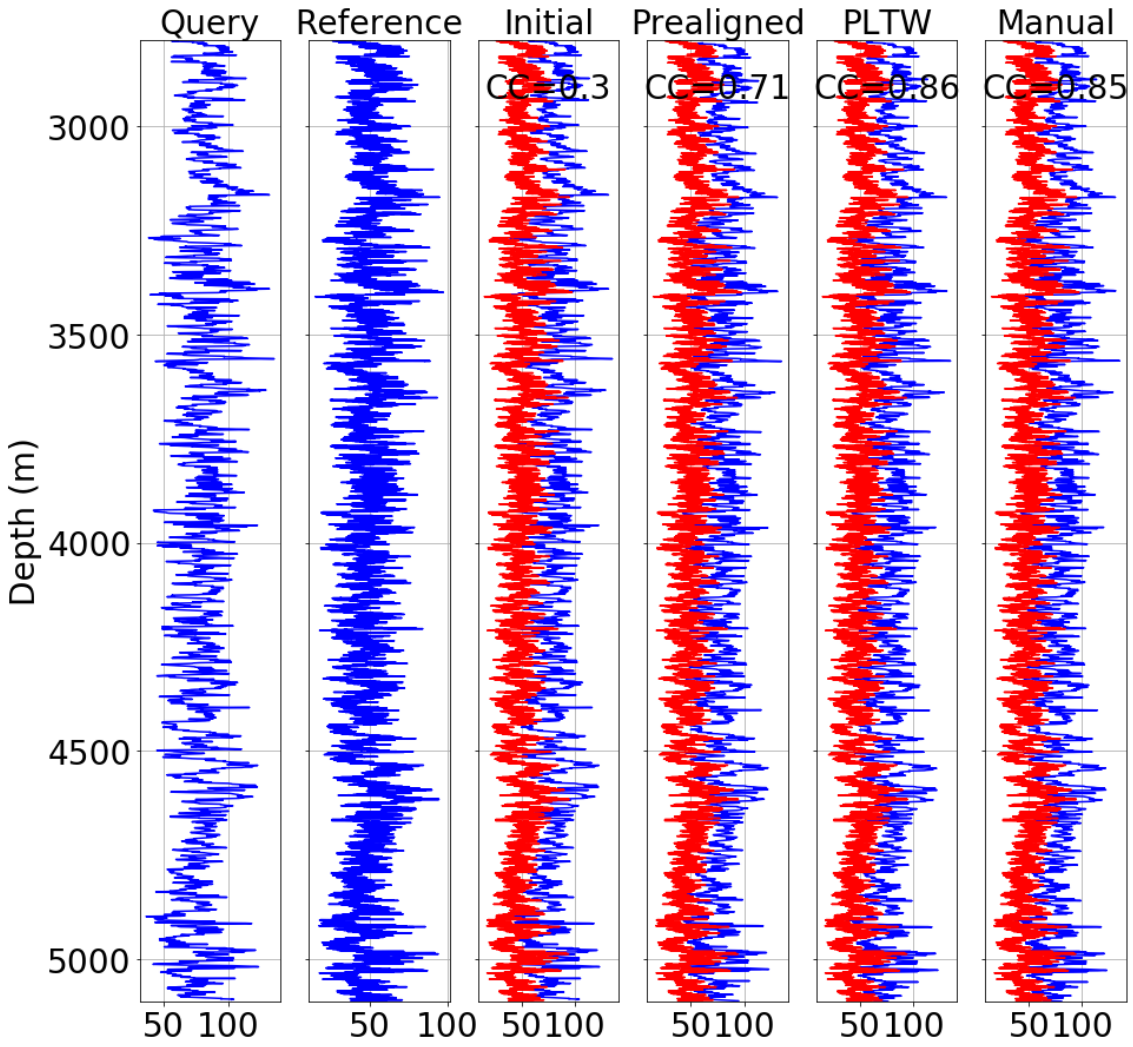


Figure 5.10: From left to right: firstly the query and secondly the reference. Thirdly the warping before the prealignment, fourthly the warping after the prealignment, fifthly the warping obtained with the Piecewiese Linear Time Warping and finally the warping obtained manually by a geologist. In red the reference and in the top the total cross-correlation are shown.

The example shown mainly suffers from a constant offset, which is first corrected to give a total cross-correlation of 0.71. In spite of this value being relatively high, there are linear shifts present as well on the result because the cross-correlation obtained manually is 0.85. This details are aligned by the PLTW which obtains an outcome with a cross-correlation of 0.86, slightly better than manual depth matching. So, the algorithm works correctly with real examples and can provide similar results as obtained by experienced petrophysicists.

Furthermore, the main advantage of automating the process is the time saving for a petrophysicist, which can focus instead on the analysis and interpretation of the data. Depth matching manually the example shown in figure 5.10 took about an hour for an experienced professional while the algorithm did it in less than half a minute giving the same outcome. The PLTW can; thus, significantly improve the current depth matching workflows.

6 ITERATIVE ALGORITHM

Using the developed methodology and applying preprocessing techniques the PLTW can be applied to any log and it will give good results when soft or mild distortion is present. However, it will suffer with strong distortions and in this chapter a solution to extend the method to more complex alignments is given by applying the method in a iterative fashion.

6.1 PIECEWISE LINEAR TIME WARPING WITH KNOT ITERATION

In section 4.6.3 the influence of the number of knots was tested. It was seen that the more knots used, the more flexible the warping is. Therefore, it is better to use few knots in very distorted signals because fewer artifacts would be present in the final result. On the other hand, if the two signals are already relatively well aligned, using more knots will be beneficial because local mismatches will be better corrected. Thus, it is sensible to use a cascade approach starting with few knots and adding new knots in iterative fashion. So, first the warping is done with few knots, the result is obtained and it is used as the starting alignment for another warping using more knots. In consequence, the result will be better aligned than the previous and the usage of more knots without creating artifacts will be possible.

The whole procedure is as follows. Having computed the offset, first a low pass filter will be applied for the reasons already mentioned in section 5.2. Afterwards, the first iteration is done using just two knots, so a global linear stretching or squeezing is calculated. The outcome will be two parameters, the two projections from the reference time into the query at the edges. From this information more parameters can be estimated. The simplest expansion is to increase the knots to 3, so the two edges and an extra knot in the middle can be used. The value of the latter will be the computed using the resulting piecewise linear function defined by the known projections, **a**. The knot will be located in the reference time, so $t_{knot,i}$ is known. In this case, it will be the time in the middle, $\frac{t_{knot,n}-t_{knot,1}}{2}$. Once this value is known its projection, $a(t_{knot,i})$, is computed using linear interpolation, so

$$a(t_{knot,i}) = w(t_{knot,i}) = f_{BL}(t_{knot,i}) = a_j + \left(t_{knot,i} - t_{knot,j}\right) \frac{a_{j+1} - a_j}{t_{knot,j+1} - t_{knot,j}}, \tag{6.1}$$

where $t_{knot} \in \left[t_{knot,j}, t_{knot,j+1}\right]$.

Having computed the projection of this extra knot, the PLTW is launched one more time and a set of three parameters will be returned as the result. Subsequently the same can be done and the two segments can be divided and two more projections that will be introduced in the next iteration. Therefore, if this process is done iteratively, the number of knots at certain iteration will be

$$n[i+1] = 2 \cdot n[i] - 1, \tag{6.2}$$

considering that the first iteration, $n[1]$, has 2 knots.

This was the first idea and the most basic implementation of the knot iteration but it has some drawbacks. The most significant is that the knot number will increase exponentially, which

would affect the computational time and the result. Due to this it was decided to use a more moderate increase ratio by adding some fixed number of knots after each iteration, $n_{increase}$. In consequence, the number of knots can be calculated in an iterative fashion with

$$n[i+1] = n[i] + n_{increase}. \tag{6.3}$$

While in the first approach the location of the knots is fixed once they are introduced, in the latter this is not the case and the values change from one iteration to the next. This is not a priori negative because each iteration is an approximation, so there is no reason to believe that the calculated projection of the knots is correct. It is only known that those are the parameters that minimise the current optimisation problem, so not preserving them and using others which are calculated by interpolation for the next iteration is not per se bad. In theory, if the number of iterations is big enough it will indeed converge to the correct solution. Thus, there is no reason why the knot location must be preserved from one iteration to the next.

Nonetheless, it is hard to tell the optimum value of $n_{increase}$ as it might depend on the signals themselves. In order to have a robust calculation the increase will depend on the total depth covered by the reference signal; therefore, being

$$n_{inc} = \frac{t_{r,N} - t_{r,1}}{300}. \tag{6.4}$$

This value was established after some testing with a number of logs. As in equation 4.20, the denominator will define the result, so this value could be adjusted depending on the expected distortion the signals have suffered. The harder the alignment problem the bigger the denominator should be and the smaller the increase. For the datasets considered in this thesis n_{inc} was defined as in equation 6.4.

It is important to mention that using this formulation the increase might not be an integer and; thus, neither the number of knots. In order to provide a better adjustment, they will be real values and the rounded value at each iteration will be the number of knots and projections for that iteration. Due to this fact the parameter n_{inc} can effectively influence the result and increase the adaptability of the iterative approach.

A final remark is the convergence criteria, the conditions that must be met to stop the iterations. Three were used in this case: a maximum number of iterations was reached, which was set to 8, the total cross-correlation was bigger than 0.8 or when the absolute change on the cross-correlation is smaller than 0.01. The first avoids the optimisation to be run infinitely, the second allows to exit once a good result has been obtained and the last stops the process if the optimization has converged to a solution. Once the iterations finish a last PLTW is performed using n_{max} knots. This value is set to 4.20 but more knots could be used as after the iterations the initial alignment will be good. The whole process is described in algorithm 6.1.

Using this configuration, a warping experiment was performed on a heavily distorted sample data, shown in figure 6.1. The result is very good and all the features are aligned properly

giving a final cross-correlation of 0.89. The warping is similar to what it would be obtained if a weak distortion were present.

Algorithm 6.1: Knot iteration PLTW

Result: Warped result

estimate the offset a_{offset};
initialise n and f_{cutoff};
construct $\mathbf{t}_{knot}$;
$\mathbf{a}_0 = \mathbf{t}_{knot} + a_{offset}$;
low pass filter $\mathbf{s}$ and $\mathbf{r}$ using f_{cutoff};
undersample $\mathbf{t}$ to $\mathbf{t}_{und}$;
while *Convergence conditions are not satisfied* **do**
  $\mathbf{a} \leftarrow \arg\min 1 - \frac{\mathbf{r}_{filtered}(\mathbf{t}_{und})^H \cdot \mathbf{s}_{filtered}(w(\mathbf{t}_{und},\mathbf{a}))}{\sigma_r \sigma_s}$ with constraints;
  $n = n + n_{inc}$;
  update $\mathbf{t}_{knot}$ and $\mathbf{a}$;
end
$n = n_{max}$;
update $\mathbf{t}_{knot}$ and $\mathbf{a}$;
$\mathbf{a}^* \leftarrow \arg\min \left(1 - \frac{\mathbf{r}_{filtered}(\mathbf{t}_{und})^H \cdot \mathbf{s}_{filtered}(w(\mathbf{t}_{und},\mathbf{a}))}{\sigma_r \sigma_s}\right)$ with constraints;
$\mathbf{s}_W = \mathbf{s}\left(w\left(\mathbf{t}, \mathbf{a}^*\right)\right)$;

Moreover, the result that would be obtained using the default PLTW is shown as a comparison. Its warping is relatively good in some sections but others are wrongly aligned. However, it can be seen that despite being a complex misalignment the PLTW does not give a terrible answer, significantly improving the prealigned result.

The last remark is regarding the computational time. The main drawback of this iterative approach is that the computational time is significantly incremented. Fortunately, the first iterations are virtually costless because very few parameters are estimated. The total time will obviously depend on the number of iterations done before convergence is reached. In the example shown 4 iterations plus the final PLTW were done and it took 2.5 times longer than just applying one PLTW. This must be considered especially when dealing with long logs.

6.2 PIECEWISE LINEAR TIME WARPING WITH FREQUENCY ITERATION

Considering the success of the knot iteration, a similar approach was tried for the frequencies. Using a very low pass filter helps to avoid converging into local optimums, but it achieves it by heavily distorting the cost function. A solution would be to match the frequencies separately, so first only the lowest frequencies would be considered and steadily more would be introduced until the whole signal is warped. With this approach the complexity of the warping problem is reduced.

Thus, the idea is the same as with the knot iteration, the first iteration will be done having used a very low cutoff frequency and the resulting set of parameters, $\mathbf{a}$, will be used for the next iteration. Afterwards a higher cutoff frequency will be applied and the starting set of parameters will be set to the ones calculated in the previous iteration. Since the data is different, the result will also be different and after a certain number of iterations the approximation will converge to a final result. Thus, the signals will be aligned by parts, starting from the lowest frequencies and steadily adding higher frequencies until the whole signal is aligned.

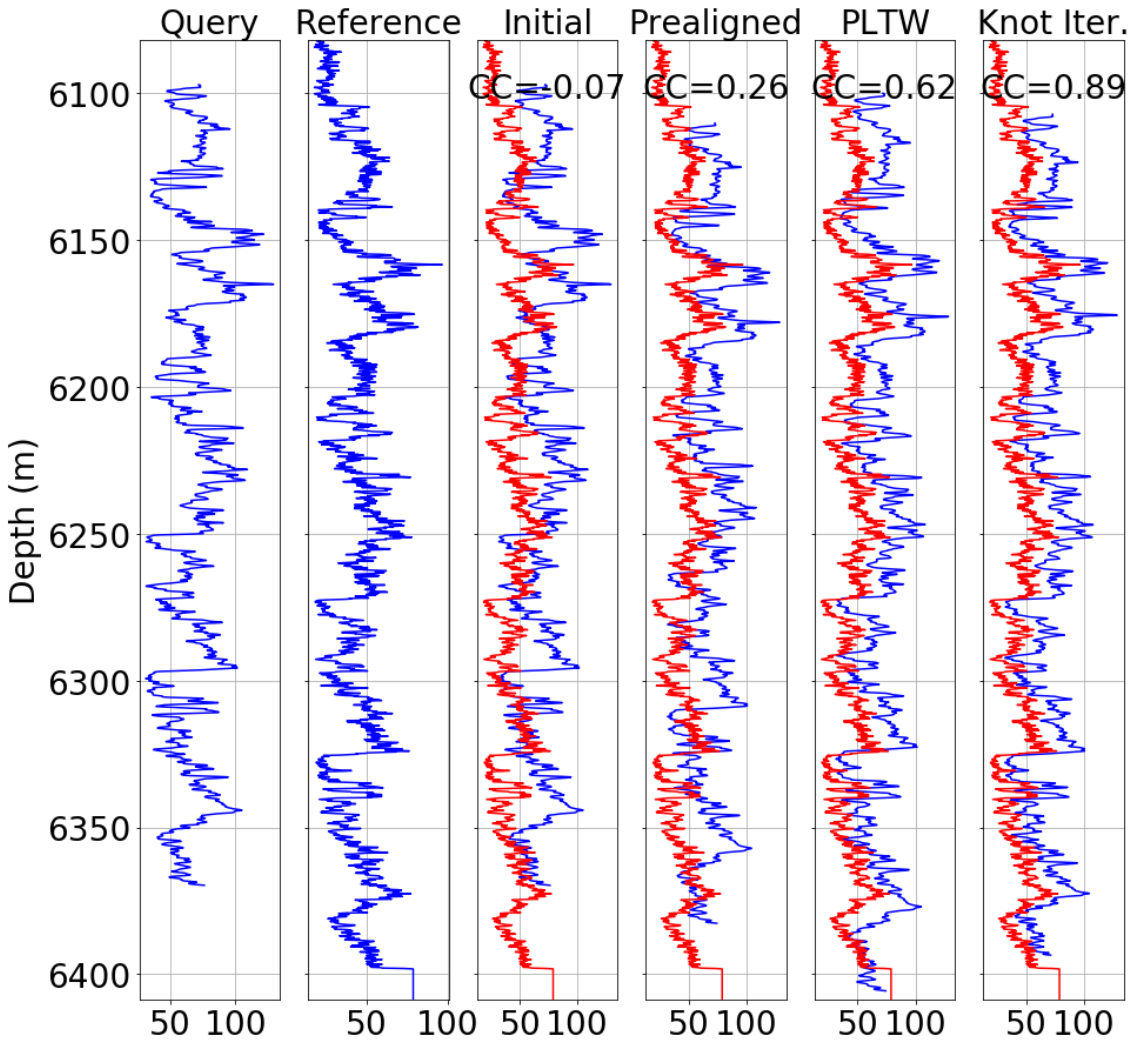


Figure 6.1: From left to right: firstly the query and secondly the reference. Thirdly the initial warping, fourthly the initial warping applying an offset correction, fifthly the result of a direct PLTW and finally the warping obtained with the knot iteration. In red the reference and in the top the total cross-correlation are shown.

The method is initialised with an initial cut-off frequency, $f_{cutoff,1}$. The original signal is filtered and the PLTW is run obtaining $\mathbf{a}_1$. The result is feed to the next iteration as the set of initial parameters. The cut-off frequency at each iteration is calculated from the previous as

$$f_{cutoff,i+1} = f_{cutoff,i} + f_{inc}, \tag{6.5}$$

where f_{inc} is a fixed value that defines the increment of each iteration. The process is repeated certain number of times and at the end a standard PLTW, with a cut-off frequency of $f_{cutoff,max}$, which was set to $0.5m^{-1}$, is run to obtain the final result.

The converging criteria will be the same as in the knot iteration, three conditions will be evaluated and as soon as one is fulfilled the iterations will be stopped: a maximum number of 8 iterations is reached, the total cross-correlation is over 0.8 or the absolute change on the

cross-correlation is smaller than 0.01. The same criteria for the knot iteration is set in order to have a fair comparison. This procedure is explained in algorithm 6.2.

Algorithm 6.2: Frequency iteration PLTW

Result: Warped result

estimate the offset a_{offset};
initialise n and f_{cutoff};
construct $\mathbf{t}_{knot}$;
$\mathbf{a}_0 = \mathbf{t}_{knot} + a_{offset}$;
low pass filter $\mathbf{s}$ and $\mathbf{r}$ using f_{cutoff};
undersample $\mathbf{t}$ to $\mathbf{t}_{und}$;
while *Convergence conditions are not satisfied* **do**
  $\mathbf{a} \leftarrow \arg\min \left(1 - \frac{\mathbf{r}_{filtered}(\mathbf{t}_{und})^H \cdot \mathbf{s}_{filtered}(w(\mathbf{t}_{und},\mathbf{a}))}{\sigma_r \sigma_s}\right)$ with constraints;
  $f_{cutoff} = f_{cutoff} + f_{inc}$;
  low pass filter $\mathbf{s}$ and $\mathbf{r}$ using f_{cutoff};
end
$f_{cutoff} = f_{cutoff,max}$;
low pass filter $\mathbf{s}$ and $\mathbf{r}$ using f_{cutoff};
$\mathbf{a}^* \leftarrow \arg\min \left(1 - \frac{\mathbf{r}_{filtered}(\mathbf{t}_{und})^H \cdot \mathbf{s}_{filtered}(w(\mathbf{t}_{und},\mathbf{a}))}{\sigma_r \sigma_s}\right)$ with constraints;
$\mathbf{s}_W = \mathbf{s}\left(w\left(\mathbf{t}, \mathbf{a}^*\right)\right)$;

In figure 6.2 the result of this approach in the same sample used for testing the knot iteration is shown. It gives a similar outcome as the knot iteration and it retrieves the warping properly.

The main advantage comes from aligning low frequencies. The projection, $\mathbf{a}$, is adapted from the beginning, so at first local optimums are avoided. In the following iterations, the knots are adjusted around the previous value, so they will converge to the real optimum. This is specially useful with heavily distorted signals where the regular PLTW would create a mismatch. However, in general, it is not guaranteed that the final convergence will be proper. It can occur that due to an inadequate value of the increase, the problem will converge into a local optimum because working in very low frequencies makes the warping very flexible. Therefore, this approach must only be used with samples with strong distortion.

Another drawback is the computational time. All the iterations are done using the same number of knots; therefore, each iteration is very costly, mainly when long runs are being processed. In general, no more than 8 iterations are necessary and after a few, a significant improvement is seen. But this will depend on the samples themselves, for instance, in the example given a good result was obtained using just 4 iterations. Thus, the total time will be the number of iterations times the computational time of one PLTW.

6.3 ITERATIVE PIECEWISE LINEAR TIME WARPING

Finally, the two previous approaches were combined to create the Iterative PLTW, combining the advantages from both. An initial knot number, n_1, and cut-off frequency, $f_{cutoff,1}$, are chosen. In each iteration the raw data is taken through a low pass filter with $f_{cutoff,i}$ and the initial values are taken from the previous iteration, $\mathbf{a_{i-1}}$. Since the number of knots is different from one iteration to the next, the set of parameter is interpolated as shown in section 6.1 so that there are n_i knots. In each iteration the values are updated with

$$n_{i+1} = n_i + n_{inc} \quad and \quad f_{cutoff,i+1} = f_{cutoff,i} + f_{inc}, \tag{6.6}$$

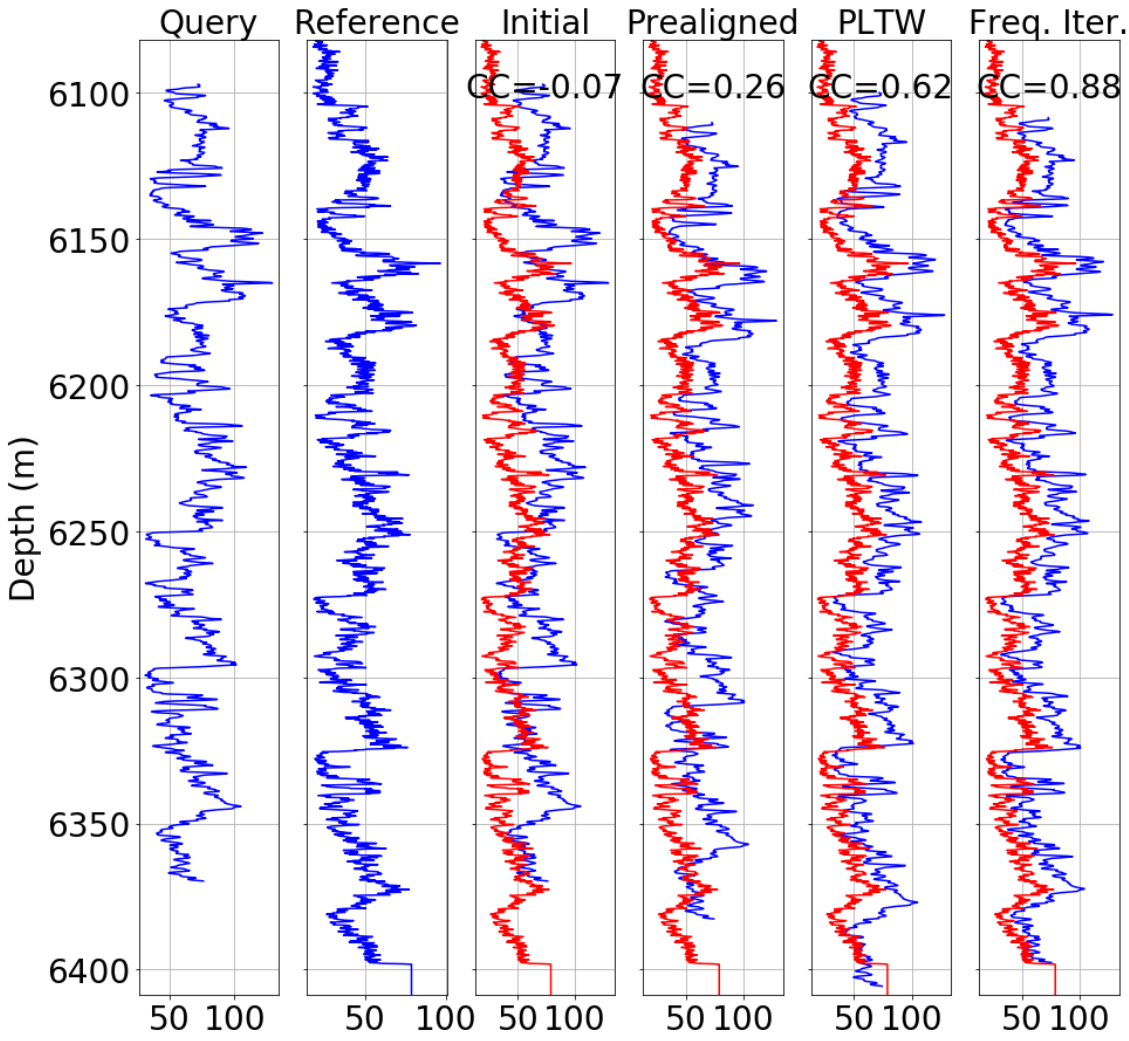


Figure 6.2: From left to right: firstly the query and the reference. Thirdly the initial warping, fourthly the initial warping applying an offset correction, fifthly the result of a direct PLTW and finally the warping obtained with the frequency iteration. In red the reference and in the top the total cross-correlation are shown.

and the result of the PLTW, $\mathbf{a_i}^*$, will be used as the initial parameters for the next iteration. The stopping criteria is maintained as for the previous iterative methods, so it will stop when either a maximum number of iterations have been reached, a good total cross-correlation has been obtained or the improvement from the previous iteration has been very small.

The key for the success of the algorithm is how the variables are updated and the choice of the initial parameters. Regarding the knots, it seems sensible to start with two knots in order to give an overall stretching to the curve. And the knot increase will be of one extra knot per 300 meters, as in equation 6.4 as it gave a good result in the tested cases.

The frequency related parameters are trickier to define, as they will affect the result in a not easily predictable manner. The initial value must be low in order to avoid the local optimums and to be able to move between peaks, so a similar value as used in the offset calculation should be applied. Thus, a value of 0.01 m^{-1} was selected. The update must also be low because after 0.5 m^{-1} low pass filtering provides no considerable effect, so, after several attempts, a constant increase of 0.012 m^{-1} was used with a maximum number of 6 iterations.

First the offset is estimated as explained in chapter 5 and subsequently the iterations start. Furthermore, the undersampling described in section 5.3 is also applied to speed up the algo-

rithm. Since the sampling period of these measurements was $0.15m$, an undersampling of 5 was used. Once the iterations converge a final regular PLTW is done using the number of knots given by equation 4.20 to obtain the final result as explained by algorithm 6.3.

Algorithm 6.3: Iterative PLTW

Result: Warped result

estimate the offset a_{offset};
initialise n and f_{cutoff};
construct $\mathbf{t}_{knot}$;
$\mathbf{a}_0 = \mathbf{t}_{knot} + a_{offset}$;
low pass filter $\mathbf{s}$ and $\mathbf{r}$ using f_{cutoff};
undersample $\mathbf{t}$ to $\mathbf{t}_{und}$;
while *Convergence conditions are not satisfied* **do**
  $\mathbf{a} \leftarrow \arg\min \left(1 - \frac{\mathbf{r}_{filtered}(\mathbf{t}_{und})^H \cdot \mathbf{s}_{filtered}(w(\mathbf{t}_{und},\mathbf{a}))}{\sigma_r \sigma_s}\right)$ with constraints;
  $n = n + n_{inc}$;
  update $\mathbf{t}_{knot}$ and $\mathbf{a}$;
  $f_{cutoff} = f_{cutoff} + f_{inc}$;
  low pass filter $\mathbf{s}$ and $\mathbf{r}$ using f_{cutoff};
end
$n = n_{max}$;
update $\mathbf{t}_{knot}$ and $\mathbf{a}$;
$f_{cutoff} = f_{cutoff,max}$;
low pass filter $\mathbf{s}$ and $\mathbf{r}$ using f_{cutoff};
$\mathbf{a}^* \leftarrow \arg\min \left(1 - \frac{\mathbf{r}_{filtered}(\mathbf{t}_{und})^H \cdot \mathbf{s}_{filtered}(w(\mathbf{t}_{und},\mathbf{a}))}{\sigma_r \sigma_s}\right)$ with constraints;
$\mathbf{s}_W = \mathbf{s}\left(w\left(\mathbf{t}, \mathbf{a}^*\right)\right)$;

In figure 6.3 the result obtained using this configuration is shown. This method also retrieves the warping properly and the result is slightly better than the obtained with the previous methods based on the resulting total cross-correlation. Therefore, smartly combining the knot and the frequency iteration gives the best possible result. Due to this fact this method is defined as the standard iterative method.

Regarding its properties, the frequency iteration helps avoiding local optimums and combined with the knot iteration the result is guided towards the best warping possible. Therefore, a very flexible warping is obtained which is especially useful to treat signals that have suffered a severe distortion.

The main drawback of this approach is the increase of the number of input parameters. In the PLTW the only important parameter, assuming that an offset correction is performed, is the number of knots, while the iterative version needs more input variables. For instance, in addition to the initial number of knots, the knot increase, the initial cut off frequency and its increment, all of which affect the final result. This fact, again, can be either positive or negative depending on the context. On the one hand, it provides flexibility, so it can be adapted to different scenarios, but this means that some tuning of the parameters must be done beforehand. In depth matching the knot increment can be defined based on the length of the signals. Regarding the cut off frequencies, this information can be again taken from the frequency analysis done previously in section 5.2. Furthermore, after trying out different values, in spite of being an optimum configuration for each alignment, these parameters seem not to affect dramatically the result, so there is a wide range of configurations that will give the correct result.

The computational time will obviously be higher than the one from the regular PLTW, but it will not be as high as the frequency iterative version. The performance will be similar to the knot iterative PLTW because the most influencing property that determines the computing time

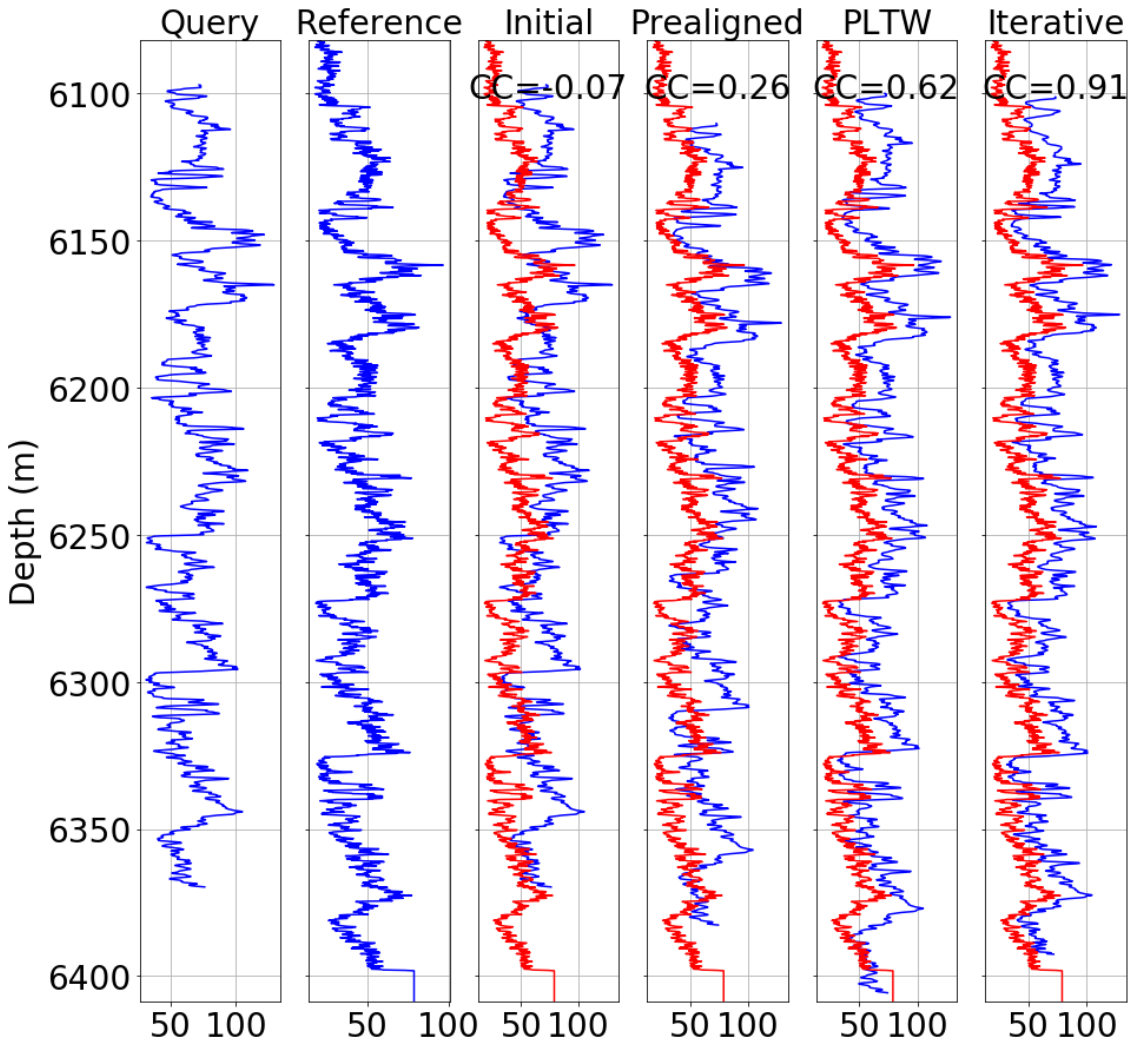


Figure 6.3: From left to right: firstly the query and the reference. Thirdly the initial warping, forthly the initial warping applying an offset correction, fifthly the result of a direct PLTW and finally the warping obtained with the iterative PLTW. In red the reference and in the top the total cross-correlation are shown.

is the number of knots the PLTW employs. Thus, it can be estimated that the total computing time will be about 2.5 times bigger than the normal PLTW would need.

The last consideration is that the update consists on a fix increment for both the knot number and the cut off frequency. This is clearly not ideal as the optimum would be to have an adaptive increment as it is the case for gradient methods, for instance, adapting it based on the total normalised cross-correlation and its improvement from the previous iteration. Nonetheless, this is a complex subject and it is left as a possible further research direction.

6.4 EXPERIMENTAL RESULTS

This chapter has addressed an iterative PLTW approach to depth matching borehole logs in order to increase the application range of the PLTW algorithm so that it can solve problems with heavy distortions. In the previous sections an example with a long stretched section that could not be solved using the regular PLTW was used as an example. The result was good in all the

iterative versions, and thus the iterative approach seems to be a good solution or adaptation to enable solving complex alignments.

In figure 6.4 a more complex example is treated. The query is an artificially manipulated version of the reference, so there exists a warping function that achieves a perfect match. Nonetheless, this artificial warping is very challenging as it has multiple first order shifts, some of which are very severe.

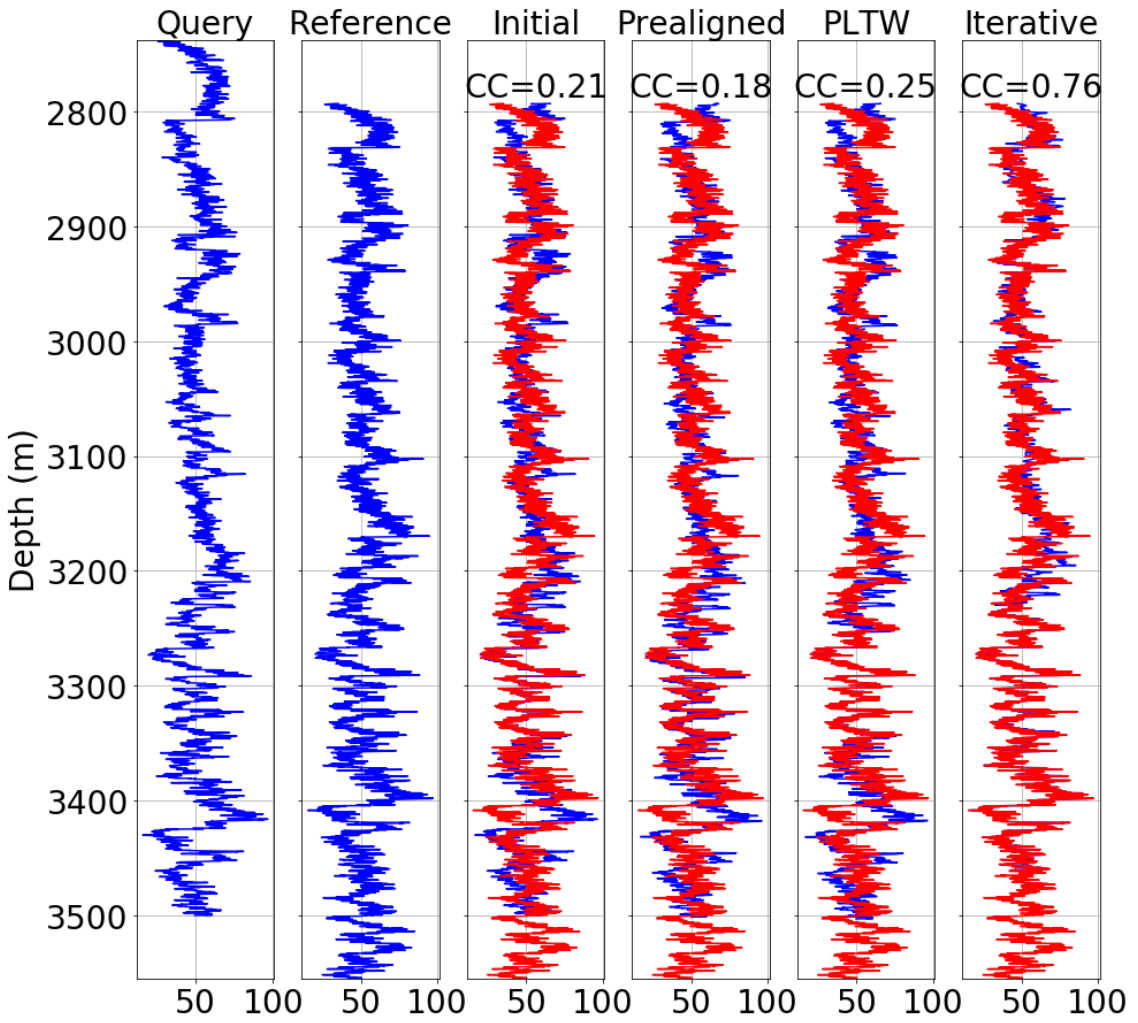


Figure 6.4: From left to right: firstly the query and the reference. Thirdly the initial warping, fourthly the initial warping applying an offset correction, fifthly the result of a direct PLTW an finally the warping obtained with the iterative PLTW. In red the reference and in the top the total cross-correlation are shown.

The correct warping is difficult to spot by eye, so it is indeed a difficult example. However, the algorithm gives a considerably good result. The lower section of the log has a perfect match while other sections are not so well aligned. The main cause of misalignments are again those sections where the algorithm has converged to a local optimum, so two different peaks have been aligned. This is the main problem of the PLTW and it appears again in this case. Thus, even though more knots were used, the result would not change significantly and the only solution would be to further exploit the low frequencies. Nonetheless, a normalised cross-correlation of 0.76 is obtained, which is a big improvement from the initial and can be considered a promising result.

A similar case is given in figure 6.5 where another hard case was investigated. A warping result is obtained with a normalised cross-correlation of 0.83. By visual inspection it can be

concluded that the match or alignment is considered good and only some regions are slightly misaligned.

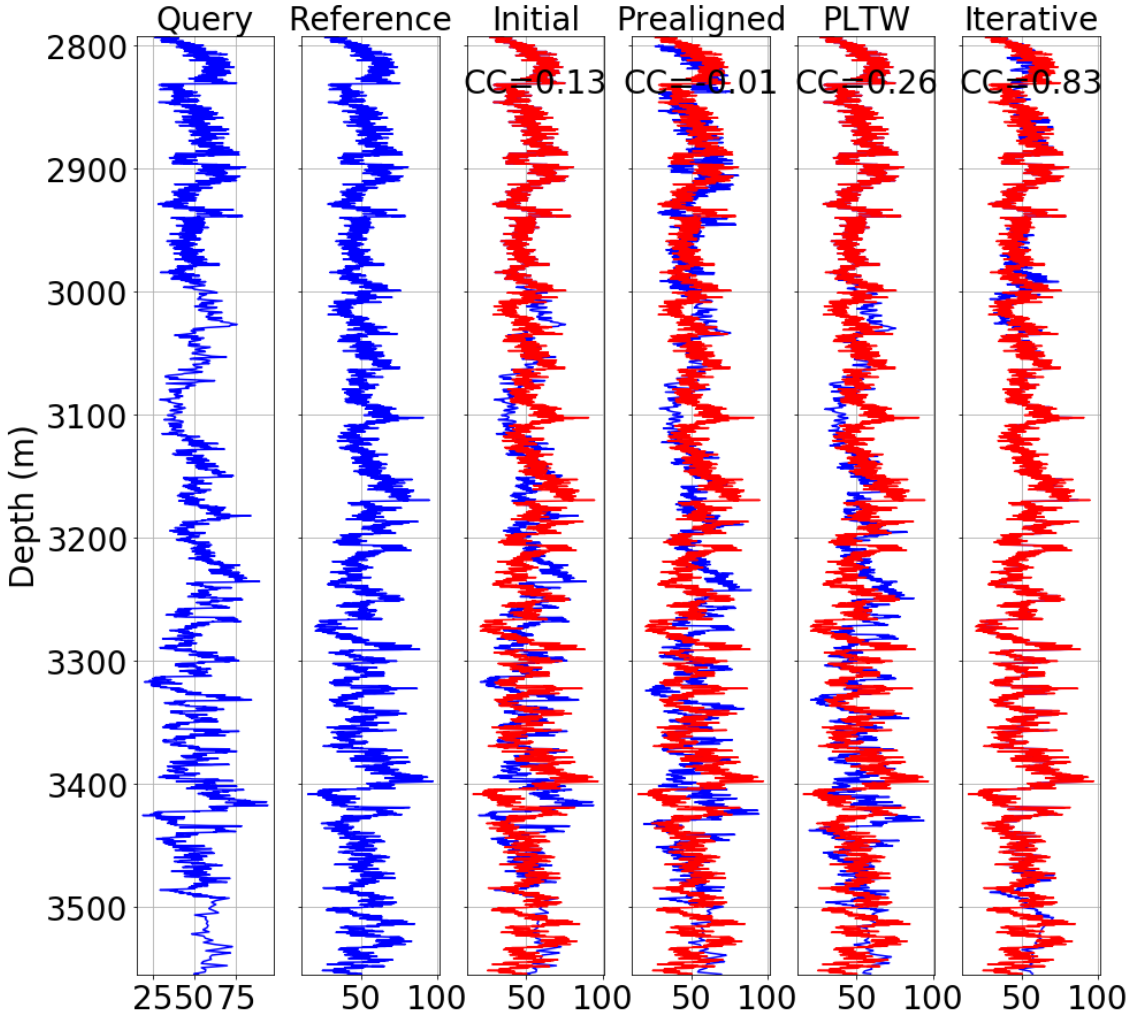


Figure 6.5: From left to right: firstly the query and the reference. Thirdly the initial warping, fourthly the initial warping applying an offset correction, fifthly the result of a direct PLTW and finally the warping obtained with the iterative PLTW. In red the reference and in the top the total cross-correlation are shown.

The iterations combined with the use of a variety of low pass filters enable to have a very flexible warping. Different input parameters have been tried and the default values given in the previous section give the best performance in general. Moreover, small variations of these values do not affect the result significantly, which is positive as no prior analysis needs to be done.

The results obtained with the iterative PLTW approach are good and certainly better than those the regular PLTW would provide. The sections that are not warped correctly do not show artifacts nor peak destruction such as the DTW would create. Even with suboptimal automatic warping, the result will be easier to align manually because fewer sections will be misaligned and the shapes will not have been dramatically changed.

Due to the multiple iterations the main drawback is the extra computational time. This will vary from log to log as some will be easier to align than others and the result can be reached in fewer iterations. With the current examples the algorithm took around 3 times longer than the PLTW, as it was expected. Thus, it is sensible to estimate that using the iterative PLTW it will take 3 times longer to warp in the worst case scenario. This can be a matter to consider when dealing with long logs. A solution for cases where the time is a constraint would be to increase

the undersampling in order to have the result faster at the expense of slightly decreasing the accuracy of the result.

In conclusion, applying the PLTW iteratively provides a tool to tackle complex alignments at the expense of a higher computational complexity. In consequence, samples that show a good alignment after the offset correction can be matched using the regular PLTW while the more difficult cases can be treated with the iterative PLTW.

6.5 FINAL TOOL

Having performed several tests the final warping workflow to tackle depth matching problem was created combining the different methods designed during the project. The workflow is given in the flow chart in figure 6.6. First, the global offset is calculated as explained in chapter 5 and the total cross-correlation of the prealigned signal is computed. If this value is bigger than certain threshold, it is assumed that the offset correction has solved most of the misalignment and the PLTW can be directly applied using the settings explained in chapter 5. On the other hand, if this value is smaller than the threshold it means that there are more misalignment causes than a constant offset, so the iterative PLTW described in this chapter will be run.

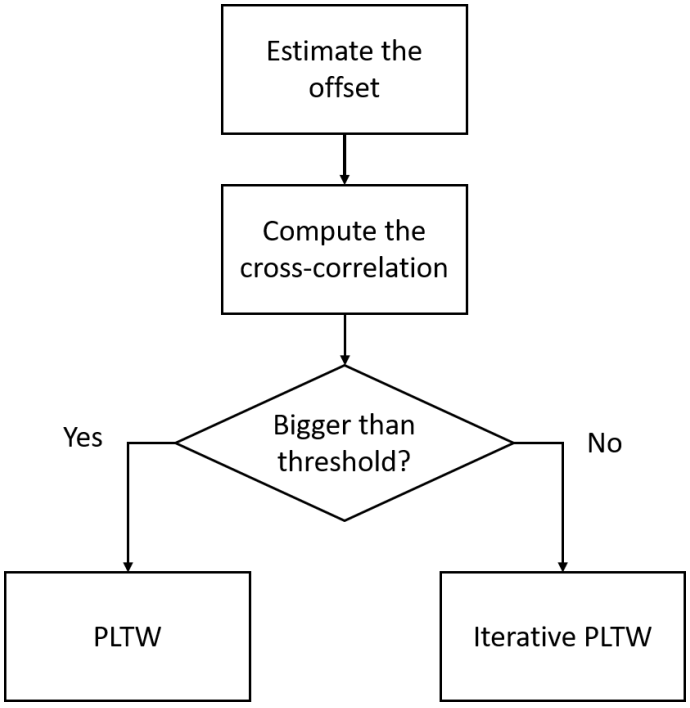


Figure 6.6: Flow chart of the final tool.

The mentioned threshold was set to 0.6 after experimenting with the provided dataset. In general, the vast majority of the data sets will surpass the threshold and the PLTW will be run, which has already shown great results as concluded in chapter 5. Therefore, only the exceptional cases which have suffered severe misalignment will be corrected using the Iterative PLTW.

7 CONCLUSION

The main purpose of this project was to investigate and create an automated algorithm for matching two misaligned borehole logs, which is currently still being done manually in the so-called depth matching process. In this work using the theory of Parametric Time Warping (PTW) a tool, written in Python, has been created for addressing the depth matching problem using the theory of the PTW. The application of the methodology to real data has shown a great success.

In order to model the real distortions that occur during acquiring log measurements, the warping function is assumed to be a piecewise linear function described by a fixed number of knots and their projections into the depth of the query. This method was called Piecewise Linear Time Warping (PLTW), where a cost function is defined based on the cross-correlation and the optimum projections are computed by solving a minimization problem. This mimics the manual depth matching currently done where a petrophysicist handpicks several points and the different sections are either stretched or squeezed.

The initial idea was enhanced in chapter 5 where an offset estimation was introduced using PTW and the frequency domain was exploited. The final result is a robust tool that is able to warp properly the most common cases of misalignment in borehole logs. Due to being based on the PTW theory the result is not as flexible as a DTW would be, so the features will be preserved and no information would be lost as in [4]. Moreover, the results would be obtained in the order of seconds.

In chapter 6, the iterative version was introduced, where multiple PLTW were done in iterations, feeding the result from the previous to the next iteration. This last technique gave the best results and was able to solve severe distortion cases, expanding the field of applications. Nonetheless, this will come at the expense of a higher computational time. It is encouraged to use the regular PLTW in general and only make use of the iterative approach if the PLTW does not give good enough results. Thus, the final method proposed combines these two algorithms and uses either one or the other depending on how good the initial prealignment is with the estimated constant global offset. The algorithms have been thoroughly tested on the provided dataset and the results are comparable to those an experienced petrophysicist would obtain.

7.1 DISCUSSION

Many methods were discussed in chapter 3 and it was hard to make a choice on which to focus for this application. The suggested method by latest studies [4] and currently the most widely used was the Dynamic Time Warping (DTW), but its main drawback was that it would delete or destroy minor peaks and valleys which are relevant data for understanding subsurface. In spite of multiple variations of the DTW [26; 29] it is currently still a problem to use this technique in applications were the shape and features of the curves must be preserved, so it was decided to look for other methods which had a friendlier warping. A similar problem to the automatic depth matching was encountered in chromatogram alignment [28] and based on the previous research it was decided to try to implement the PTW (Parametric Time Warping) and the STW (Semiparametric Time Warping). These methods were known for preserving the features and not being very flexible, and the latter had not been widely implemented yet. Since the warping in depth matching is almost perfectly described by a piecewise linear function it was decided to exploit this idea and focus the research efforts on it.

There are some papers that suggest splitting the whole signal in sections and applying the PTW to each of them individually [32]. Nonetheless, since the optimisation is done locally this could prompt errors, for instance assigning one peak to the incorrect section, and a suboptimal solution could be obtained. This is why a STW was preferred and the PLTW was investigated, because the problem is handled globally and the segmentation will not crucially influence the outcome.

7.2 FUTURE DIRECTIONS

To further improve developed methodology, the research could focus on two main directions: one regarding the PLTW method itself and another regarding the automatic depth matching topic in general.

7.2.1 Parametric Warping

Firstly the actual method developed must be exhaustively tested in order to discover its robustness and flaws. From those learning possible improvements could be derived. The best way would be to add the algorithm to the software currently being used by petrophysicist for performing the depth matching, so that the feedback from the users can be used for evaluating the results and the algorithm. These tests would be crucial for instance to find the optimum input values for the knot number and the increments.

It became clear that the offset correction is vital, and it is the main enhancer of the method. Moreover, regardless of the method used (PLTW, DTW, COW, etc.), before running it an offset correction should be performed. In this case a basic PTW has been used for this purpose but it is encouraged to investigate other methods to estimate the constant global offset.

It is highly encouraged to improve the prealignment, which currently just consists in an offset correction. This could be done by different means such as a first DTW or by applying first the PTW per windows [32] to have a better estimate of the initial warping function.

Lastly, although the computational time is reasonable, it is worth investigating how to seed up the computations. One solution would be to test different solvers for the optimization problem and see if any major improvement can be achieved. It could also be possible to try to solve the problem using dynamic programming as the Correlated Optimised Warping (COW) tries [26]. A last option could be to use cloud computing in order to avoid running the program in the local machine and provide the results to the software petrophysicist used to perform the depth matching.

7.2.2 Automatic Depth Matching

In spite of the success of this method for depth matching, it would be beneficial to investigate other methods for achieving the automatic depth matching. One of the most used methods for curve alignment is still the DTW and it would be interesting to understand if the DTW and the PLTW can be combined in order to obtain the fast computation from the DTW and the shape preserving from the PLTW.

A very good candidate would be the COW algorithm which has been previously used for chromatogram alignment [31]. This method tries to constrain the flexibility of the DTW in order to preserve its original shape and avoid overfitting. Other algorithms that aim to decrease the flexibility of the DTW are also promissing candidates such as the DDTW [30] or the SSDTW [29]. Variations of the DTW with reduced flexibility can be tried in order to have a faster computation.

A different approach could be to investigate the problem in the wavelet domain to extract certain features and ease the warping problem. For instance, they could be used to obtain the different stratigraphic regions as in [56] and later directly aligning the regions with each other.

Moreover, the wavelets could be combined with the machine learning study [6] in order to have an automated procedure based on the features given from the wavelet domain.

A PIECEWISE LINEAR FUNCTION COMPUTATION

This appendix contains the method used to compute the piecewise linear function efficiently based on [39]. The problem is to obtain the values of the piecewise linear function at the depth instances **t** given the knot location, $\mathbf{t}_{knot}$, and their projections, **a**.

There are different ways to compute the value of the function at certain depth instances, **t**. Firstly, it is assumed that **t** is monotonous, so $t_i > t_j \quad \forall \quad i > j$. Secondly, the projections, **a** can be considered to create a set of basis, **H**, where each has a value of

$$H_i(t) = \begin{cases} \frac{t - t_{knot,i-1}}{t_{knot,i} - t_{knot,i-1}} & \text{if } t_{knot,i-1} < t < t_{knot,i}, \\ \frac{t_{knot,i+1} - t}{t_{knot,i+1} - t_{knot,i}} & \text{if } t_{knot,i} < t < t_{knot,i+1}, \\ 0 & \text{otherwise,} \end{cases} \tag{A.1}$$

at the time instance t. In figure A.1 this is illustrated and some elements are shown together. Thus, each projection, a, generates an impulse response at its knot, but at assuming a wide spike instead of a pure impulse the **H** set is created. It is important to note that in this case the values smaller than $t_{knot,1}$ and bigger than $t_{knot,n}$ will have a value of 0.

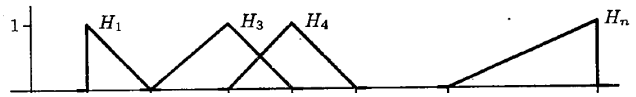


Figure A.1: Some elements of the basis functions taken from [39]

.

Using this definition the input vector **t** is segmented depending on their value with respect to the knots; therefore dividing the vector in $n + 1$ segments, where n is the number of knots. The value for a depth instance $t \in [t_{knot,i}, t_{knot,i+1}]$ can be computed using just the two closest knots giving the value of

$$f_{BL}(t) = H_i(t) \cdot a_i + H_{i+1}(t) \cdot a_{i+1} \tag{A.2}$$

Since the data is segmented only and a monotonous **t** is assumed the computation is the fastest as possible because only the needed operations are done.

BIBLIOGRAPHY

[1] J. Zangwill, "Depth matching a computerized approach," 07 1982.

[2] M. Kerzner, "A solution to the problem of automatic depth matching," 01 1984.

[3] Z. Steven, P. Ramoj, and D. Steve, "Curve alignment for well-to-well log correlation," *Proceedings - SPE Annual Technical Conference and Exhibition*, 09 2004.

[4] R. Herrera and M. Baan, "Automated seismic-to-well ties using dynamic time warping," 09 2012.

[5] M. Kerzner, "Automatic relative depth matching of borehole information," *Geophysical Transactions*, vol. 32, pp. 333–353, 1987.

[6] T. Zimmermann, "Machine-learning-based automatic well-log depth matching," *Petrophysics*, vol. 59, pp. 863–872, 12 2018.

[7] T. Bloemberg, J. Gerretzen, A. Lunshof, R. Wehrens, and L. Buydens, "Warping methods for spectroscopic and chromatographic signal alignment: A tutorial," *Analytica chimica acta*, vol. 781C, pp. 14–32, 06 2013.

[8] S. Chan and Y. H. Tay, "Online signature verification using dynamic time warping," 01 2006.

[9] M. Müller, *Information Retrieval for Music and Motion*. 01 2007.

[10] P. Eilers, "Parametric time warping," *Analytical chemistry*, vol. 76, pp. 404–11, 02 2004.

[11] J. Mari and F. Coppens, "Seismic well surveying," 01 1991.

[12] Z. Bassiouni, *Geophysics and Geosequestration*, pp. 181–194. 04 2019.

[13] T. Darling, "Well logging and formation evaluation," 01 2005.

[14] R. Maliva, *Borehole Drilling and Well Construction*, pp. 127–170. 05 2016.

[15] J. Fanchi and R. Christiansen, *Introduction to Petroleum Engineering*, pp. 161–184. 09 2016.

[16] H. Liu, *Principles and Applications of Well Logging*, pp. 211–236. 11 2017.

[17] P. Hagan and G. West, "Interpolation methods for curve construction," *Applied Mathematical Finance*, vol. 13, pp. 89–129, 02 2006.

[18] J. Benesty, J. Chen, Y. Huang, and I. Cohen, *Pearson Correlation Coefficient*, vol. 2, pp. 1–4. 04 2009.

[19] M. G. Kerzner, "Formation dip determination an artificial intelligence approach," *The Log Analyst*, vol. 5, pp. 10–22, 9 1983.

[20] J. Fang, H. Chen, A. Shultz, and W. Mahmoud, "Computer-aided well log correlation," vol. 76, pp. 307–317, 01 1992.

[21] P. Senin, "Dynamic time warping algorithm review," 01 2009.

[22] L. Muda, M. Begam, and I. Elamvazuthi, "Voice recognition algorithms using mel frequency cepstral coefficient (mfcc) and dynamic time warping (dtw) techniques," *J Comput*, vol. 2, 03 2010.

[23] T. Rakthanmanon, B. Campana, A. Mueen, G. Batista, M. B. Westover, Q. Zhu, J. Zakaria, and E. Keogh, "Searching and mining trillions of time series subsequences under dynamic time warping," vol. 2012, 08 2012.

[24] D. Berndt and J. Clifford, "Using dynamic time warping to find patterns in time series," vol. 10/16, pp. 359–370, 01 1994.

[25] P.-F. Marteau, "Time warp edit distance with stiffness adjustment for time series matching," *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, pp. 306–18, 03 2009.

[26] G. Tomasi, F. Berg, and C. Andersson, "Correlation optimized warping and dynamic time warping as preprocessing methods for chromatographic data," *Journal of Chemometrics*, vol. 18, pp. 231 – 241, 05 2004.

[27] S. Salvador and P. Chan, "Toward accurate dynamic time warping in linear time and space," *Intell. Data Anal.*, vol. 11, pp. 561–580, 10 2007.

[28] T. Bloemberg, J. Gerretzen, A. Lunshof, R. Wehrens, and L. Buydens, "Warping methods for spectroscopic and chromatographic signal alignment: A tutorial," *Analytica chimica acta*, vol. 781C, pp. 14–32, 06 2013.

[29] J. Hong, S. Park, and J.-G. Baek, "Ssdtw: Shape segment dynamic time warping," *Expert Systems with Applications*, vol. 150, p. 113291, 02 2020.

[30] E. Keogh and M. Pazzani, "Derivative dynamic time warping," *First SIAM International Conference on Data Mining*, vol. 1, 01 2002.

[31] N.-P. Nielsen, J. Carstensen, and J. Smedsgaard, "Aligning of single and multiple wavelength chromatographic profiles for chemometric data analysis using correlation optimised warping," *Journal Of Chromatography A*, vol. 805, pp. 17–35, 05 1998.

[32] T. Bloemberg, J. Gerretzen, H. Wouters, J. Gloerich, M. Dael, H. Wessels, L. Heuvel, P. Eilers, L. Buydens, and R. Wehrens, "Improved parametric time warping for proteomics," *Chemometrics and Intelligent Laboratory Systems*, vol. 104, pp. 65–74, 11 2010.

[33] R. Wehrens, T. Bloemberg, and P. Eilers, "Fast parametric time warping of peak lists," *Bioinformatics (Oxford, England)*, vol. 31, 05 2015.

[34] B. Taylor, "Methodus incrementorum directa inversa.," 1715.

[35] C. Runge, "Über empirische funktionen und die interpolation zwischen äquidistanten ordinaten," *Z. Math. Phys.*, vol. 46, 01 1901.

[36] S. De Marchi, F. Marchetti, E. Perracchione, and D. Poggiali, "Polynomial interpolation via mapped bases without resampling," *Journal of Computational and Applied Mathematics*, vol. 364, p. 112347, 07 2019.

[37] H. Sorenson, "Least-squares estimation: from gauss to kalman," *Spectrum, IEEE*, vol. 7, pp. 63 – 68, 08 1970.

[38] J. C. Yoo and T. Han, "Fast normalized cross-correlation," *Circuits, Systems and Signal Processing*, vol. 28, pp. 819–843, 12 2009.

[39] C. de Boor, *A Practical Guide to Splines*, vol. 27. 01 1978.

[40] F. Girosi, M. Jones, and T. Poggio, "Regularization theory and neural networks architectures," *Neural Comput*, vol. 7, 10 1998.

[41] A. Nederkassel, M. Daszykowski, P. Eilers, and Y. Heyden, "A comparison of three algorithms for chromatograms alignment," *Journal of chromatography. A*, vol. 1118, pp. 199–210, 07 2006.

[42] S. Ruder, "An overview of gradient descent optimization algorithms," 09 2016.

[43] R. Kelley, *Iterative Methods for Optimization*, vol. 19. 01 1999.

[44] D. Liu and J. Nocedal, "On the limited memory bfgs method for large scale optimization," *Mathematical Programming*, vol. 45, pp. 503–528, 08 1989.

[45] S. Nash, "A survey of truncated-newton methods," *Journal of Computational and Applied Mathematics*, vol. 124, pp. 45–59, 12 2000.

[46] D. Kraft, "A software package for sequential quadratic programming," 01 1988.

[47] X. Zhao, C. Zhang, B. Yang, and P. Li, "Adaptive knot placement using a gmm-based continuous optimization algorithm in b-spline curve approximation," *Computer-Aided Design*, vol. 43, pp. 598–604, 06 2011.

[48] J. Luo, H. Kang, and Z. Yang, "Knot calculation for spline fitting based on the unimodality property," *Computer Aided Geometric Design*, vol. 73, 07 2019.

[49] T. Tjahjowidodo, T. Dung Van, and M. Han, "A fast non-uniform knots placement method for b-spline fitting," pp. 1490–1495, 07 2015.

[50] G. Beliakov, "Least squares splines with free knots: Global optimization approach," *Applied Mathematics and Computation*, vol. 149, 04 2004.

[51] S. Williams, "Offsets in global positioning system time series," *Journal of Geophysical Research B: Solid Earth*, vol. 108, pp. ETG 12–1, 06 2003.

[52] S. Luque and R. Fried, "Recursive filtering for zero offset correction of diving depth time series with gnu r package divemove," *PloS one*, vol. 6, p. e15850, 01 2011.

[53] E. Keogh and P. Smyth, "A probabilistic approach to fast pattern matching in time series databases," *Proceedings of Third International Conference on Knowledge Discovery and Data Mining, Newport Beach, CA, USA, 1997*, 09 1997.

[54] L. Rabiner and R. Schafer, "Theory and application of digital speech processingl," 07 2020.

[55] P. Burt, "Moment images, polynomial fit filters. and the problem of surface interpolation," pp. 144 – 152, 07 1988.

[56] X. Wu, "Automated stratigraphic interpretation of well-log data," *Geophysics*, vol. 52, 12 1987.