

BOREHOLE IMAGE CORRESPONDENCE AND AUTOMATED ALIGNMENT

Andriy Gelman[†], Arnaud Jarrot[†], [‡]Alexis He, [‡]Josselin Kherroubi, *Robert Laronga

[†]Schlumberger-Doll Research Center, 1 Hampshire Street, Cambridge, MA 02139, USA

[‡]Schlumberger-Riboud Product Center, 1 rue Henri Becquerel, 92140 Clamart, France

*Schlumberger Oilfield Services, 1325 South Dairy Ashford Road, Houston, TX 77077, USA

ABSTRACT

Borehole images are often misaligned due to a depth offset in the acquisition sensors of a wireline tool. We propose an algorithm that identifies matching feature points and aligns the image. One of the main challenges is that imaging pads have no azimuthal overlap. This has been solved by extrapolating the pixels on the boundary to create a synthetic overlap that facilitates feature matching. Feature matching is implemented in two stages: first by finding a local correspondence among feature points and second by performing a global minimization with additional regularization constraints on the estimated shifts. The method has been successfully used to align several imaging logs in less than 1 minute whereas a manual alignment would take 6 hours.

Index Terms— Borehole images, image alignment, stereo matching.

1. INTRODUCTION

Borehole images are used in the oil and gas industry to understand and quantify the geology of hydrocarbon reservoirs, and to optimize strategies for hydrocarbon extraction. The images can be obtained while a well is being drilled or after the drilling is completed. In the latter case, a separate logging tool is lowered into a well on a wireline cable. The formation is sampled and the measurements are transmitted to the surface. An example of a wireline tool and the measurements are shown in Fig. 1. The images are made from data from several pads that sample the formation at different depths along the tool.

To achieve an accurate reconstruction of wireline images requires that the tool depth is precisely known. However, due to the stretching of the cable and uncertainty in the tool velocity, these measurements are distorted. An established technique to improve the accuracy of tool depth is to use a Kalman filter to perform a double integration of the acceleration measured downhole. Due to the limits of measurement precision and accuracy, significant errors remain. The errors result in misaligned features on the displayed image, which are manually corrected by shifting the images to create a consistent set of features. This is a time-consuming process; an accurate

alignment on a 1,000-ft (100,000 scan lines) depth interval can take up to 6 hours.

The objective of our work is to develop an automated algorithm that aligns borehole images. In terms of existing work, this problem is related to stereo matching [1] in which the aim is to estimate a disparity map between two baseline images. The main difference between stereo matching and our problem is that borehole images have no physical overlap between the individual pads, and the techniques discussed in the literature cannot be applied. The issue of non-overlapping images for mosaicing was discussed in [2]. The authors proposed to extrapolate the missing pixels and perform the alignment on the extended images. However, this method assumes a fixed homography mapping, whereas in borehole images, the mapping is a function of the measured depth.

In the proposed method, we also use image inpainting to synthesize an overlap between neighboring pads. Feature matching is then performed in two stages, first by applying a local matching on the overlap region, followed by a global minimization with additional smoothness constraints on the estimated shifts.

This paper is structured as follows. We formulate the problem in Section 2. The main stages of the algorithm are described in Section 3. Example results are shown in Section 4, and the paper is concluded in Section 5.

2. PROBLEM FORMULATION

As illustrated in Fig. 1(a), wireline tools are made of a series of pads located at different depths along the tool. The pads contain sensor elements that measure a formation property, typically resistivity or conductivity. The k -th pad on the tool is defined by \mathcal{P}_k and the corresponding subimage is expressed by

$$I_k(d, \theta), \quad (1)$$

where d is the measured depth and $\theta \in [0, 2\pi)$ is the azimuthal location of the sensor. Following acquisition, both variables are resampled and defined on a discrete grid. Assuming that the measurements are zero outside the non-imaged azimuthal locations, the output image $I(d, \theta)$ is given

by

$$I(d, \theta) = \sum_{k=0}^{N_p-1} I_k(d, \theta). \quad (2)$$

The image in (2) often appears misaligned since a scan line from each pad in the image is sampled at a different time by the tool due to a depth offset between the pads. The objective is therefore to shift each pad such that the image is aligned. The resulting image can be expressed as

$$\tilde{I}(d, \theta) = \sum_{k=0}^{N_p-1} I_k(d + \tau^k(d), \theta), \quad (3)$$

where the function $\tau^k(d)$ defines by how much the k -th subimage is shifted for each depth.

We focus on one particular setup of (3) shown in Fig. 1, where the number of pads $N_p = 8$ and there are two distinct depth levels. The even pads $\{\mathcal{P}_0, \mathcal{P}_2, \mathcal{P}_4, \mathcal{P}_6\}$ are located at the same depth (bottom of the tool), and the odd pads are a certain distance above the even pads. This allows (3) to be simplified as:

$$I(d, \theta) = \sum_{k=0}^3 I_{2k}(d, \theta) + \sum_{k=0}^3 I_{2k+1}(d + \tau(d), \theta). \quad (4)$$

Here, the even pads are chosen as reference, and the odd pads are aligned to the even pads by searching for a common shift $\tau(d)$. In the following section, we describe how this shift is estimated.

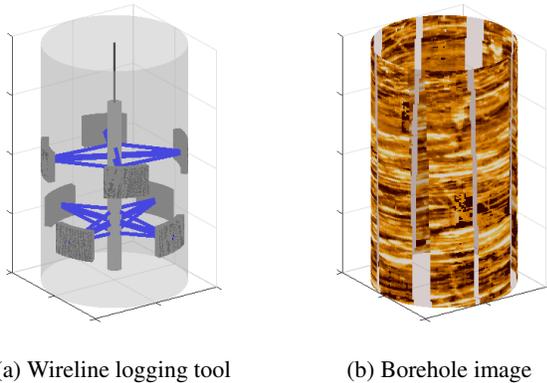


Fig. 1: Borehole images can be obtained with a wireline logging tool. The logging tool in (a) has eight pads; each of the pads creates a separate subimage. The full set of eight subimages is shown in (b). The subimages often appear misaligned due to the depth offset in the pads.

3. ALGORITHM DESCRIPTION

An overview of the method to estimate the shift is shown in Fig. 2. To simplify the algorithm description, we assume that

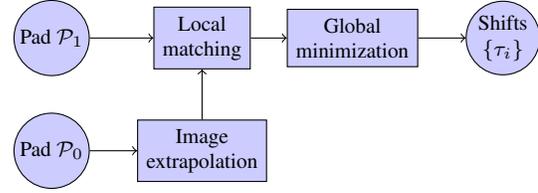


Fig. 2: Proposed method to align subimages \mathcal{P}_0 and \mathcal{P}_1 . In the first step, the boundary in \mathcal{P}_0 is extrapolated to create an overlap with \mathcal{P}_1 . Local feature matching is then performed on the overlap region. An additional smoothness constraint is imposed in the global minimization stage, in which the joint cost function is minimized by the Viterbi algorithm to give the estimated shifts $\{\tau_i\}$. The shifts are used to align the subimage \mathcal{P}_1 with \mathcal{P}_0 .

only pads \mathcal{P}_0 and \mathcal{P}_1 need to be aligned. In the first stage, the edges in \mathcal{P}_0 are extrapolated to create an overlap with the pixels in \mathcal{P}_1 . Given the overlap, a local feature matching can then be implemented. This stage selects a template around a feature point in \mathcal{P}_1 and searches for a similar patch in the extrapolated region of \mathcal{P}_0 along the depth dimension. A challenge in local matching is that there may be several good candidate matches. This is particularly the case in borehole images, where events such as structure and texture are often repeated along the well trajectory. Hence, minimizing the cost function locally for each feature point often results in an incorrectly aligned image.

This issue is solved by imposing a smoothness constraint on the estimated shifts. The combined local matching and regularization terms are jointly minimized by a Viterbi algorithm to give the output shift. Finally, the user can validate the aligned image and insert additional shifts as necessary. Although not explicitly shown in Fig. 2, the method can also be rerun with \mathcal{P}_1 selected as reference. The new shifts can then be used to evaluate a robust quality control (QC) metric by crosschecking the results [3].

3.1. Image extrapolation

The image extrapolation stage extends the image boundary in \mathcal{P}_0 to create an overlap with the data in \mathcal{P}_1 . Intuitively, this allows feature matching to be performed on the overlap region in the subsequent stage. An example of an image before and after extrapolation is shown in Fig. 3. The extrapolated image is defined by $\hat{I}_0(d, \theta)$.

In general, any image inpainting algorithm can be used to extend the boundary. We have used a low-complexity kernel regression method proposed in [4], where the missing pixels are estimated as a weighted sum of available neighboring data. The weights determine the reconstruction and are defined by the intersection of a Gaussian kernel at the location of the missing pixel. The parameters of the kernel are estimated from local image gradients, and this ensures that the extrapolation is in the direction of local edges.

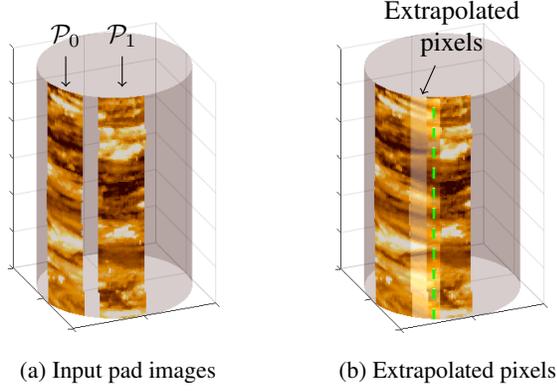


Fig. 3: Extrapolating the boundary of \mathcal{P}_0 allows local matching to be performed on the overlap region. The input pads \mathcal{P}_0 and \mathcal{P}_1 are shown in (a). In (b) the boundary in \mathcal{P}_0 is extrapolated to create an overlap with \mathcal{P}_1 using [4]. The inpainted pixels are illustrated by the lower-opacity segment. Local matching can then be applied along the dashed line in the overlap region.

3.2. Local matching

The aim is to find a correspondence between the extrapolated pixels in \mathcal{P}_0 and \mathcal{P}_1 along the dashed line in Fig. 3(b). The matching is performed at N_f feature points, where the location of the i -th feature point is (d_i, θ_i) . The feature points can be evenly spaced along the dashed line in the overlap region, or non-uniformly distributed at horizontal edges¹.

For each feature point, the method selects a patch of data centered at (d_i, θ_i) in \mathcal{P}_1 and searches for a similar patch in the extrapolated pixels of \mathcal{P}_0 . Metrics to measure patch similarity is a highly researched topic [1]. The metrics can take into account the pixel intensity or edge information [5], or they can be an aggregation of several models [6]. To limit the computational burden, we have used the following absolute difference metric:

$$J_i(\tau_i) = \sum_{(d,\theta) \in \mathcal{W}(d_i, \theta_i)} \left| \hat{J}_0(d, \theta) - I_1(d + \tau_i, \theta) \right|, \quad (5)$$

where $\mathcal{W}(d_i, \theta_i)$ is a window of size $(N_d \times N_\theta)$ centered at the feature point (d_i, θ_i) .

The local matching cost in (5), especially in the case of borehole images, contains many local minima. Therefore, minimizing the cost function directly for each feature point will often give an incorrectly aligned image. We discuss how this problem can be solved in the following section by imposing a smoothness constraint on the estimated shift.

3.3. Global minimization

It is common practice to impose a smoothness constraint on the estimated shift in stereo matching algorithms [1]. This

¹Horizontal edges are easier to match since they are less sensitive to extrapolation errors than smooth regions or vertical edges.

technique takes into account prior knowledge about the disparities and significantly improves the matching result. We use a similar approach in our problem and define a global cost function as

$$J(\tau_{1:N_f}) = \sum_{i=1}^{N_f} [J_i(\tau_i) + \lambda f(\tau_i)], \quad (6)$$

where $\tau_{1:N_f}$ is a set of shifts across all feature points, $f(\tau_i)$ is the new regularization term, and λ is a constant that determines the trade-off between the local matching cost and the regularization term. It is evident that when $\lambda = 0$, the cost is equivalent to a local matching in (5).

The regularization term $f(\tau_i)$ can be specifically designed for borehole images. In our formulation, it has the following form:

$$f(\tau_{i:i-1}) = f_S(\tau_{i:i-1}) + f_T(\tau_{i:i-1}), \quad (7)$$

where $f_S(\tau_{i:i-1})$ is a smoothness constraint and $f_T(\tau_{i:i-1})$ is an additional penalty term to enforce an invertible mapping. The first term is expressed as:

$$f_S(\tau_{i:i-1}) = g(|d_i - d_{i-1}|) |\tau_i - \tau_{i-1}|, \quad (8)$$

where $g(x)$ is a monotonically decreasing function with $g(0) = 1$. The estimated shifts are typically smooth, but occasionally there may be significant transitions due to sudden changes in the tool dynamics. This analysis supports using an ℓ_1 penalty term. The additional weighting by the $g(\cdot)$ function has been added to take into account the nonuniform spacing in the feature points. The scaling factor adds a large penalty to closely spaced feature points and reduces the regularization when $|d_i - d_{i-1}| \gg 0$.

The second penalty term $f_T(\tau_{i:i-1})$ ensures that there exists a bijection mapping between the original and shifted feature points. In most cases this assumption is satisfied since the logging tool is only moving in one direction. It can be shown that the mapping is invertible if the inequality

$$d_i + \tau_i > d_{i-1} + \tau_{i-1} \quad (9)$$

is satisfied. We enforce this constraint by setting the regularization

$$f_T(\tau_{i:i-1}) = \lambda_1 [(d_i + \tau_i) - (d_{i-1} + \tau_{i-1})]^2 \quad (10)$$

if (9) does not hold and zero otherwise. Here $\lambda_1 \gg 0$ is a large constant that sufficiently penalizes the cost function. Note that in traditional stereo matching, this condition is not applied because objects can be occluded due to scene geometry.

In the stereo matching community, the most widely used methods to minimize (6) is belief propagation or graph cuts [1][7]. In our formulation, we have used the Viterbi algorithm, which is a special case of belief propagation as discussed in [8]. The main idea of Viterbi is that, although

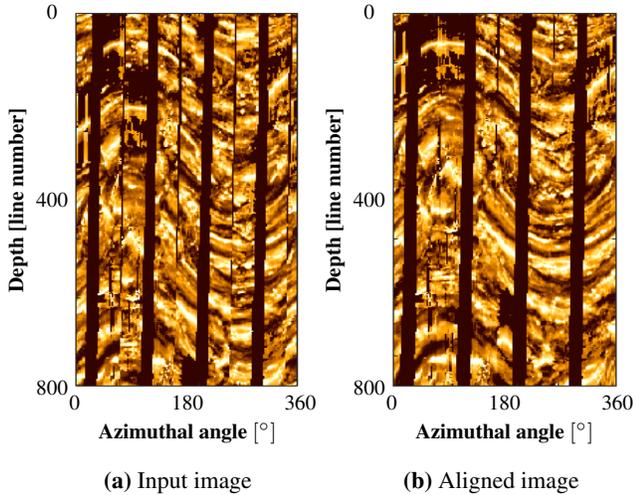


Fig. 4: Comparison between (a) input image and (b) aligned result.

the solution space grows exponentially with each new feature point, many of the solutions are suboptimal and do not need to be considered during minimization. The algorithm iteratively minimizes (6) at each feature point by only taking into account the feasible set. Finally, once all the feature points are considered, the path through the solution space (out of the feasible solutions) that has the lowest cost is taken as the output shift.

4. DISCUSSION

The proposed algorithm is available as a plug-in in the Techlog* wellbore software platform. The method has been tested on numerous datasets and has been shown to significantly reduce the processing time in comparison to manual alignment. As an example, the method aligned an 800-ft log (80,000 scan lines) in 60 seconds (Intel i7 CPU @ 3.00GHz). Reviewing the results, only two 5-ft data intervals were deemed as incorrectly aligned. As a comparison, a manual alignment of the same log would take 6 hours to complete. Two data intervals from this log are shown in Figs. 4 and 5. In both cases, the perceptual quality of the aligned image is significantly improved. The corresponding estimated shift to align the images is shown in Fig. 6.

The algorithm may have limitations in regions where the image has been corrupted or cannot be visually aligned. An image-based alignment may not be possible if the image has only vertical features. However, after the algorithm is run, these regions can be easily identified by the QC metric, where the crosscheck between forward and backward alignment will indicate an error. The user can then increase the regularization parameter λ or manually align the intervals.

*Mark of Schlumberger.

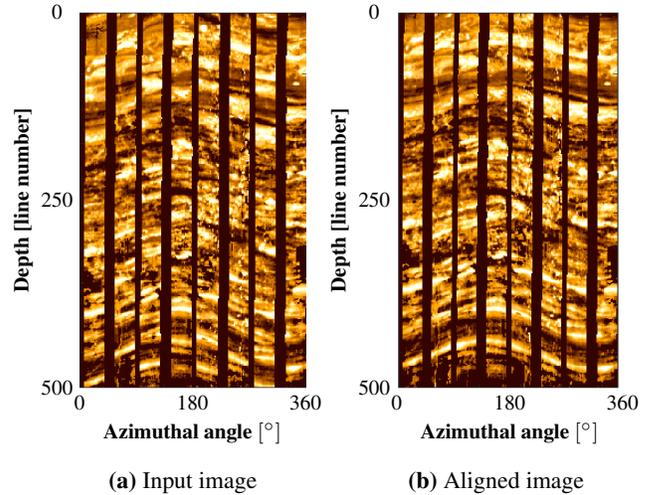


Fig. 5: Comparison between (a) input image and (b) aligned result.

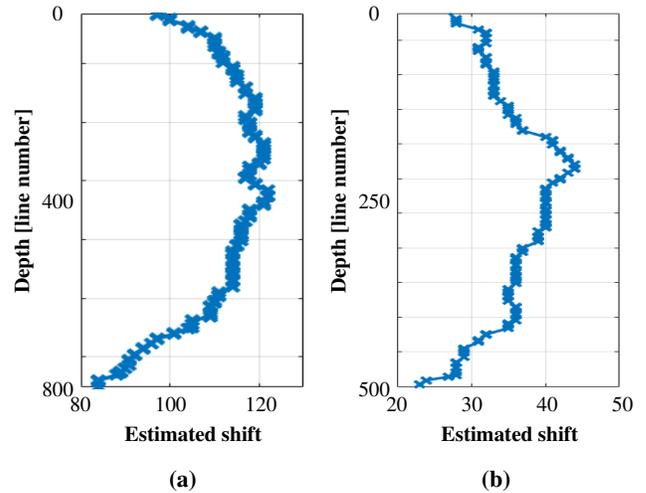


Fig. 6: Estimated shift using proposed method; (a) aligns Fig. 4(a) and (b) aligns Fig. 5(a).

5. CONCLUSION

This paper proposed an automated algorithm to align features in borehole images from wireline logging tools. The method first extrapolates the image boundaries to create an overlap region between the pads. Feature matching is then performed on the overlap region in two stages. A local matching is run on finite set of feature points to identify possible candidate solutions. The local solutions may have many local minima due to repeating patterns in borehole images. To find a consistent result across all the feature points, the method performs a global optimization with additional constraints on the smoothness of the estimated shift. The full method has been implemented as plug-in in the Techlog* wellbore software platform and tested on several challenging datasets. In comparison to manual alignment, the algorithm can reduce the processing time for a 1,000-ft image log ($\approx 100,000$ scan lines) log from 6 hours to 1 minute. Future work includes improving the alignment for images with no crossing features between the pads and low-quality data.

6. REFERENCES

- [1] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *International Journal of Computer Vision*, vol. 47, no. 1, pp. 7–42, 2002.
- [2] Y. Poleg and S. Peleg, "Alignment and mosaicing of non-overlapping images," in *IEEE International Conference on Computational Photography*, April 2012, pp. 1–8.
- [3] G. Egnal and R. P. Wildes, "Detecting binocular half-occlusions: empirical comparisons of five approaches," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 8, pp. 1127–1133, Aug 2002.
- [4] H. Takeda, S. Farsiu, and P. Milanfar, "Kernel regression for image processing and reconstruction," *IEEE Transactions on Image Processing*, vol. 16, no. 2, pp. 349–366, Feb 2007.
- [5] L. Xu, J. Jia, and Y. Matsushita, "Motion detail preserving optical flow estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 9, pp. 1744–1757, Sept 2012.
- [6] F. Tombari, S. Mattoccia, L. Di Stefano, and E. Addimanda, "Classification and evaluation of cost aggregation methods for stereo correspondence," in *IEEE Conference on Computer Vision and Pattern Recognition*, June 2008, pp. 1–8.
- [7] M. F. Tappen and W. T. Freeman, "Comparison of graph cuts with belief propagation for stereo, using identical mrf parameters," in *IEEE International Conference on Computer Vision*, Oct 2003, vol. 2, pp. 900–906.
- [8] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Exploring artificial intelligence in the new millennium," chapter Understanding Belief Propagation and Its Generalizations, pp. 239–269. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2003.