Non-local means filter for trim statics

Yunsong Huang, Xin Wang, and Gerard T. Schuster*
King Abdullah University of Science and Technology

SUMMARY

Structures will be mispositioned across prestack migration gathers in the presence of inaccuracies in the velocity model. Stacking these misaligned gathers runs the risk of destroying important structures in the stacked migration image. To mitigate this problem, we propose a trim statics inspired by the non-local means algorithm originally developed for image denoising. This method differs from the conventional one in two fundamental respects. First, the trim statics are computed by comparing image patches instead of individual image traces. Second, no global pilot trace is needed because only two migration images at a time participate in trim statics and are stacked into one image. A multitude of migration images are stacked recursively in this two-to-one fashion. Tests with a Gulf of Mexico dataset show a noticeable improvement in the feature coherency of the stacked migration image.

INTRODUCTION

Stacking a set of prestack migration images is typically used to reduce the migration artifacts and to boost the signal-to-noise ratio (SNR) of the stacked image (Yilmaz, 2001). When the migration velocity is inaccurate, however, the reflectors will be imaged at wrong depths that vary with each prestack migration image. For example, when the migration velocity is too slow, the reflectors will be imaged at shallower depths for a large source-to-receiver offset than for a small offset. Simply stacking those varyingly mispositioned reflector can smear out or disrupt important image features such as reflectors and faults.

This problem may be tackled by trim statics, a procedure to correct the residual statics on input gathers (Marsden, 1993; Cox, 1999). Due to a variety of factors such as anisotropy and near surface inhomogeneity, time corrections need to be applied to traces to enhance the quality of the stack response. A time correction is found by locating the maximal cross-correlation between an individual trace and a pilot trace, which is formed by stacking the input gather. This procedure can be generalized to the image domain, where the 'time' is replaced by 'depth', and 'within a gather' is replaced by 'across the common image gathers (CIGs)'. Such cross-correlation methods have been employed, for example, in Hall (2006), Perez and Marfurt (2008), and Hale (2009). One potential limitation, as stressed by Ursenbach and Bancroft (2001), is that the cross-correlation procedure in trim statics can be misled by aligning the noise instead of the signal. A second limitation is that the pilot traces obtained by stacking the input gather may be poorer in quality than individual traces because important features may have been lost in the pilot traces.

To address the first limitation, we propose to enhance the robustness of trim statics by adopting the non-local means (NLM) filter (Buades et al., 2005). This filter has seen usage in the geophysical community to denoise seismic data (Bonar and Sacchi, 2012). In image denoising, the NLM filter produces a filtered image pixel of interest by the weighted average of pixels in a search neighborhood, where the weighting factor connecting the two pixels in question (i.e., the pixel of interest and a pixel in the neighborhood) is determined by the 'similarity' between them. This 'similarity' is computed based on the L_2 distance between the two image patches (deemed as vectors) surrounding the two pixels in question, respectively. The smaller this distance is, the more similar the two pixels are regarded, and therefore the larger is the weighting factor connecting them. In addition to the inherent noise attenuating capability of the NLM filter, the patch-based similarity measure is more robust than the alternative based on individual image traces.

To address the second limitation, we propose a recursive stacking strategy that obviates the need for a global pilot trace. We first investigate a binary stacking, namely, how a pair of migration images can be optimally stacked. After this is accomplished, a number of migration images can be stacked recursively. An example is illustrated in Figure 1. Four migration images are stacked into two (B1 and B2), which in turn are stacked into one image A1.

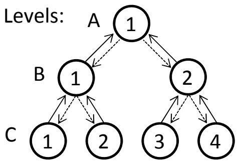


Figure 1: The workflow of stacking four migration images C1, C2, C3, and C4. The arrows are explained in Figure 3(c).

The rest of this extended abstract is organized as follows. The theory section presents the formalism of NLM and binary stacking. This is followed by the results section that demonstrates a noticeable improvement in feature coherency and the sharpness of the stacked migration image. The final section presents the conclusions.

THEORY

The NLM Filter

Given the prestack migration image $\mathbf{m}^a$, at a pixel i, the NLM filtered value $\overline{m_i^a}$ is a weighted average

$$\overline{m_i^a} = \sum_j W_{ij} m_j^a, \tag{1}$$

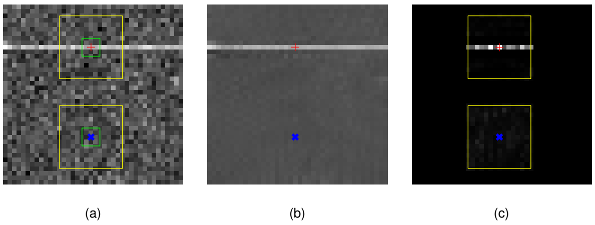


Figure 2: NLM filtering weights favor repetitive structures. (a) A noisy image $\mathbf{m}$ with a horizontal line. Two pixels of interest are marked, in all panels, by '+' and 'x', respectively. Green squares circumscribe the image patches around '+' and 'x'. Yellow squares circumscribe the search neighborhoods around red '+' and blue 'x'. (b) the filtered image $\overline{\mathbf{m}}$ by NLM. (c) the filtering weights for the two pixels. Large weights are plotted in white; zero weights are in black.

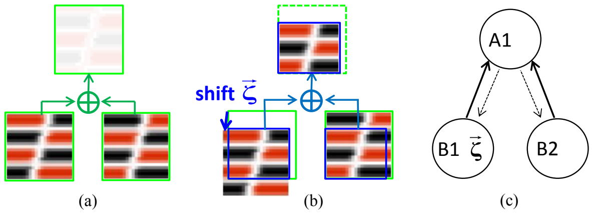


Figure 3: Stacking strategies. The color scheme for image patches in (a) and (b): black, white, and red represent negative, zero, and positive amplitudes, respectively. (a) Simple stacking. Out-of-phase image patches will be washed out. (b) Stacking after first shifting one image patches by an amount $\vec{\zeta}$ to make the two patches aligned. (c) Symbolization of the procedure in (b). The two solid arrows represent stacking, whereas the dashed arrows represent reconstruction. From the stacked patch A1 and the shift $\vec{\zeta}$ that B1 remembers, the patch B1 can be reconstructed as {patch A1 shifted by $-\vec{\zeta}$}, whereas B2 is simply reconstructed as A1, in this example.

where W_{ij} is the NLM filtering weight, and the summing index j runs in a local search neighborhood around i, of size $S_x \times S_z$. Examples of search neighborhoods are indicated by yellow squares in Figures 2(a). Before describing the expression of W_{ij}, we first illustrate the behavior of NLM with an example in Figure 2. First, note the weight pattern, which pertains to the red '+' pixel and is circumscribed by the upper yellow square in Figure 2(c). The inner product between this weight pattern and the image region delimited by the upper yellow square in Figure 2(a) produces the filtered red '+' pixel in Figure 2(b). Note also that in Figure 2(c) the two weight patterns (circumscribed in the two yellow squares) are different. This contrasts with linear space-invariant filters expressed as $\overline{m^a_i} = \sum_j W(i-j)\, m^a_j$, where the weight pattern $W(i-j)$ is space-invariant.

Next, the NLM weights (Buades et al., 2005) are given as

$$W_{ij} = \frac{1}{\sum_k \tilde{W}_{ik}} \tilde{W}_{ij}, \tag{2}$$

$$\text{where } \tilde{W}_{ij} = \exp\!\left(-\frac{||\mathbf{p}^a(i) - \mathbf{p}^a(j)||^2}{2\sigma^2}\right). \tag{3}$$

Here, $\mathbf{p}^a(i)$ represents amplitudes of the image patch of size $P \times P$ centered at the i^{th} pixel in $\mathbf{m}^a$. Patch examples are indicated by green squares in Figure 2(a). Note in passing that between patches the L_2 distance in equation 3 can be weighted, for example as $||e(x,z)||^2_\Delta \stackrel{\text{def}}{=} \iint \Delta(x)\Delta(z)e^2(x,z)\,\mathrm{d}x\mathrm{d}z$, where $\Delta(\cdot)$ denotes the triangle weighting function and $e(x,z)$ denotes any function of interest. This weighted patch is used in our numerical simulation.

In image denoising, the patch has to be "large enough to be robust to noise" (Buades et al., 2005) but not so large and content-rich that it would be hard pressed to find a replica within the search neighborhood. The search neighborhood covers an area where such patches typically recur. In terms of pixels, $P \times P \simeq 7 \times 7$ and $S_x \times S_z \simeq 21 \times 21$ are among the popular choices in image denoising. In our application however, P, S_x, and S_z are chosen differently, as will be described in the Results section.

The upper yellow square in Figure 2(c) demonstrates that the NLM weights concentrate on coherent structures. This is because, for example, if one moves the upper green square in Figures 2(a) along the horizontal white line one would see similar patches recurring, and between two similar patches is a stronger weight effected, according to equation 3.

To allow stacking two images to one, the NLM formalism is extended to deal with two images $\mathbf{m}^a$ and $\mathbf{m}^b$, and the goal is to obtain a filtered version $\widetilde{\mathbf{m}^a}$ by weighted average of pixels in $\mathbf{m}^b$, expressed as[1]

$$\widetilde{m^a_i} = \sum_j W_{ij} m^b_j. \tag{4}$$

In this case, equation 2 remains intact as the normalization, but equation 3 is replaced by

$$\tilde{W}_{ij} = \exp\!\left(-\frac{||\mathbf{p}^a(i) - \mathbf{p}^b(j)||^2}{2\sigma^2}\right), \tag{5}$$

where $\mathbf{p}^b(j)$ represents amplitudes of the image patch centered at the j^{th} pixel in $\mathbf{m}^b$.

The $\widetilde{m^a_i}$ in equation 4 can still be deemed as a filtered version of m^a_i even though it relies on the weighted average of a different image $\mathbf{m}^b$, because the weights W_{ij} are tuned in to patches in $\mathbf{m}^b$ that are similar to $\mathbf{p}^a(i)$. To differentiate the standard NLM filtering from the extended formalism, we call the former 'endo-filtering', and the latter 'exo-filtering'. After both filtered versions are computed, we obtain the fully stacked version

$$\widehat{\mathbf{m}^a} = \frac{1}{2}(\overline{\mathbf{m}^a} + \widetilde{\mathbf{m}^a}). \tag{6}$$

The selectivity of W_{ij}, in the case of exo-filtering, with respect to $||\mathbf{p}^a(i) - \mathbf{p}^b(j)||$ is controlled by the parameter σ in equation 5. Assuming there is no tie in the minimal L_2 distance

[1] We reuse the same symbols for the weights for notational economy.

$||\mathbf{p}^a(i) - \mathbf{p}^b(j)||$, we can show that

$$\lim_{\sigma\to 0} W_{ij} = \begin{cases} 1 & \text{if } j = \arg\min_j ||\mathbf{p}^a(i) - \mathbf{p}^b(j)||, \\ 0 & \text{otherwise,} \end{cases} \tag{7}$$

which reduces equation 4 to

$$\lim_{\sigma\to 0} \widetilde{m_i^a} = m_j^b, \text{ where} \tag{8}$$

$$j = \arg\min_{j} ||\mathbf{p}^a(i) - \mathbf{p}^b(j)||. \tag{9}$$

Therefore in this limit, the exo-filtering of NLM is equivalent to cross-correlating an image patch of interest in $\mathbf{m}^a$ with patches in $\mathbf{m}^b$ and selecting the best matching patch in $\mathbf{m}^b$.

The Binary Stacking

Note that in equation 4 (and consequently in equation 6), images $\mathbf{m}^a$ and $\mathbf{m}^b$ are on unequal footing because $\mathbf{p}^a(i)$ acts as the reference that $\mathbf{p}^b(j)$ tries to match. (Note in equation 4, the index i is fixed while j is a summation index.) This is illustrated in Figure 3(b), where the patch on the left moves by $\vec{\zeta}$ to best match the patch on the right. This movement is remembered by the patch on the left. This bookkeeping is useful in the following scenario.

Let $\mathbf{m}^a$ and $\mathbf{m}^b$ be two CIGs and let a diffractor of interest be located at i^a and i^b in $\mathbf{m}^a$ and $\mathbf{m}^b$, respectively. Suppose this diffractor is physically located at i^o in the subsurface. Owing to inaccuracy in the velocity model and the differences in the CIG setups, in general we have $i^o \neq i^a \neq i^b$. Take a patch $\mathbf{p}^a(i^a)$, if successful we can find the best matching patch $\mathbf{p}^b(i^b)$ according to equation 9. Stacking the central pixels in $\mathbf{p}^a(i^a)$ and $\mathbf{p}^b(i^b)$, as prescribed in equation 6, produces the correct value for the diffractor at i^a in $\mathbf{m}^a$. This procedure facilitates the LSM of $\mathbf{m}^a$ when LSM is in use, because this stacking attenuates migration artifacts in $\mathbf{m}^a$ as intended. Note that the diffractor is imaged and stacked at i^a, which is not at the actual location i^o. This is not a problem, however, because LSM does not attempt to invert the velocity model. As a wrong velocity model is being used, i^a, rather than i^o, is the apparently consistent location for this diffractor. This analysis brings out a problem for the LSM of $\mathbf{m}^b$, because the apparently consistent location for this diffractor in $\mathbf{m}^b$ is at i^b, which is at odds with where the stacked value is placed, i.e., at i^a in $\mathbf{m}^a$. To circumvent this problem, it must be remembered the displacement from i^a to $j = i^b$. With this piece of information, the stacked value can be correctly attributed to the pixel in $\mathbf{m}^b$ at $i^a + (i^b - i^a)$, where the parenthesised quantity is the remembered displacement. This is outlined in Figure 3(c).

To make images $\mathbf{m}^a$ and $\mathbf{m}^b$ to be on equal footing we try out both points of view as follows.

1. Take $\mathbf{m}^a$ as the reference to obtain the stacked image $\widehat{\mathbf{m}^a}$.
2. Take $\mathbf{m}^b$ as the reference to obtain the stacked image $\widehat{\mathbf{m}^b}$.
3. Probe the sharpness of the stacked migration image by an ad hoc function $J_s(\cdot)$. If $J_s(\widehat{\mathbf{m}^a}) > J_s(\widehat{\mathbf{m}^b})$, the first approach is adopted; otherwise the second.

This ad hoc function is expressed as

$$J_s(\mathbf{m}) = P_{75}(|\frac{\partial^2}{\partial z^2}\mathbf{m}|), \tag{10}$$

where $P_{75}(\cdot)$ denotes the 75th percentile. Evaluating this function entails the following steps.

1. Take $\frac{\partial^2}{\partial z^2}\mathbf{m}$, where $\mathbf{m}$ is an image of size $M \times N$. Let the result be $\mathbf{c}$, of size $M \times N$ as well.
2. Let $\mathbf{b} = |\mathbf{c}|$, where $|\cdot|$ denotes element-wise absolute value. Cast $\mathbf{b}$ as an $MN \times 1$ array.
3. Let $\mathbf{d}$ be the sorted array in ascending order of $\mathbf{b}$.
4. We obtain $J_s(\mathbf{m}) = d_l$, where $l = \lceil 0.75MN \rceil$. This is known as the 75 percentile of $\mathbf{b}$.

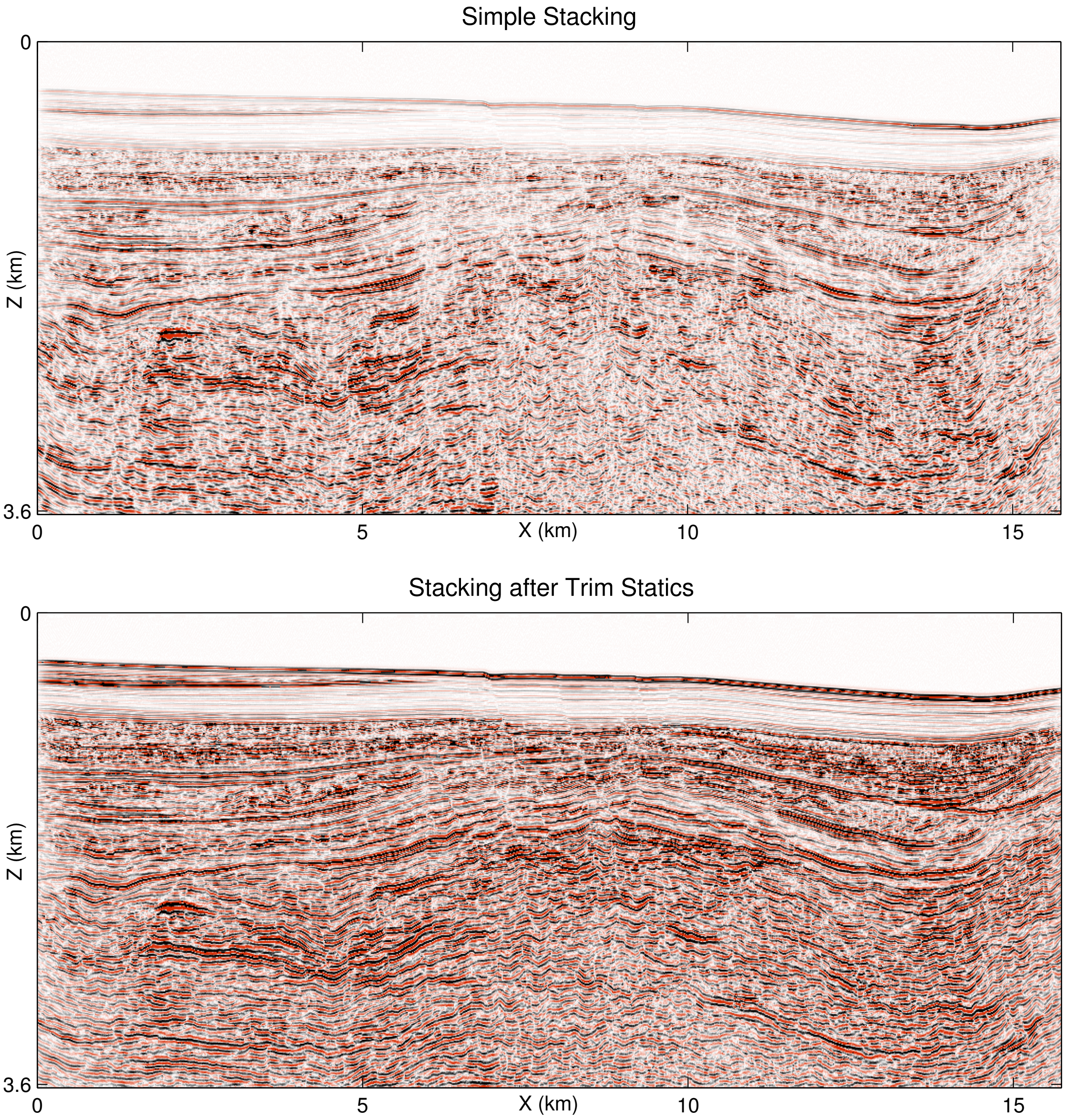
Note that the 100th percentile is equivalent to the maximum, which is not robust to outliers, whereas the 50th percentile is equivalent to the median, which might not be sensitive enough to pick up subtle increases in $|\partial^2/\partial z^2 \widehat{\mathbf{m}}|$. To strike a balance, we choose the 75th percentile.

RESULTS

We test the proposed NLM trim statics on a Gulf of Mexico (GOM) dataset. The size of the velocity model is 3.63 km in depth × 15.7 km in width, with 1260 receivers on the surface spacing at 12.5 m. The source wavelet is extracted from the near offset water-bottom reflection. Thirty-one plane-wave gathers are generated at the tenth iteration of a plane-wave Kirchhoff LSM. The plane-wave ray parameters vary evenly from -0.33 ms/m to 0.33 ms/m.

The parameters regarding the NLM algorithm are chosen as follows. The patch size is 17×17, which is about four wavelengths across, as a wavelength extends about 3 pixels in the shallow part of the model and about 5 pixels in the deeper part. As for the search neighborhood, the width is smaller than the height, as the depth of the features are more likely to be mispositioned. The sizes also depend on the stacking stride. Refer to Figure 3, the stride on level B, i.e., the difference in CIG parameters (e.g., source-to-receiver offsets) between neighboring nodes such as B1 and B2, is larger than that on level C. Larger stride implies larger relative mispositioning, and therefore larger size of the search neighborhood is appropriate. Ascending the stacking tree, the width of the search window increases from 1 to 7, while the height increases from 3 to 9. Finally, the σ in equation 3 is chosen to be 5 percent of the standard deviation of the prestack image pixels.

Stacking straightforwardly the 31 prestack migration images leads to a stacked image shown in the upper panel of Figure 4, whereas our proposed stacking after NLM trim statics leads to a stacked image shown in the lower panel of Figure 4. Comparing the two, we see evidently that in the former image quite a few reflectors are smeared and disrupted, and some faults are smeared, whereas in the latter image the continuity of reflectors and the distinctiveness of faults are noticeably improved. The downside of the former approach is illustrated in Figure 3(a).


Simple Stacking

Stacking after Trim Statics

Figure 4: Migration images stacked by two methods: (upper panel) simple stacking, and (lower panel) NLM trim statics, based on 31 plane-wave gathers of a GOM dataset.

CONCLUSIONS

We show that the mispositioning of reflectors in prestack migration images due to an inaccurate velocity model can be corrected by a trim statics based on image patches utilized by the non-local means (NLM) filter and a recursive binary stacking scheme that obviates the need for a pilot trace.

Tests with a GOM dataset demonstrate a noticeable improvement in the continuity of reflector and the distinctiveness of faults.

The limitation, however, is that trim statics will not correct mispositioning of reflectors migrated with strong velocity errors. The reflectors can be clearly revealed by migration+trim statics, but their locations and geometry might still be wrong. In this case, common receiver gathers should be migrated, trim statics with different parameters should be applied, and the final image should be similar to the final one using common shot gathers. If not, then this indicates erroneous positions of reflectors.

ACKNOWLEDGMENTS

We wish to thank the sponsors of Center for Subsurface Imaging and Fluid Modeling (CSIM) at KAUST for their financial support.