Efficient Migration*

Yazil Abbas[1]

Search and Discovery Article #40816 (2011)
Posted October 17, 2011

*Adapted from extended abstract presented at GEO-India, Greater Noida, New Delhi, India, January 12-14, 2011

[1]Department Of Earth Sciences, IIT Roorkee (yazilabbas@yahoo.com)

Introduction

Approximately 85-95% of a Seismic Data Processing center's computing power is used for pre-stack migration (in various forms) – a huge capital investment. Migration run times are typically 2000-9000 msec/trace. A 3D Kirchhoff PSTM for 1000 sq kms could take ~7000 days or ~20 years to run on one node. Therefore we need to use 100's of nodes to make the job run in a sensible time. With these statistics it is easy to understand that any re-run at this level has a significant impact on a total project and depending on the capabilities of a particular processing center it also has potential to affect other projects needing the same migration resources. In many cases the need for re-migration arise due to the presence of some noise contamination left in the processed dataset which ultimately gives poor migration results. So what we do is correct the errors in pre-migrated data and re-migrate the whole data set again. And this way we are spending money on the same job and also not generating any revenue. Since, problems do occur due to many variables of processing, we need to create a way to remigrate in a cost effective manner. This paper presents a method to ensure that such large money losses are reduced.

The Solution

The thought that surrounds this theory in fairly simple and it's referred to here as A, B, and C. Let's see how:

In Figure A and Figure B I have tried to schematically depict the flow of the thought to solve the above mentioned problem. In Figure A we have a seismic data pool "C" and a corrupted portion "B". Lets say B has got a time shift which has not been corrected and the data was migrated as it is. Now, with the wrong results in our hand what should we do to get to the Migration of "C+A" where A is the corrected version of the shifted B. The conventional way is to correct B and re- migrate the whole data set "C+A". Migration places energy at its right position. Every point in a model contributes energy which ultimately gets cancelled or summed up by destructive and constructive interference processes. And hence we get a final picture of true geology of the area. All in all we can say that what we require is the presence of right contribution of the right part of the reflector or any event as we say in seismic. So now if we just see our problem then we can easily judge that what we require is the left over part of the portion “B” which makes it “A”. So why not take the difference of the two by just picking out “B” in particular and correcting it to “A” and taking the difference. Now we can estimate the contribution of this much part by doing migration. Next adding the Migration of this difference to the Bad Migrated data will serve our purpose.

Mig(C+B) + Mig(A-B) = Mig (C+A)

Example

Figures C, D and E are some RMS plots measured over 3-D seismic data.

We will follow the algorithm shown above but here our starting point will be the error free data and then we will introduce a known error into it. The erroneous data thus obtained is subjected to the above algorithm and then we match the results of the above algorithm with the starting point of our endeavour that is error free data. So, we have here three migrated outputs viz. migrated output of error free data, migrated output of erroneous data, migrated output of the difference of pre-migrated error-free data, and erroneous data.

The data in Figure C is 3-D seismic data and in the RMS plots we have inlines at abscissa and crosslines at ordinate. The first plot Figure D presents the RMS plot of an error free dataset. The portion subjected to error in the pre-migration data of this plot will be along a central inline. We have introduced a bulk shift of around 10 msec. This erroneous data is dealt with according to the algorithm and then compared to the first figure to check the efficiency of the algorithm.

By migrating the difference Figure E and adding it to the migrated incorrect data (C+B), the wanted or correctly migrated data is achieved at a much lower cost. Below are some figures of cost savings by doing the above mentioned process.

Statistics of the Migration Run Times

For Time Shifted inline

Migration of whole dataset	500 cpu hrs.
Migration of difference	100 cpu hrs.

Now, we have results for cases in which error was introduced in particular traces. They are as follows:

For single inline with a single trace (portion subject to error introduction) shifted by 4 msec

Migration of whole dataset	2 cpu hrs.
Migration of difference of traces	7 cpu mins.

For single trace

Migration of trace	8 cpu mins.
Migration of difference	7 cpu mins.

We can say time is reduced to 20% approximately in the migration job for the difference of the corrected and the bad data.

Conclusions

We propose here an efficient way of saving a large amount of money which is the money generated due to errors. The algorithm proposed above shows how we can use the corrupted migration output which we were about to discard. What we have to do is to just go to the pre-migrated domain and correct the errors which were the cause of our present corrupted migration output. Then we take the difference of the corrected and the erroneous portions, migrate this difference, and then add this to our present corrupted output. The resultant figure will be exactly the same as if we have migrated the error-free data. This way we save a lot of time and huge amount of money which would have been invested if we were migrating the whole data set again after correction of those small erroneous positions in our data which were left uncorrected in our first run of migration. Now we just have to migrate the difference, which is of course a small job. This procedure can be a big fortune for any seismic processing enterprise.

Acknowledgements

I thank WesternGeco for allowing me to produce these results and for authorization to present this work.

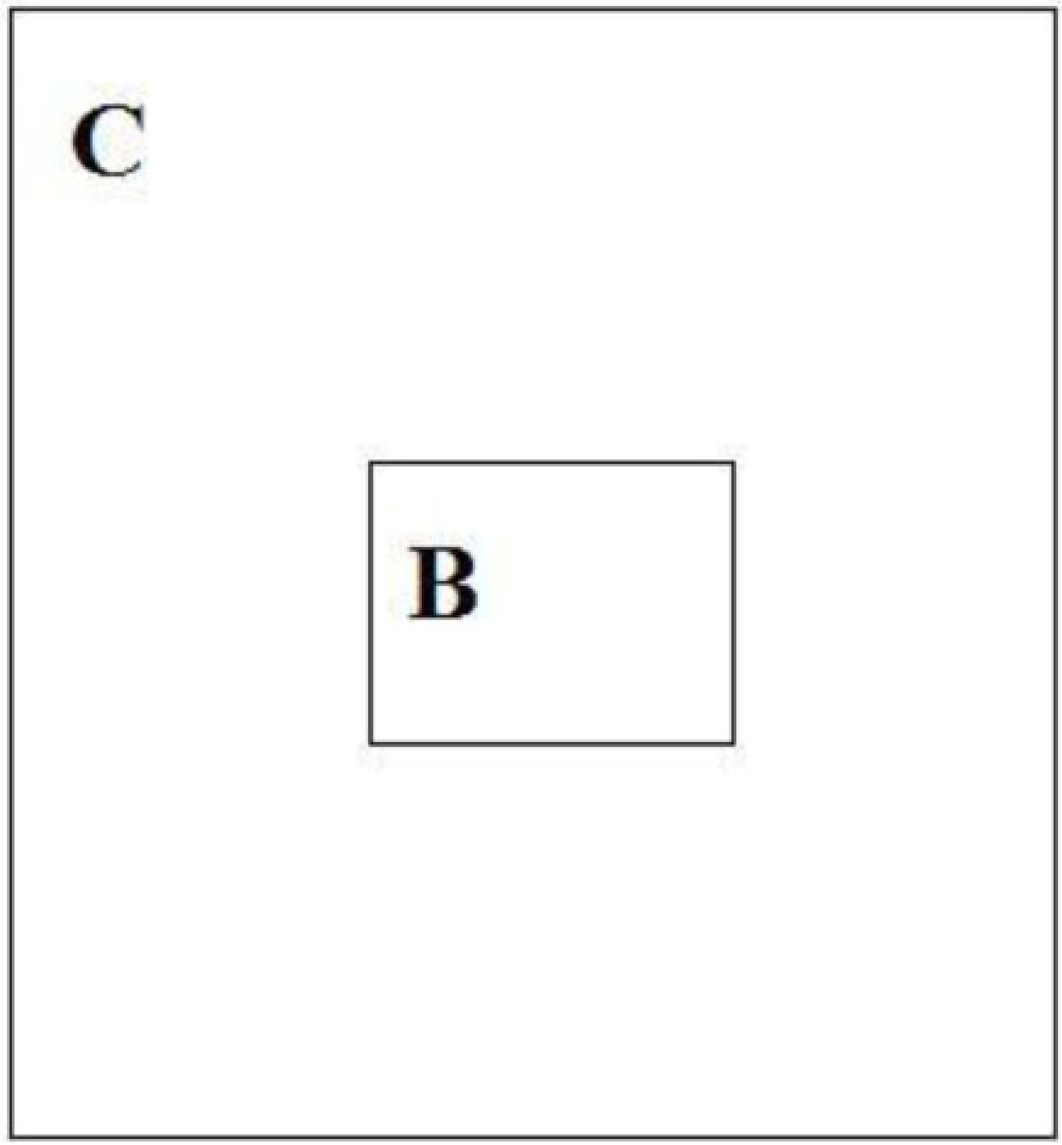

Figure A. Incorrect Pre-Migration Data.

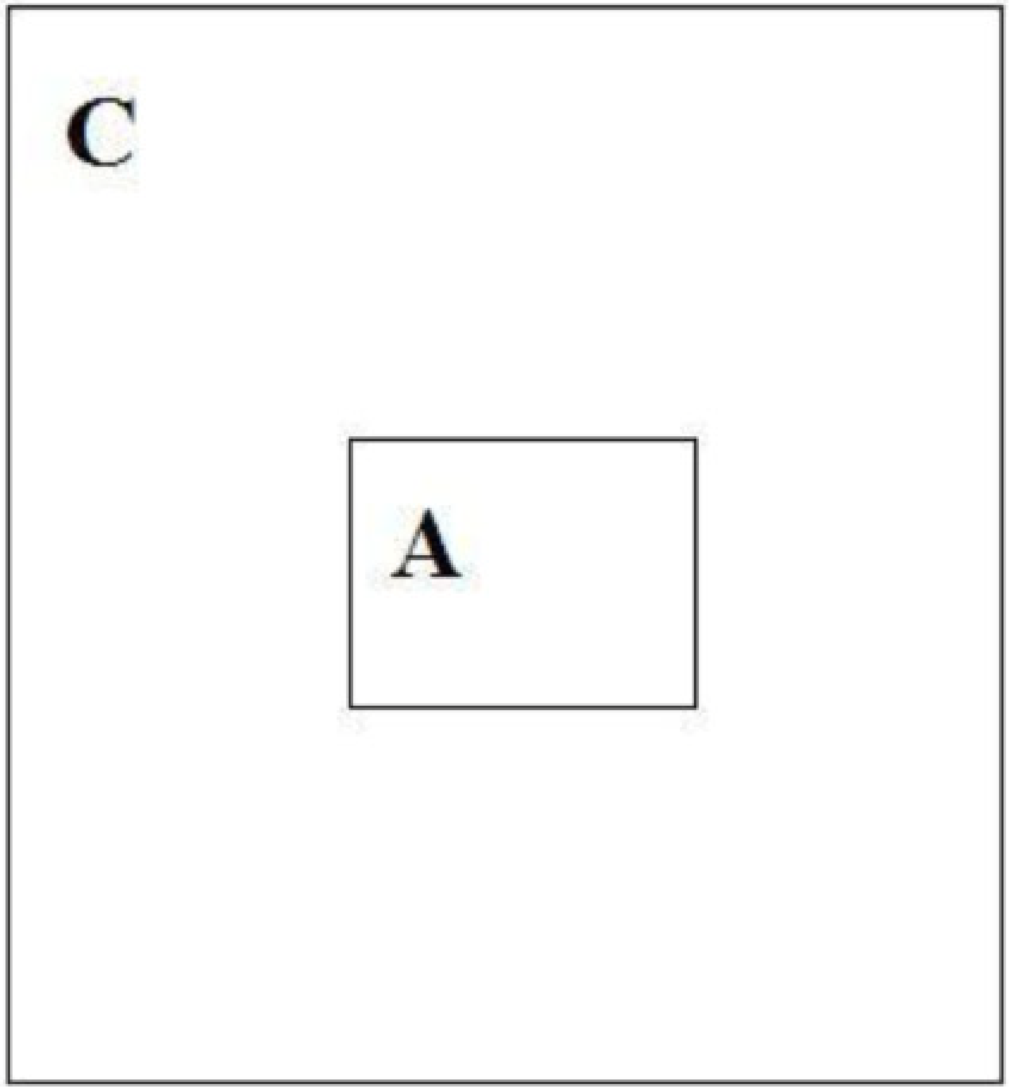

Figure B. Correct Pre-Migration Data.

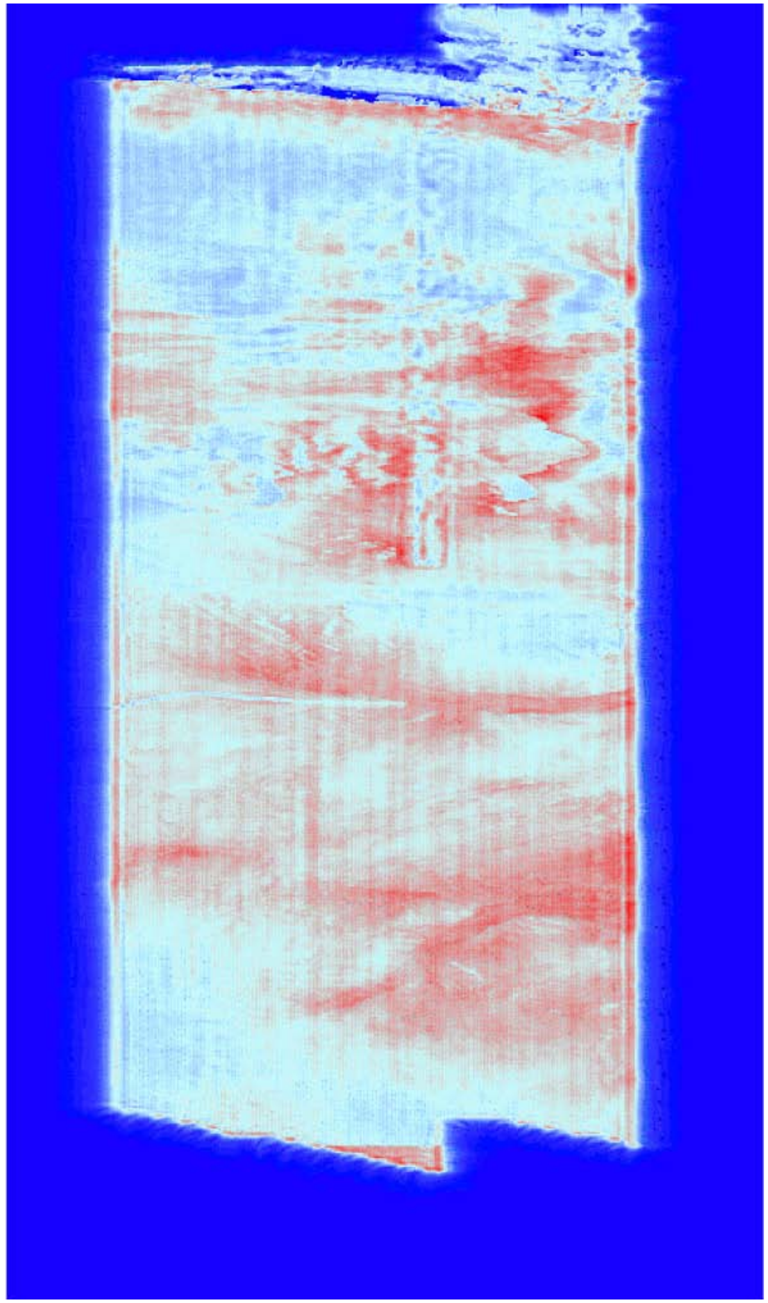

Figure C. RMS plot of a migrated error-free data or Mig (C+A).This is the pre-stack migration of correct data. We will be corrupting this by introducing a time shift along a central Inline.

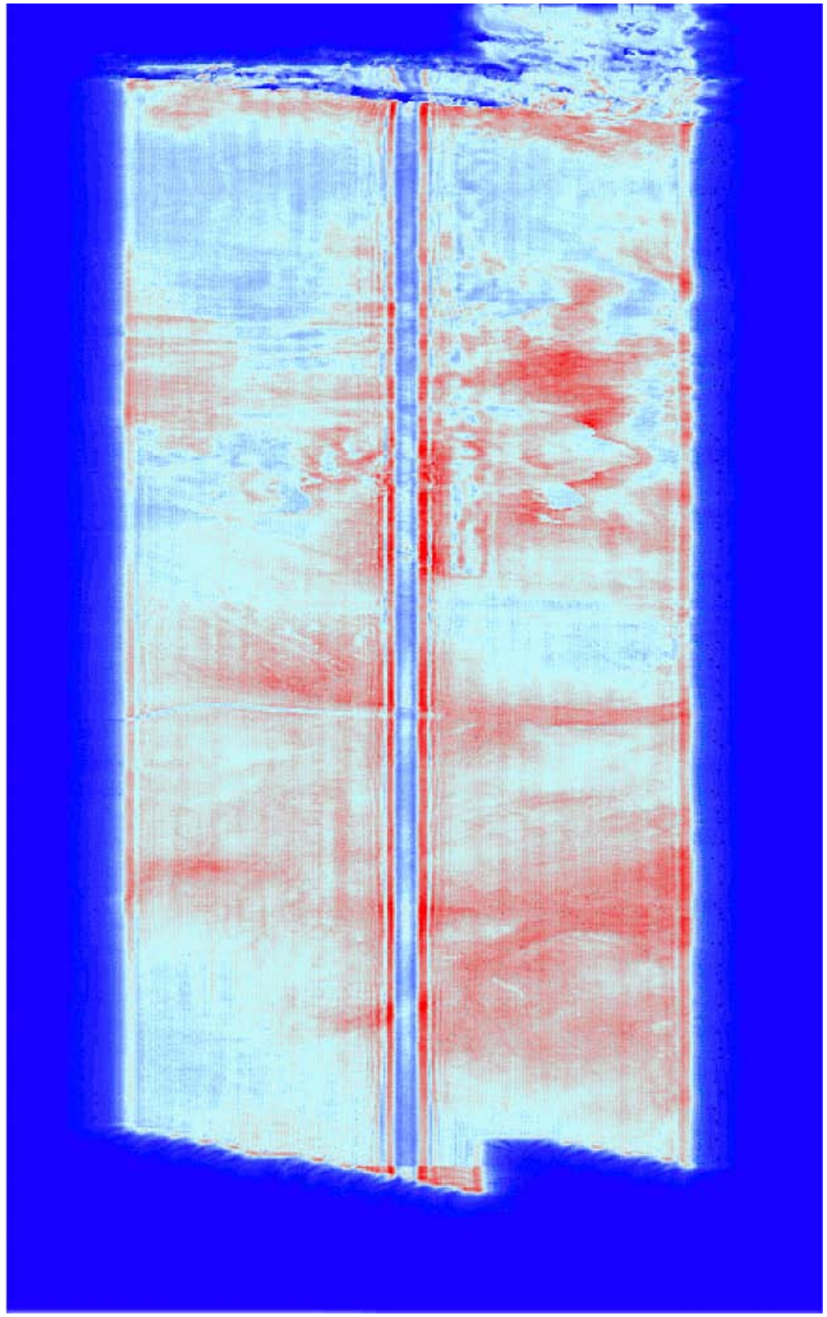

Figure D. RMS plot of Mig(C+B) or the Migration of the corrupted version of the normal data. You can see the anomaly along the central Inline.

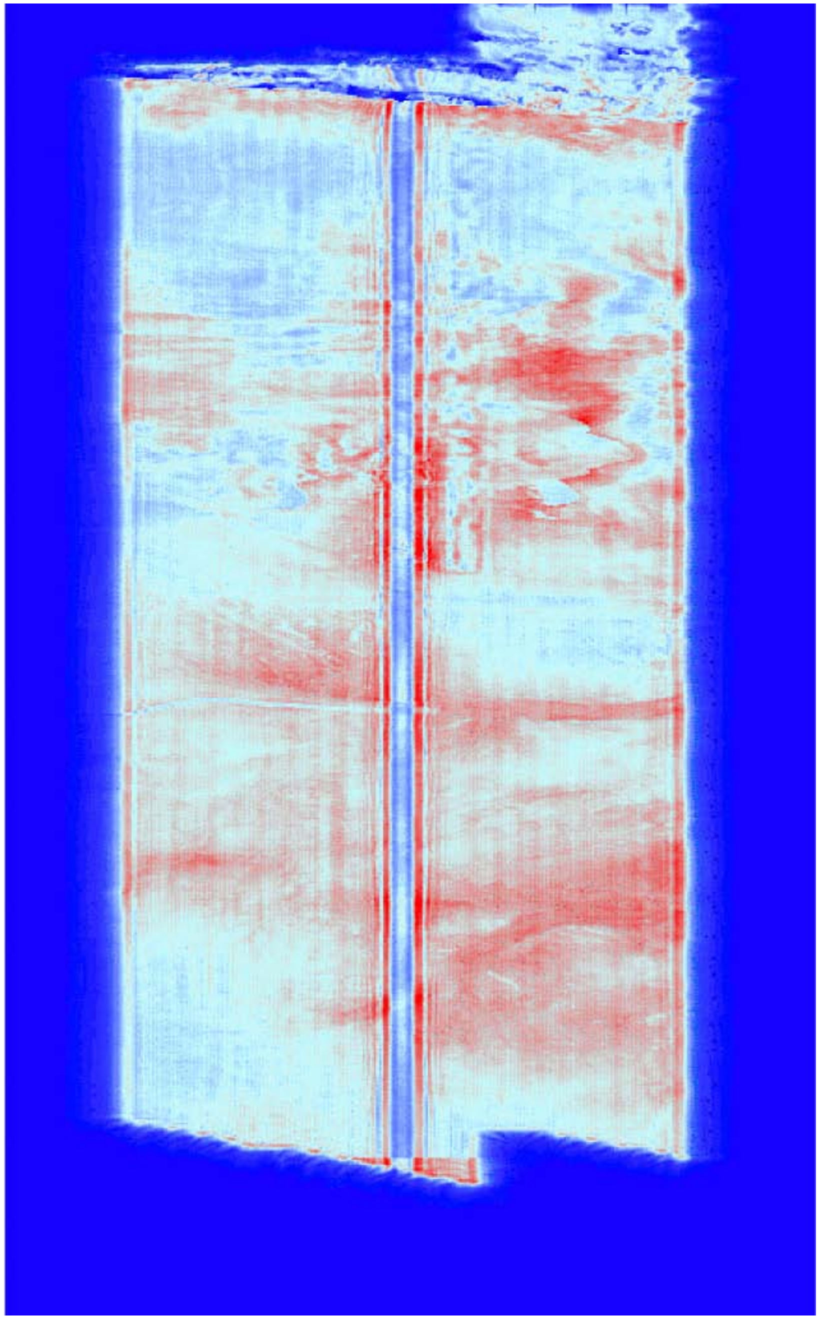

Figure E. RMS plot after the migrated difference between B and A has been added to the incorrect migration (C+B) or Mig(C+B) + Mig (A-B). Note how perfectly it matches with Figure C Mig(C+A).