Dip-Guided Auto-tracker for Seismic Interpretation

Brett Hern, Barton Payne, Anne Dutranois Coumont, Mike Li

Brett.Hern@Chevron.com

November 18th 2020
About the Author

Computational Geoscientist
Chevron Technical Center

Biography
• PhD Electrical Engineering with 7 years experience at Chevron
• Professional Interests: Seismic Interpretation, Computational Geometry, Static Reservoir Modeling, Software Engineering, Artificial Intelligence

© 2020 Chevron
Acknowledgements

• Project team members
  – Barton Payne, Anne Dutranois-Coumont, Bin Qiu
• Feature advice and feedback:
  – Barton Payne, Anne Dutranois-Coumont, Steve Holdaway, Eric Stromboe, Antonio Nocioni, and many others
• Petrel plugin deployment team: Fred Xu, Sarah Vitel, Edmund Ing
• MathWorks
Motivation

• MATLAB’s efficient language structure, mathematical libraries, and flexible visualizations are powerful tools for rapidly designing algorithms

• Quality of algorithm designs depends on the variety of test datasets

• The easiest way to get feedback and testing from busy interpreters is to bring your algorithm conveniently to their environment

• MATLAB provides flexible capabilities to deploy algorithms directly into Petrel plugins for rapid iterative improvement and deployment.
Conventional Horizon Auto-tracking

‘The Parihaka seismic data shown is courtesy of the Government of New Zealand Petroleum and Minerals’
Simple/flexible visualization tools

- MATLAB’s visualization libraries allow for rapid prototyping and algorithm analysis
Algorithm Deployment Workflow

Typical Deployment Workflow

- Weeks to Months
- MATLAB Algorithm
- C++ (C#) Algorithm
- Petrel Plugin
- User Feedback

Iterative Deployment Workflow

- MATLAB Algorithm
- MATLAB .Net Library
- User Feedback
- Petrel Plugin
- Minutes

• User feedback starts when the algorithm is in Petrel
• Easier to iteratively improve to MATLAB implementation
Algorithm Deployment Workflow

- Adding MATLAB libraries to Petrel plugins is simple and well documented

Compiler SDK

MATLAB function

MATLAB or MATLAB Runtime Installation
Algorithm Enhancement and QC Workflow

• Several algorithm changes were enabled by this rapid prototyping process:
  – More flexible geometry of auto-correlation input
  – Corrections for seismic amplitude clipping / quantization
  – Include horizon-based dip in addition to precomputed seismic dip volumes
  – Numerous additional stopping conditions (e.g. horizon curvature)
• There are a variety of seismic dip estimators and representations.
• Typically seismic dip is QC’d by co-rendering dip with seismic amplitude
Dip QC

- Geoscientists requested that we incorporate our dip QC visualization direction into Petrel
Dip-guided Auto-tracking
Conclusions

• MATLAB’s efficient language structure, mathematical libraries, and flexible visualizations are powerful tools for rapidly designing algorithms

• Quality of algorithm designs depends on the variety of test datasets

• The easiest way to get feedback and testing from busy interpreters is to bring your algorithm conveniently to their environment

• MATLAB provides flexible capabilities to deploy algorithms directly into Petrel plugins for rapid iterative improvement and deployment.