Semiautomatic tracking of multiple horizons within a formation – PROGRAM horizon_tracking

Contents

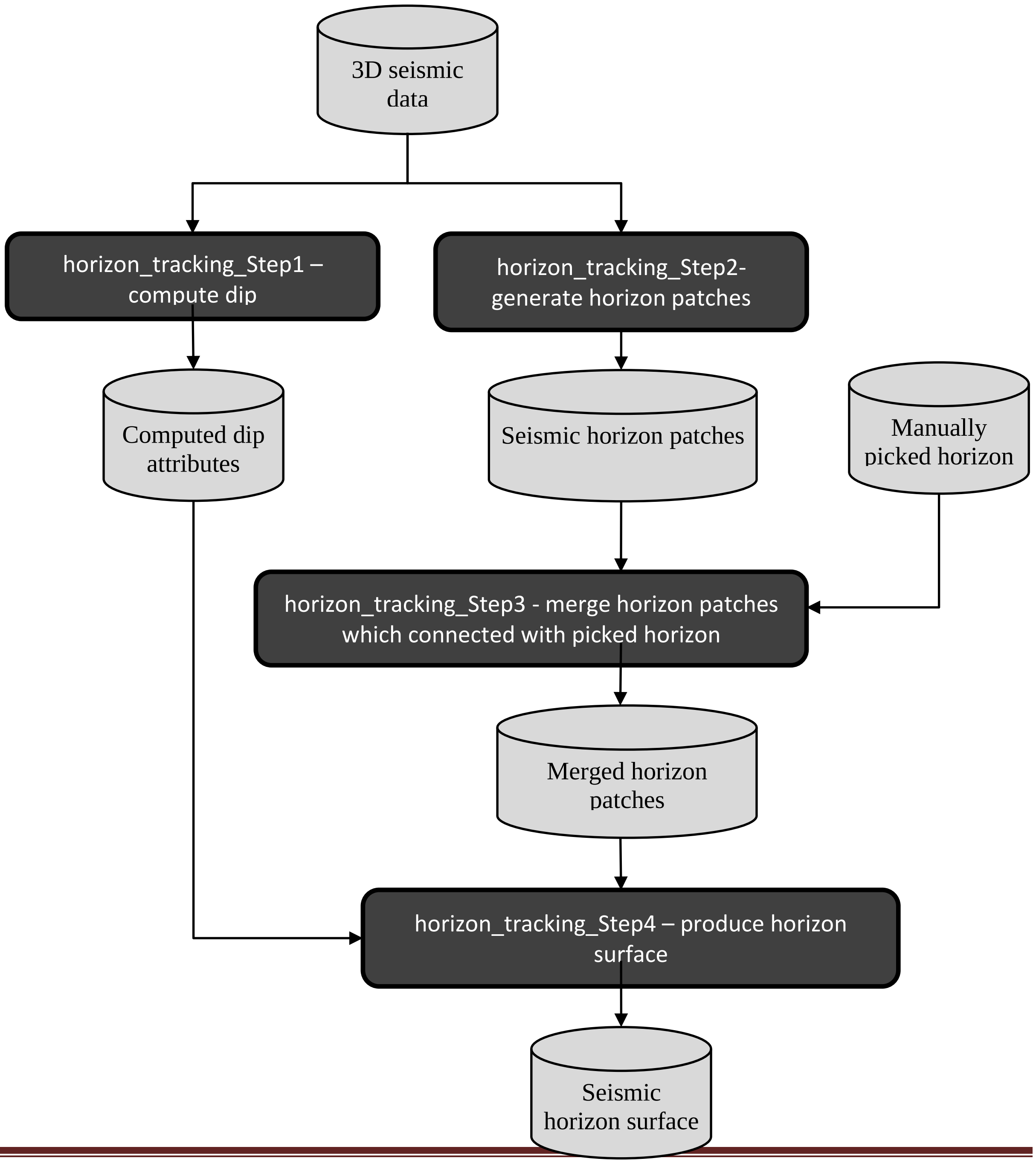
Program installation

Although several of the AASPI GUIs have limited interactive capabailities, most do not, while all the application algorithms are best described as being “batch” processes, where the Execute button starts a program that runs in the background until it finishes.

In contrast, program **horizon_tracking** is fully interactive. The authors have used the popular matlab software system as their development framework. Running the executables does not require a license to matlab, but it does require downloading and installing the matlab r2018a runtime package in the appropriate directory. Because program **horizon_tracking** is large, approaching 300 Mb, or approximately half the size of the AASPI package, we are currently storing it as a separate file. The runtime package is larger still, approaching 1500 Mb.

Computation Flow Chart

Program **horizon_tracking** reads in a seismic data volume as well as two picked horizons and generates a suite of intermediate horizons that can be used in a sequence stratigraphic interpretation framework.


3D seismic data
horizon_tracking_Step1 – compute dip
horizon_tracking_Step2- generate horizon patches
Computed dip attributes
Seismic horizon patches
Manually picked horizon
horizon_tracking_Step3 - merge horizon patches which connected with picked horizon
Merged horizon patches
horizon_tracking_Step4 – produce horizon surface
Seismic horizon surface

Output file naming convention

Program **horizon_tracking_step0** will always generate the following output files:

Output file description	**File name syntax**
3D seismic data volume	seisData_*unique_project_name_suffix*.mat

Program **horizon_tracking_step1** will always generate the following output files:

Output file description	**File name syntax**
Defined special dip attribute volume	dip_dtwPatch_2D_*unique_project_name_suffix*.mat

Program **horizon_tracking_step2** will always generate the following output files:

Output file description	**File name syntax**
Seismic horizon patches	Patch_from_Step2_*unique_project_name_suffix*.mat

Program **horizon_tracking_step3** will always generate the following output files:

Output file description	**File name syntax**
Merged horizon patches	hor_ske_*unique_project_name_suffix*.mat

Program **horizon_tracking_step4** will always generate the following output files:

Output file description	**File name syntax**
Seismic horizon surface	hor_surface_*unique_project_name_suffix*.txt

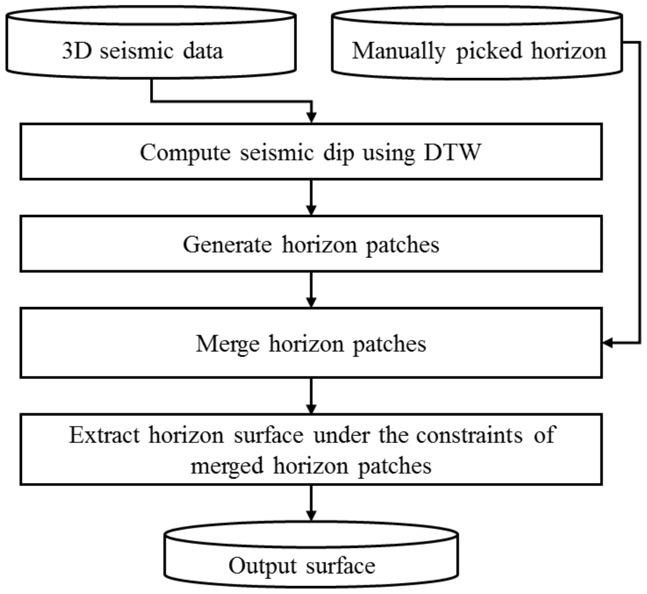
Overview

Manual seismic horizon picking is the least efficient interpretation technique in terms of time and effort. Accurate loop-ties are the key "element" and most time consuming task in manual horizon picking which ensures the accuracy of horizon picking. Auto-picking techniques were introduced in commercial software since the early 1980s. However, there are few studies regarding simulating the procedure of manual seismic horizon picking and quantitatively evaluating the auto-picked horizons. We propose to perform the auto-picking on inline and crossline seismic vertical slices independently, similar to the manual horizon picking procedure. We then evaluate the picked horizons using a loop-tie step similar to the loop-tie checking in manual horizon picking. To simulate the loop-tie step in manual picking, we define two dip attributes for each time sample of seismic traces: "left" and "right" reflector dips. We only preserve the portion of tracked horizon that meets the defined loop-tie checking. We next merge the tracked horizons centered at the seed seismic traces and the two-way travel time of merged horizons function as the "hard" controls for the final step of auto-picking. We finally use seismic dip attribute to track the horizons over the seismic survey under the hard controls.

We propose a new workflow to automatically simulate the procedure of manual seismic horizon picking. There are three main steps in our proposed workflow: (1) picking horizon patches centered at user-defined seed seismic traces, (2) merging horizon patches, and (3) automatically picking horizons over the whole seismic survey under the constrains of merged horizons. There are three main steps in generating horizon patches: (1) tracking the horizons along inline seismic slices, (2) tracking the horizons along crossline seismic slices, and (3) loop-tie checking the tracked results on inline and crossline slices and rejecting the tracked results which do not meet the defined loop-tie checking.

The reflector_tracking GUI

The Horizon tracking program consists of 5 steps (Figure below): (0) read seismic data (SEGY format), (1) calculate the seismic volumetric dip, (2) generate horizon patches, (3) merge horizon patches, and (4) generate horizon surfaces.
Our program needs interpreters at least tracking the desired horizon on one vertical slice (inline, crossline, or arbitrary line). We recommend the interpreters tracking horizon at least on one inline and crossline slices. Interpretation on both inline and crossline slices would heavily reduce computation time of generating horizon surfaces.

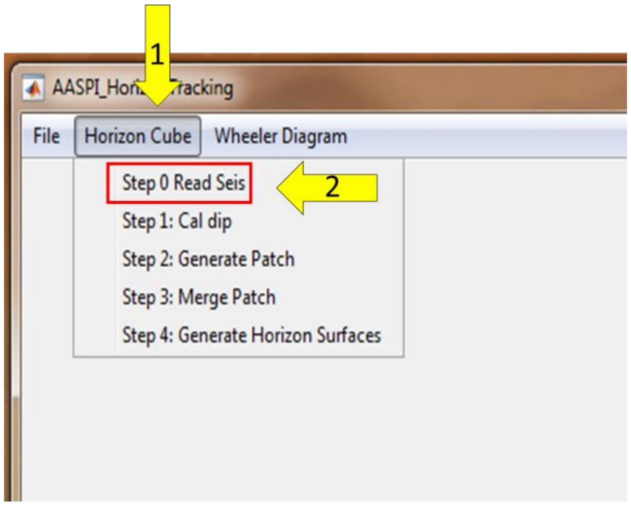

3D seismic data
Manually picked horizon
Compute seismic dip using DTW
Generate horizon patches
Merge horizon patches
Extract horizon surface under the constraints of merged horizon patches
Output surface

Our program is running under MatLab Runtime 9.5. The MatLab Runtime is free and can be download using the following link
https://www.mathworks.com/products/compiler/matlab-runtime.html

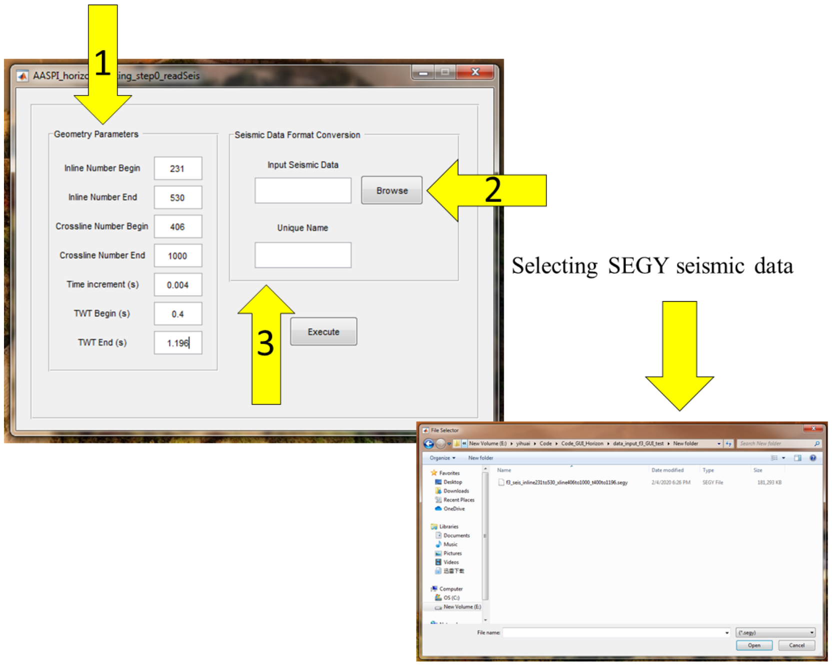
Step 0: input SEGY seismic data

This step would convert the seismic data in SEGY format to a MatLab array format. The seismic survey must have a rectangle shape. The program now can only handle seismic survey with a rectangle shape. The users need crop the shape of the seismic survey to be a rectangle if the seismic survey has a non-rectangle shape.

Click "Horizon Cube" (arrow 1) and select "step 0 Read Seis" (arrow 2).



Interpreters need input the geometry of the seismic survey (arrow 1). Interpreters need input the first and last inline number, first and last crossline number, the start and end two-way travel time, the time increment. Then click "browse" (arrow 2) to select the SEGY seismic data. Interpreters also need give a unique project name for the output seismic file (arrow 3)


1
2
3
AASPI_horizonTracking_step0_readSeis
Browse
Execute
Selecting SEGY seismic data
File Selector
Open
Cancel

Geometry Parameters

Inline Number Begin	231
Inline Number End	530
Crossline Number Begin	406
Crossline Number End	1000
Time increment (s)	0.004
TWT Begin (s)	0.4
TWT End (s)	1.196

Seismic Data Format Conversion

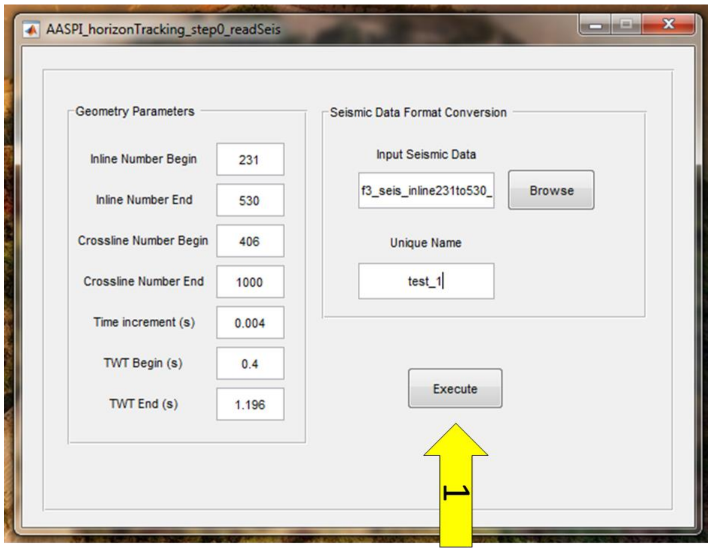
Input Seismic Data

Unique Name

New Volume (E:) ▸ yihuai ▸ Code ▸ Code_GUI_Horizon ▸ data_input_f3_GUI_test ▸ New folder

Name	Date modified	Type	Size
f3_seis_inline231to530_xline406to1000_t400to1196.segy	2/4/2020 6:26 PM	SEGY File	181,293 KB

The following figure show the interface after we defined all the parameters. Then we click execute (arrow 1) to convert the seismic data to MatLab array data. Then we finished the seismic SEGY data input.



Geometry Parameters

Inline Number Begin	231
Inline Number End	530
Crossline Number Begin	406
Crossline Number End	1000
Time increment (s)	0.004
TWT Begin (s)	0.4
TWT End (s)	1.196

Seismic Data Format Conversion

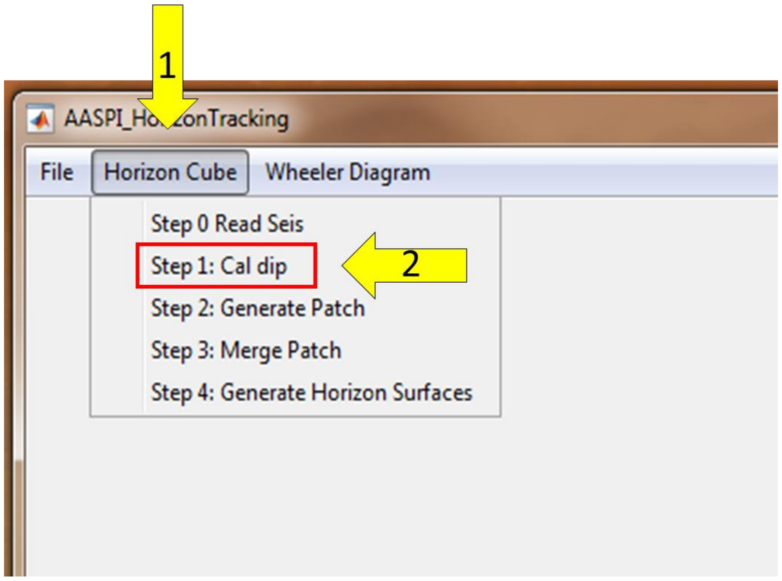
Input Seismic Data: f3_seis_inline231to530_

Unique Name: test_1

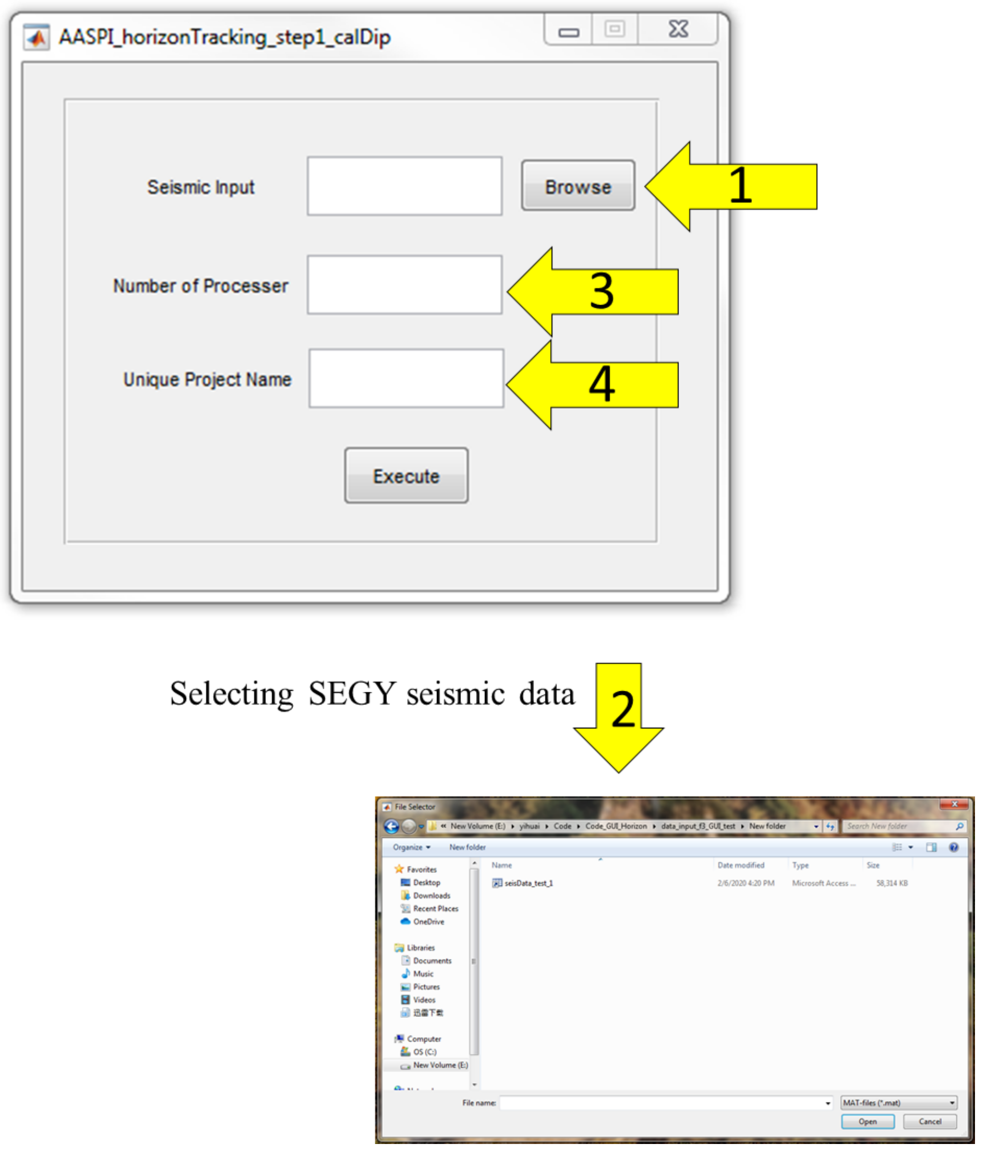
Step 1: Calculate the seismic dip

This step is designed to compute the seismic inline and crossline dip. Our dip computation algorithm compute dip attribute using dynamic time warping and this step is the most time consuming step and it may take hours and up to days.

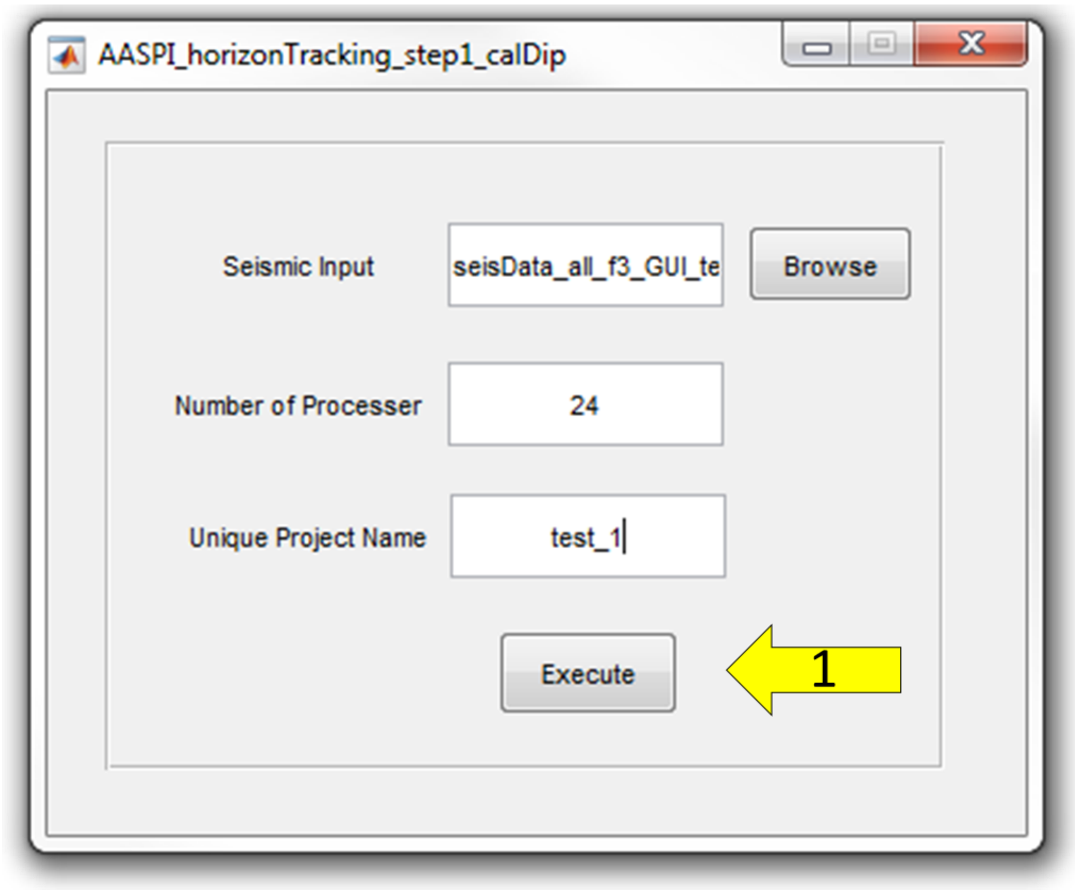
The dip computation interface will pop up after click "Horizon Cube" (arrow 1) and select "step 1 Cal dip" (arrow 2).



Interpreters need inputting the converted seismic data from the "step 0". Click "browse" (arrow 1) to select the converted seismic data (arrow 2). Interpreters need define the number of processer (arrow 3). The number of processer will be automatically set as the maximum number if the input processor number is greater than the actual processor. Interpreters also need to give a unique project name for the computed dip (arrow 4).



The following interface shows the interface after we defined all the parameters. Finally users can compute the dip attribute by clicking "execute".


AASPI_horizonTracking_step1_calDip

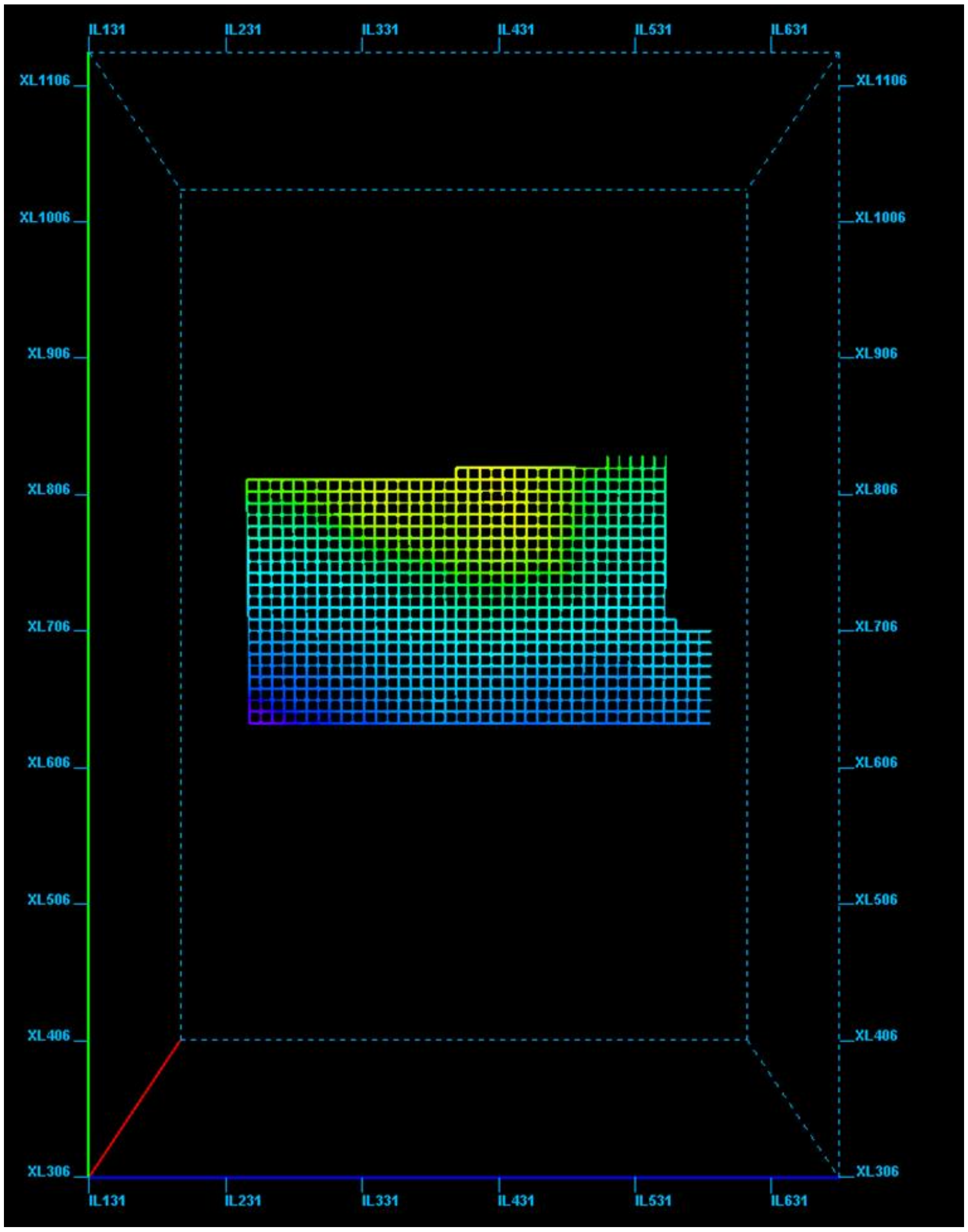
Seismic Input: seisData_all_f3_GUI_te

Number of Processer: 24

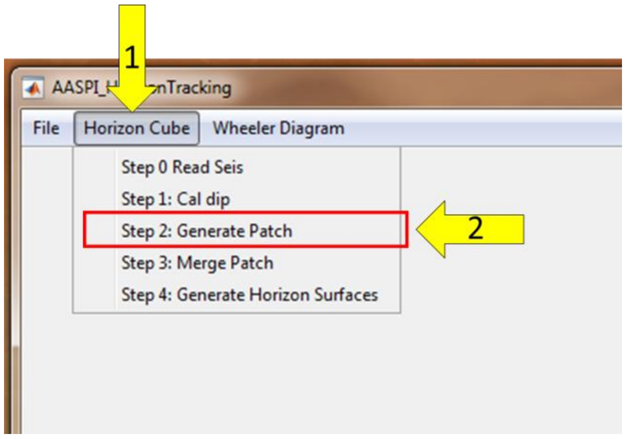
Unique Project Name: test_1

Step 2: Generate horizon patches

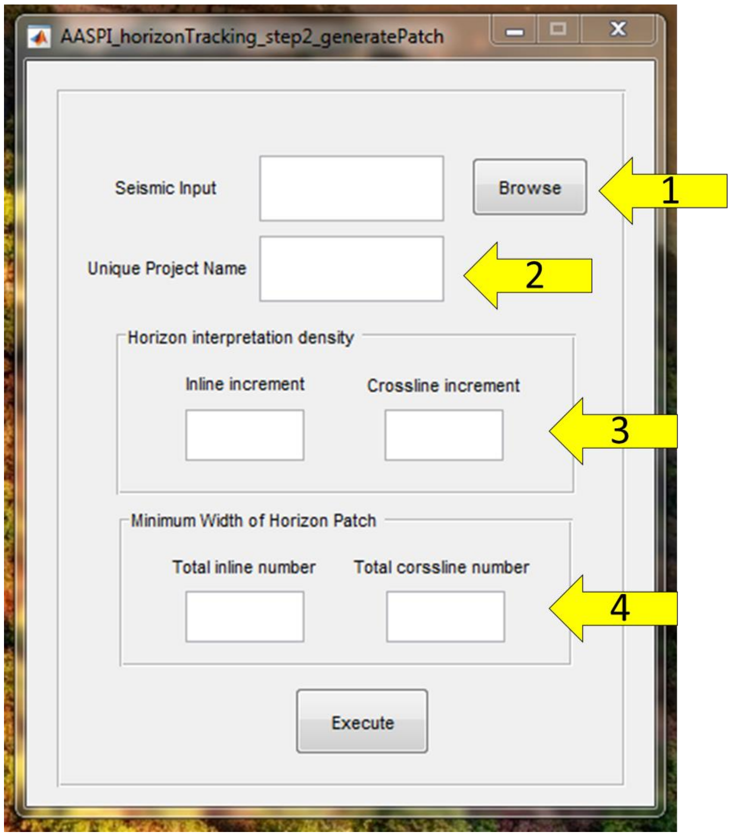
The horizon patch generation step will generate horizon patches. The horizon patches are defined “small” pieces of automatic tracked horizon. The follow figure indicate one tracked horizon patch. The size of the horizon patch is determined by number of line *times* number of crossline. The horizon patches should strictly follow the local seismic reflector and meet the “loop tie” criteria used in manual horizon interpretation



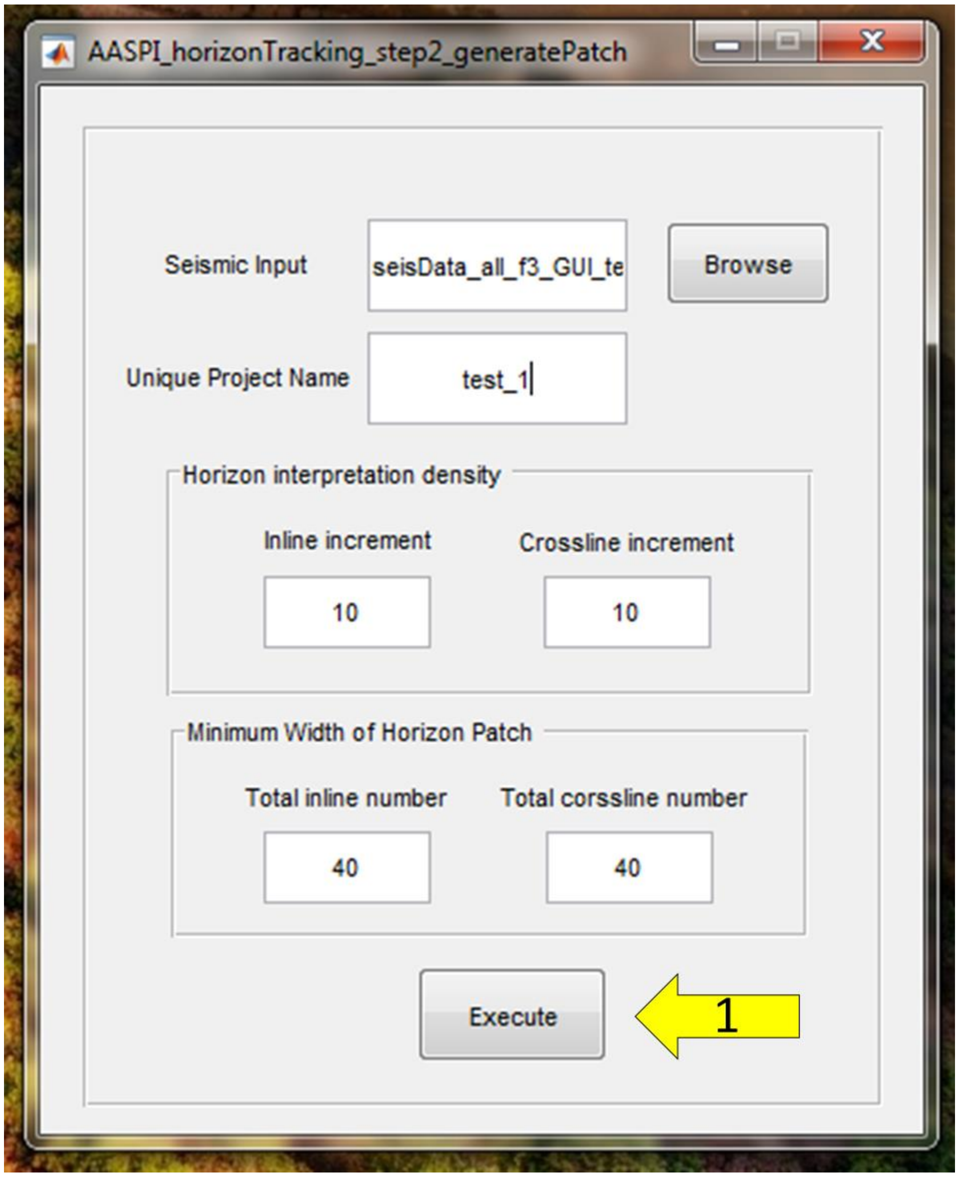
The horizon patch generation interface pops up by clicking “Horizon Cube” (arrow 1) and select “step 2 Generate Patch” (arrow 2).



Interpreters need inputting the converted seismic data in the “step 0”. Click “browse” (arrow 1) to select the converted seismic data, and give a unique project name for the generated horizon patches (arrow 2).
Interpreters need define the horizon interpretation density along inline and crossline (arrow 3). Interpreters also need defining the minimum width of horizon patches along inline and crossline (arrow 4). The program only output horizon patches whose size are larger than the minimum size of horizon patch.



The following figure shows the interface after we defined all the parameters. Then we click "execute" (arrow 1) to generate horizon patches.



Seismic Input: seisData_all_f3_GUI_te

Unique Project Name: test_1

Horizon interpretation density

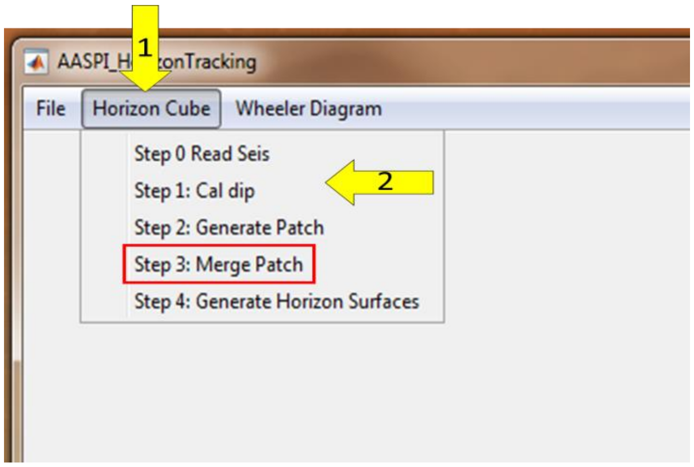
Inline increment	Crossline increment
10	10

Minimum Width of Horizon Patch

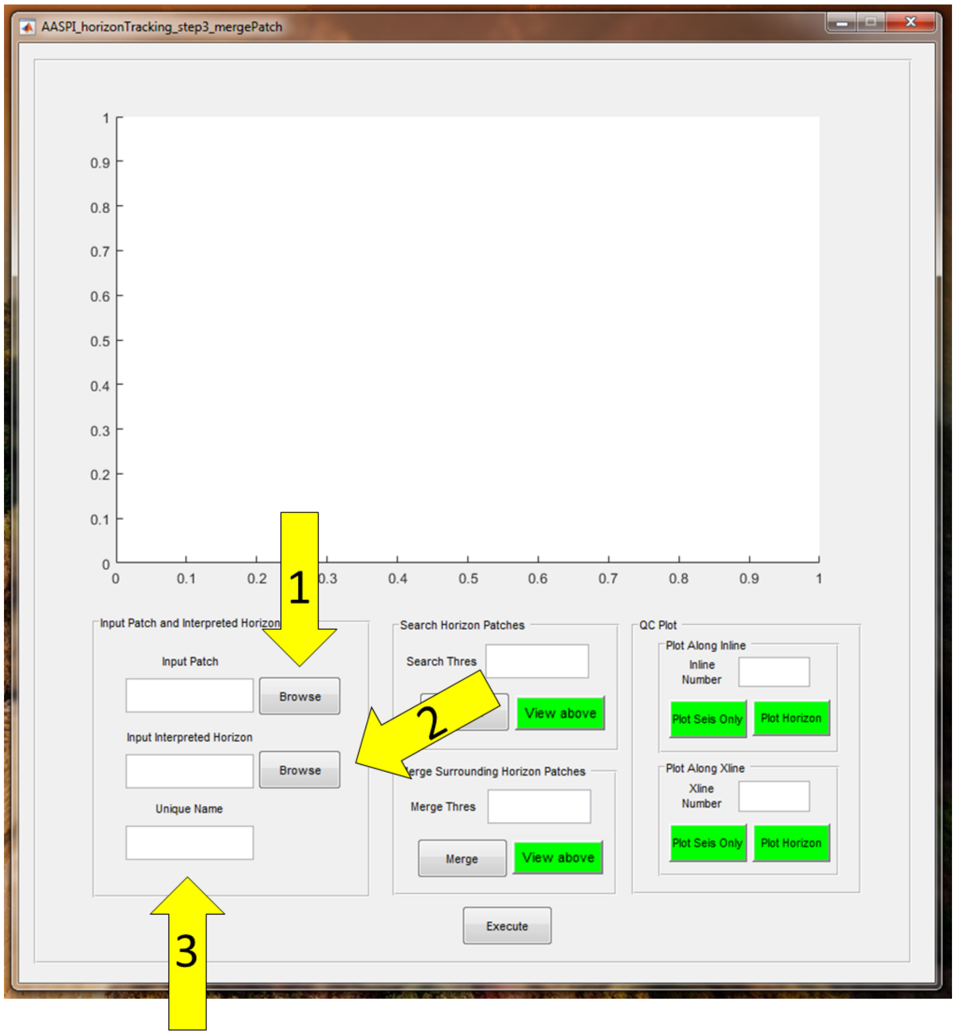
Total inline number	Total corssline number
40	40

Step 3: Merge horizon patches

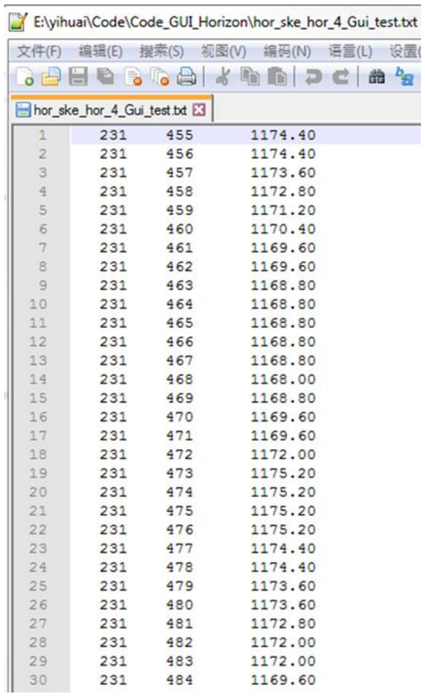
This step is designed to first merge the small pieces of horizon patch, which has interconnect with manual interpreted horizon. The algorithm then will iteratively merge horizon patches which are interconnect with merged horizon patches. We launch the merging algorithm by clicking "Horizon Cube" (arrow 1) and select "step 3 Merge Patch" (arrow 2).



Interpreters need input the calculated horizon patches in the "step 2". Click "browse" (arrow 1) to select the horizon patches, and click "browse" (arrow 2) to input the interpreted horizon. Interpreters then give a unique project name for the generated horizon (arrow 3).



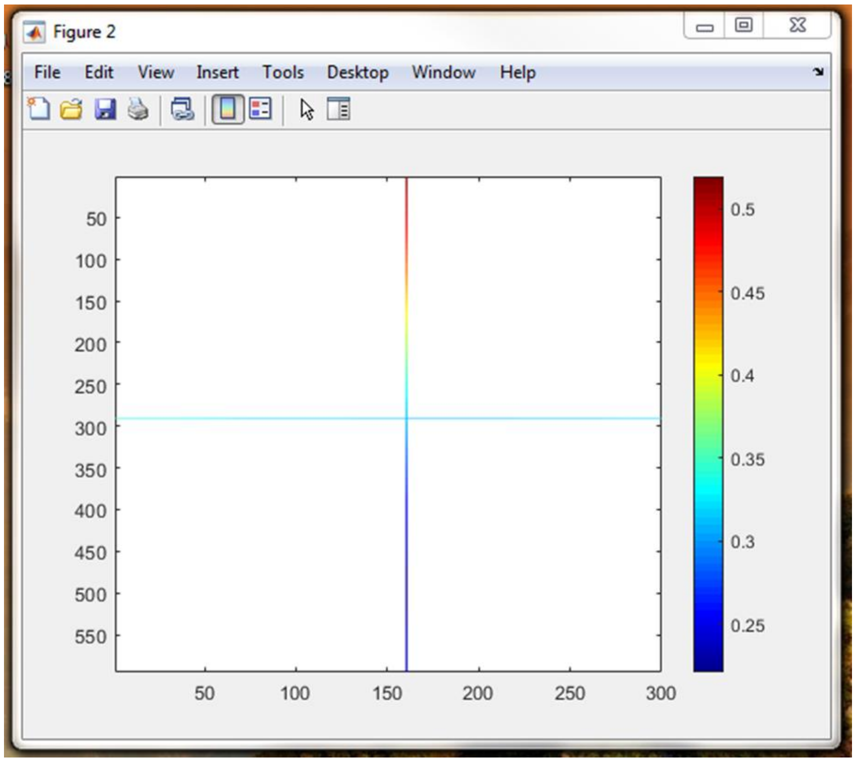
The horizon should be in ASCII format. The following figure is a demo. The first, second, and third column are inline number, crossline number, and time (unit is ms), respectively.


E:\yihuai\Code\Code_GUI_Horizon\hor_ske_hor_4_Gui_test.txt

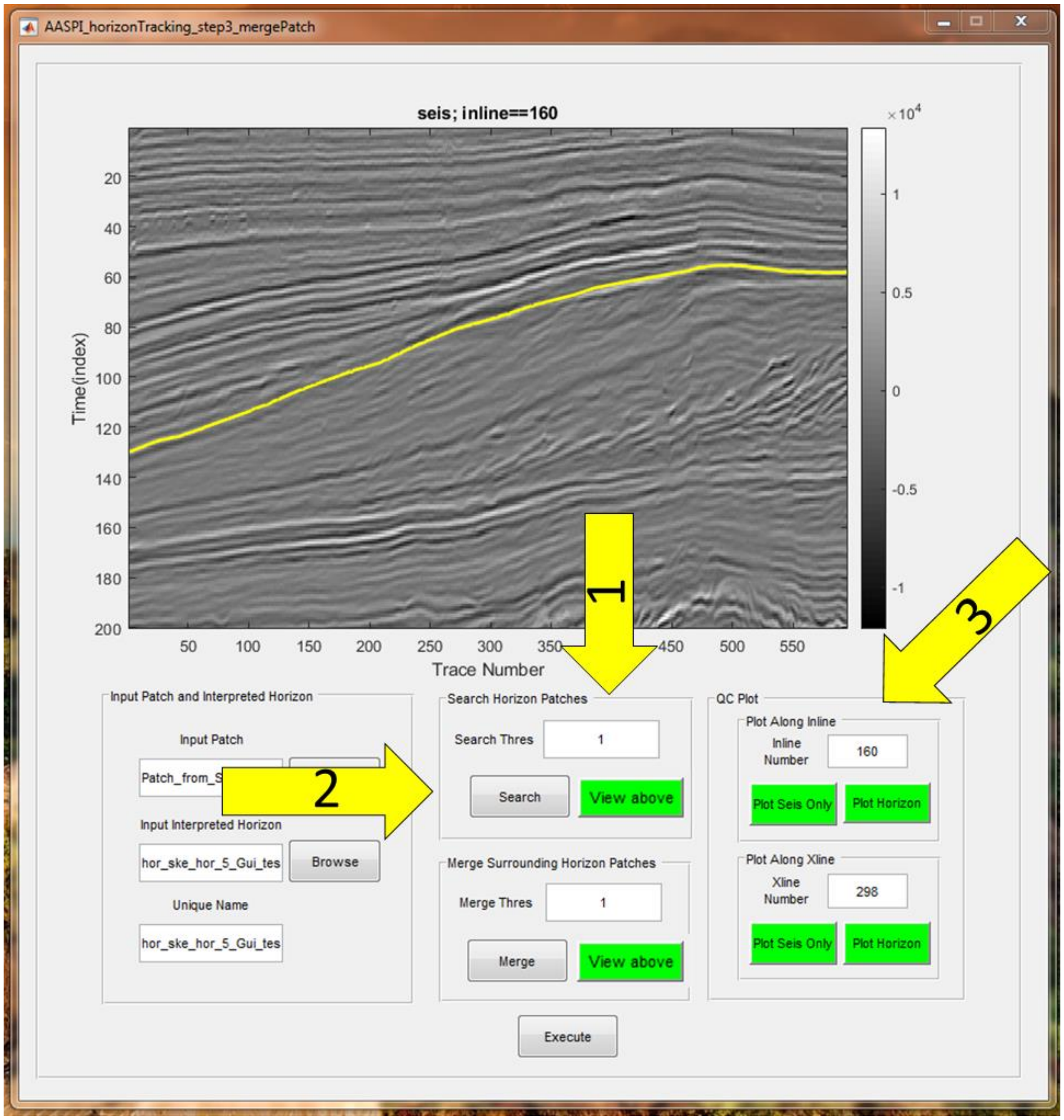
hor_ske_hor_4_Gui_test.txt

```
231    455     1174.40
231    456     1174.40
231    457     1173.60
231    458     1172.80
231    459     1171.20
231    460     1170.40
231    461     1169.60
231    462     1169.60
231    463     1168.80
231    464     1168.80
231    465     1168.80
231    466     1168.80
231    467     1168.80
231    468     1168.00
231    469     1168.80
231    470     1169.60
231    471     1169.60
231    472     1172.00
231    473     1175.20
231    474     1175.20
231    475     1175.20
231    476     1175.20
231    477     1174.40
231    478     1174.40
231    479     1173.60
231    480     1173.60
231    481     1172.80
231    482     1172.00
231    483     1172.00
231    484     1169.60
```

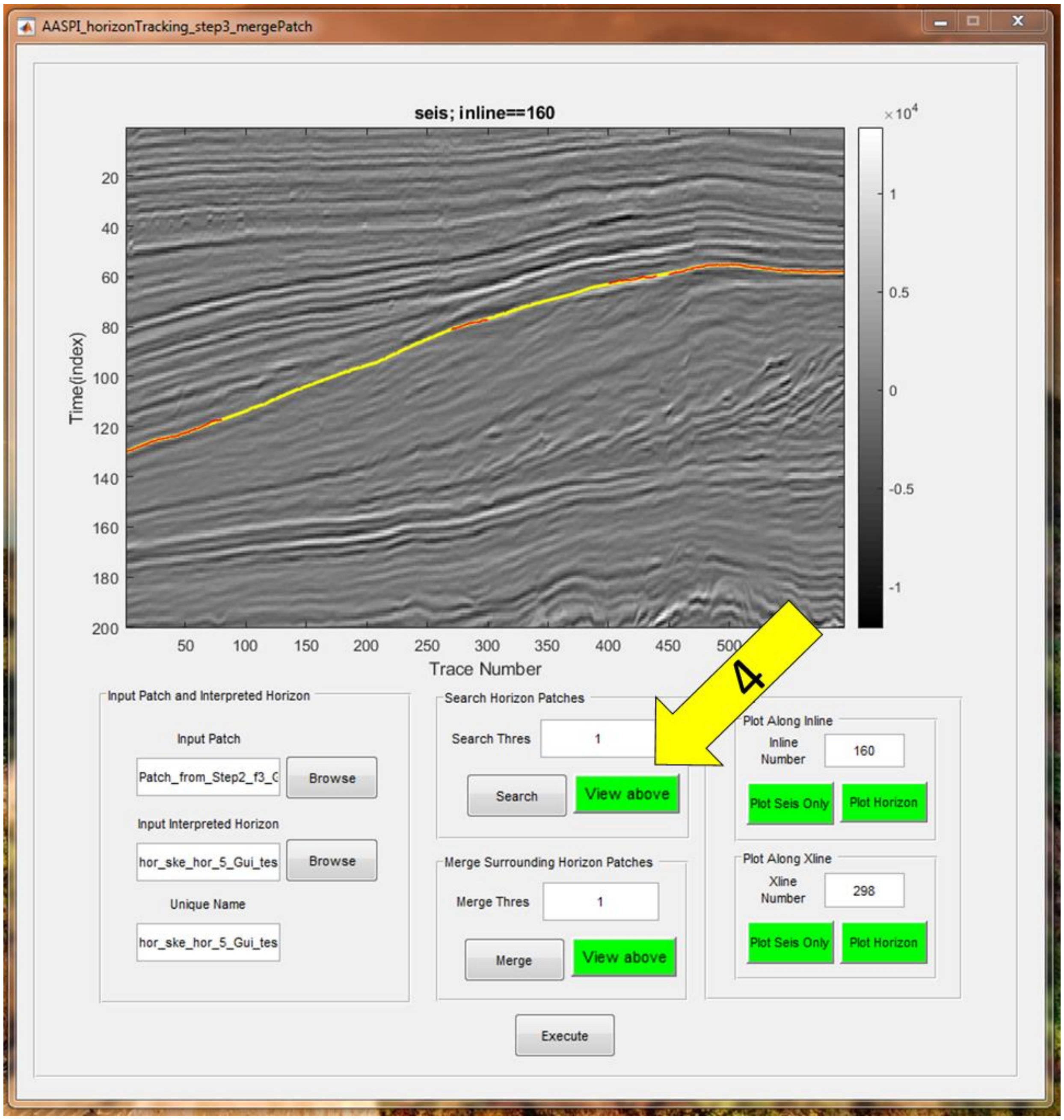
The map view of input manually interpreted horizon.



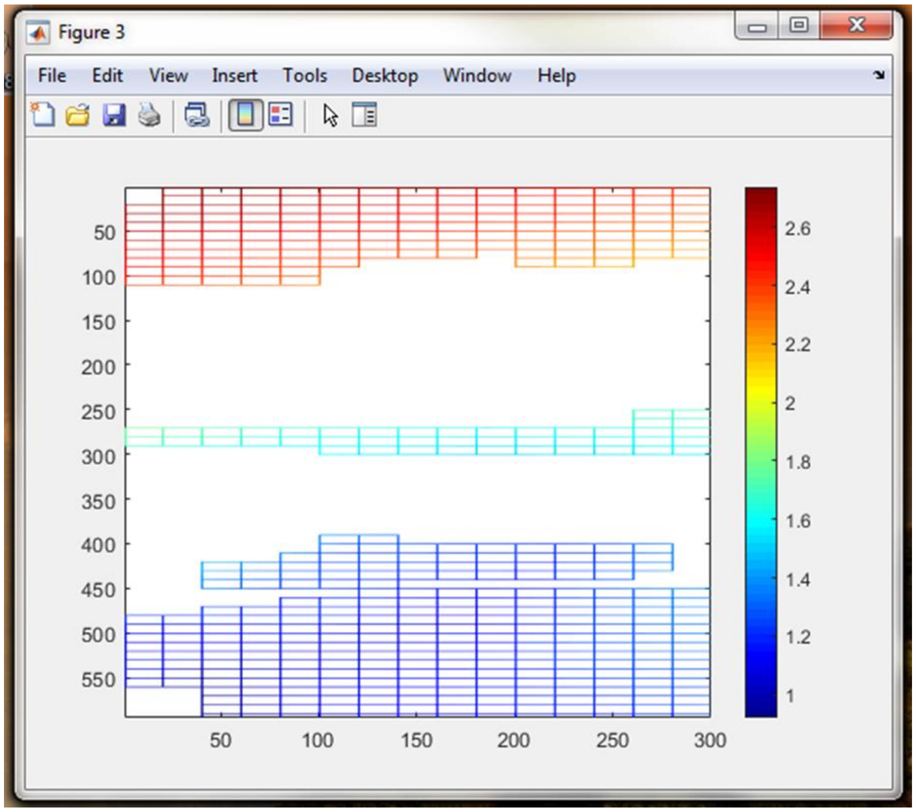
Interpreters need define the parameter (arrow 2) searching the horizon patches which are interconnected with manual interpreted horizons. The value of “1” means the program will search all the horizon patches if the closest vertical distance between the horizon patch and manual interpreted horizon is smaller than 1 sample. Users obtain all the possible horizon patches by clicking “search” (arrow 2). The parameter for searching is an interactive trial procedure. Users can QC the researched results by QC plot (arrow 3). The yellow curve is the manual interpreted horizon.



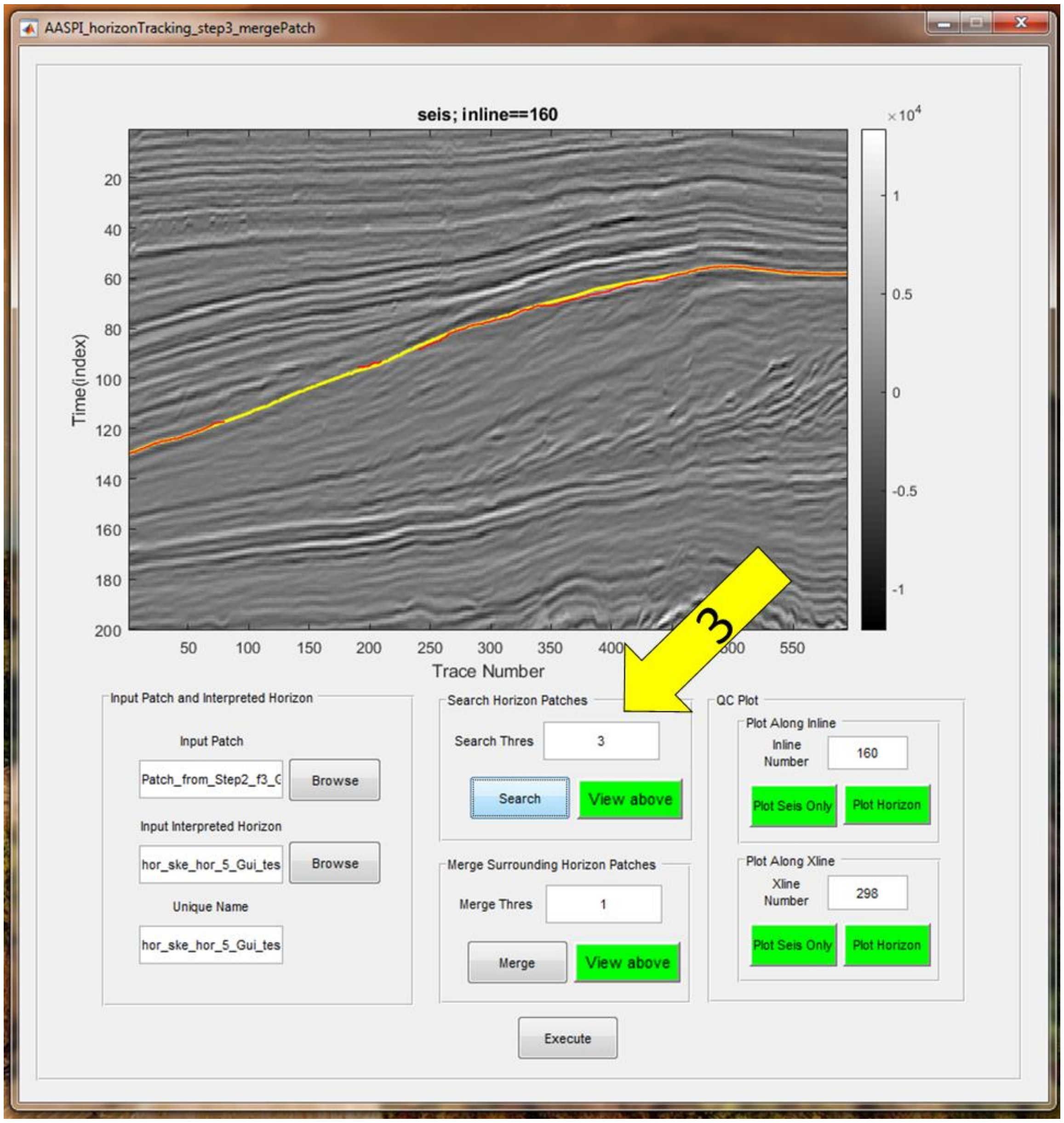
The following figure show the QC plot after the searching. The parameter for search is 1. The red curve is the searched horizon patches. The yellow curve is the manual interpreted horizon on inline slice 160. Interpreters can view how many horizon patches does the algorithm search by clicking "view above" (arrow 4).



The bird view of the searched horizon patches.

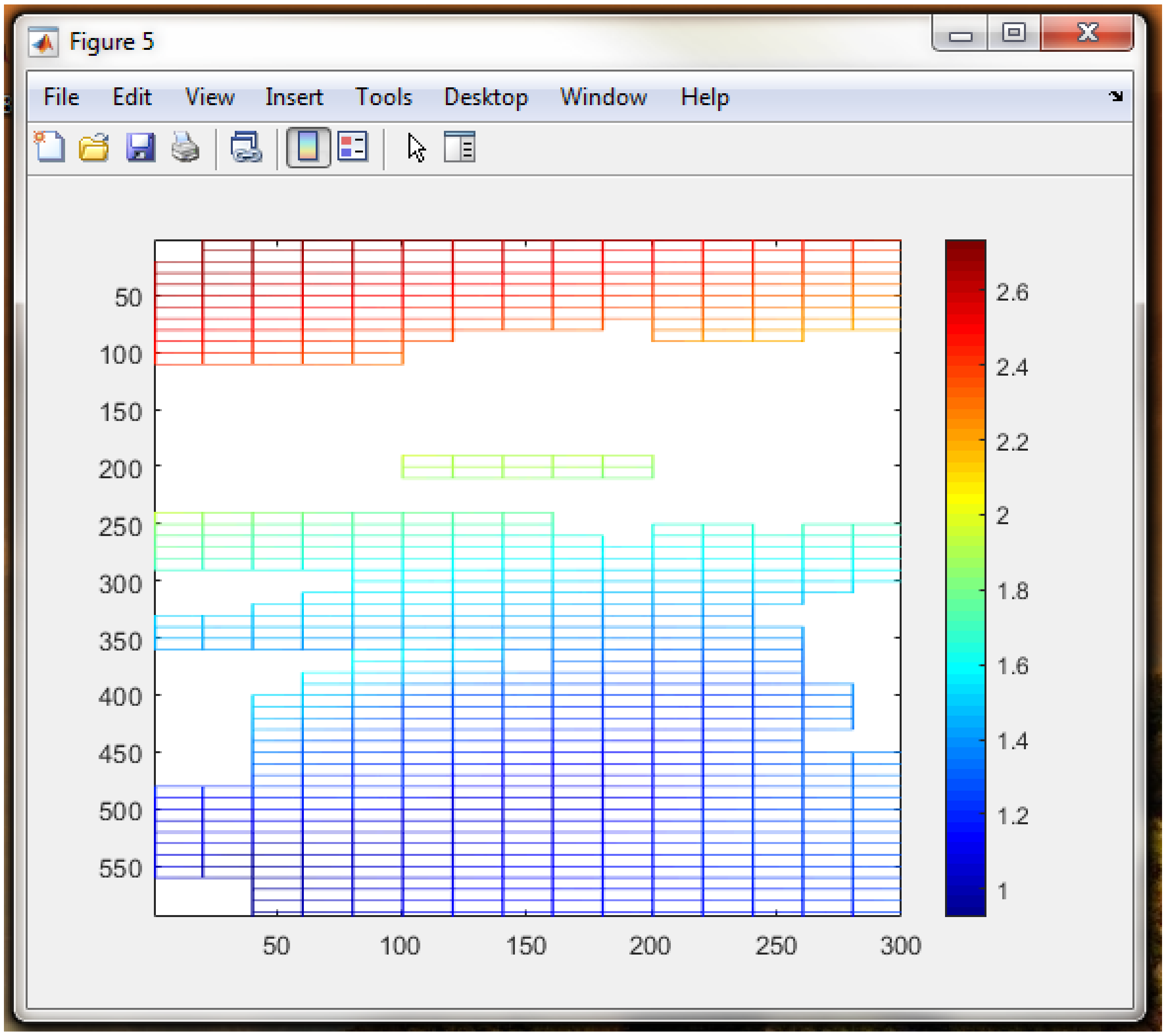


The following figure show the QC plot after the searching. The parameter for search is 3.

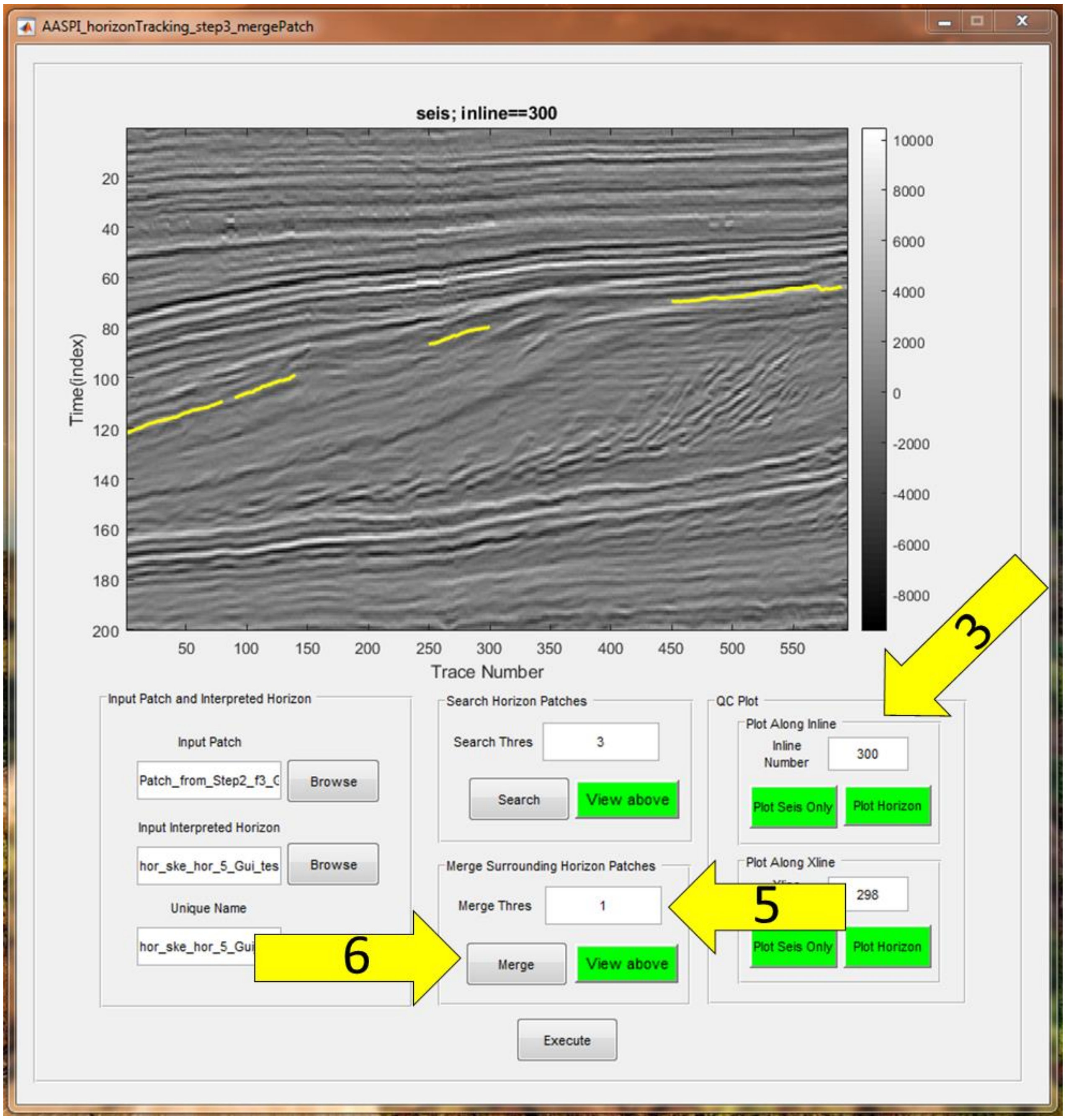


Input Patch: Patch_from_Step2_f3_C
Input Interpreted Horizon: hor_ske_hor_5_Gui_tes
Unique Name: hor_ske_hor_5_Gui_tes
Search Thres: 3
Merge Thres: 1
Inline Number: 160
Xline Number: 298

The map view of the searched horizon patches.



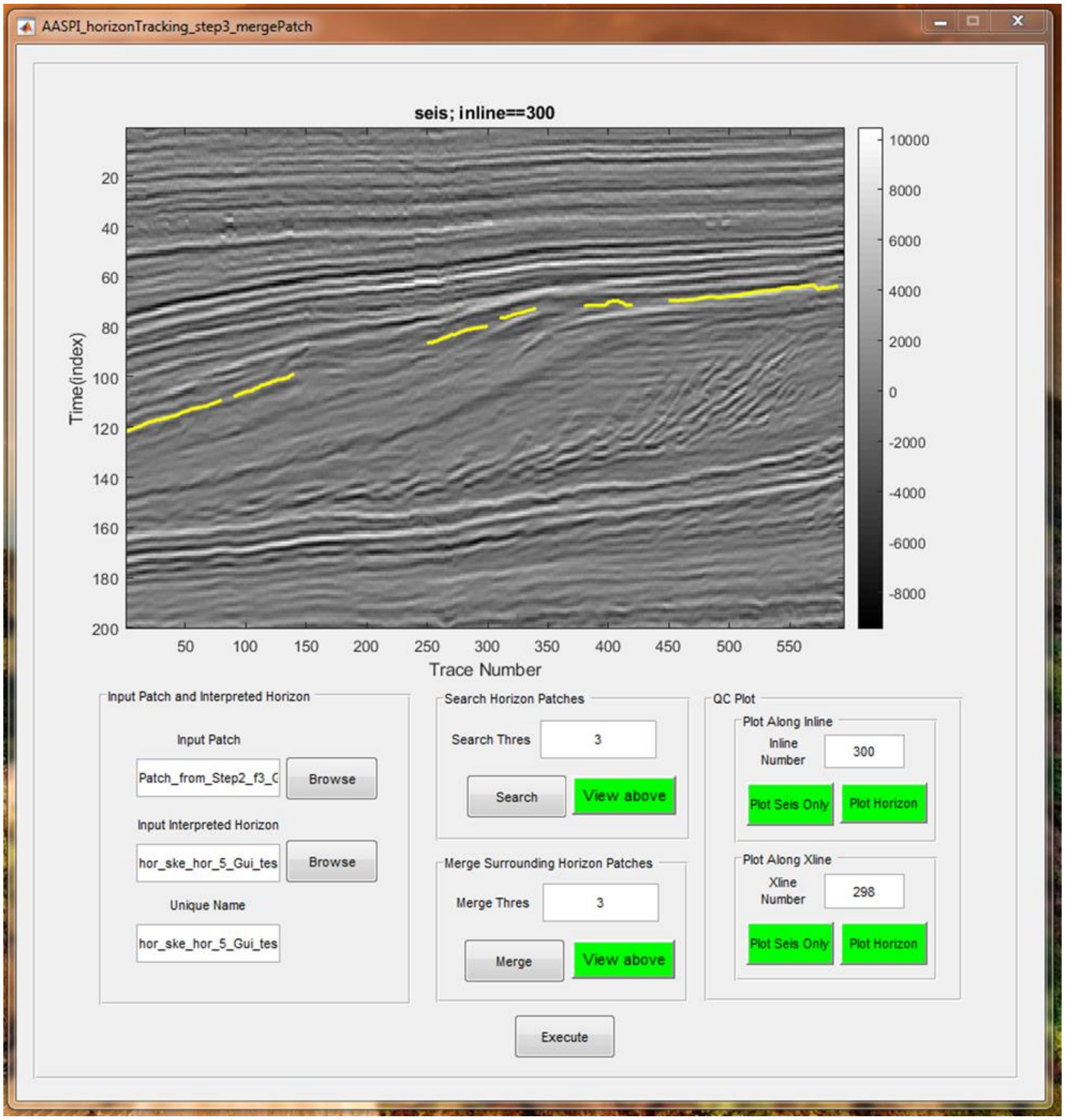
Step 3: Interpreters need define the parameter (arrow 5) merging the searched horizon patches The value of "3" means the program will search all the horizon patches if the closest vertical distance between the horizon patches is smaller than 3 samples. Users merge all the possible horizon patches by clicking "mere" (arrow 6).). The parameter for merging is an interactive trial procedure. Users can QC the merged results by QC plot (arrow 3)



Input Patch: Patch_from_Step2_f3_C
Input Interpreted Horizon: hor_ske_hor_5_Gui_tes
Unique Name: hor_ske_hor_5_Gui
Search Thres: 3
Merge Thres: 1
Inline Number: 300
Xline: 298

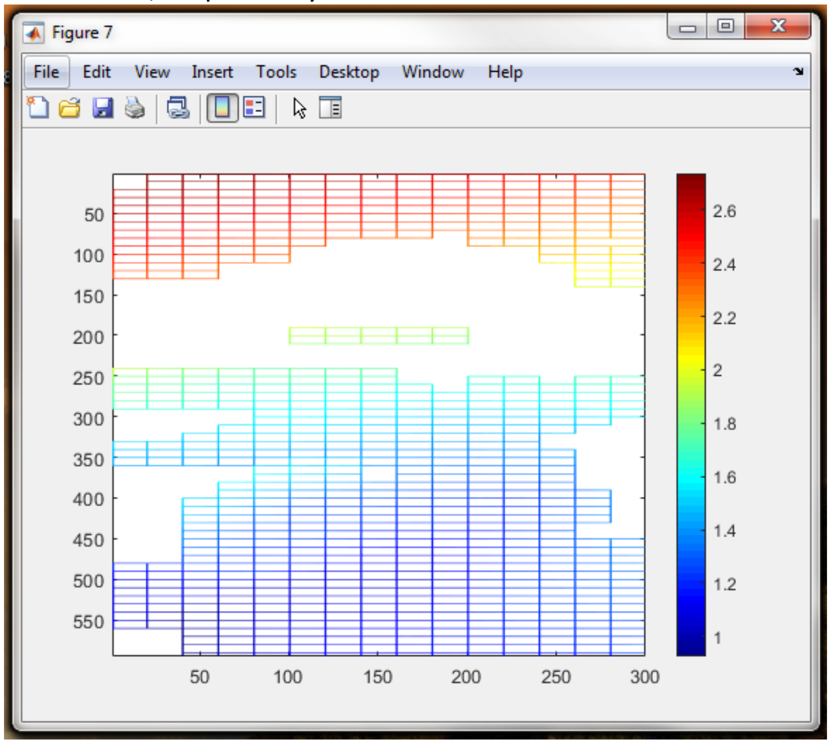
)

The following figure show the QC plot after the merging. The parameter for search is 3.

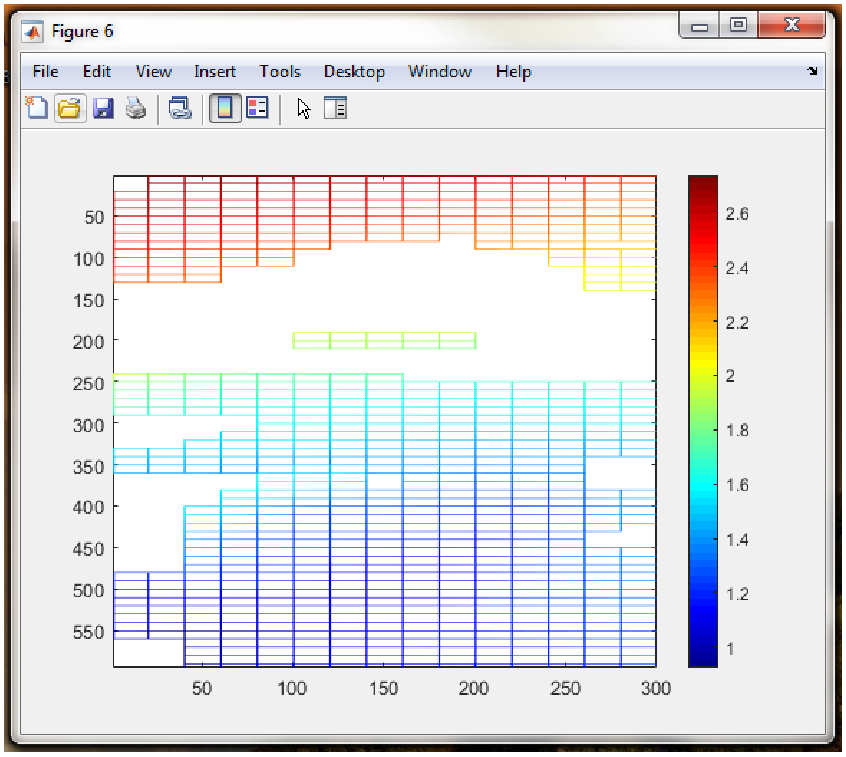


Input Patch: Patch_from_Step2_f3_C
Input Interpreted Horizon: hor_ske_hor_5_Gui_tes
Unique Name: hor_ske_hor_5_Gui_tes
Search Thres: 3
Merge Thres: 3
Inline Number: 300
Xline Number: 298

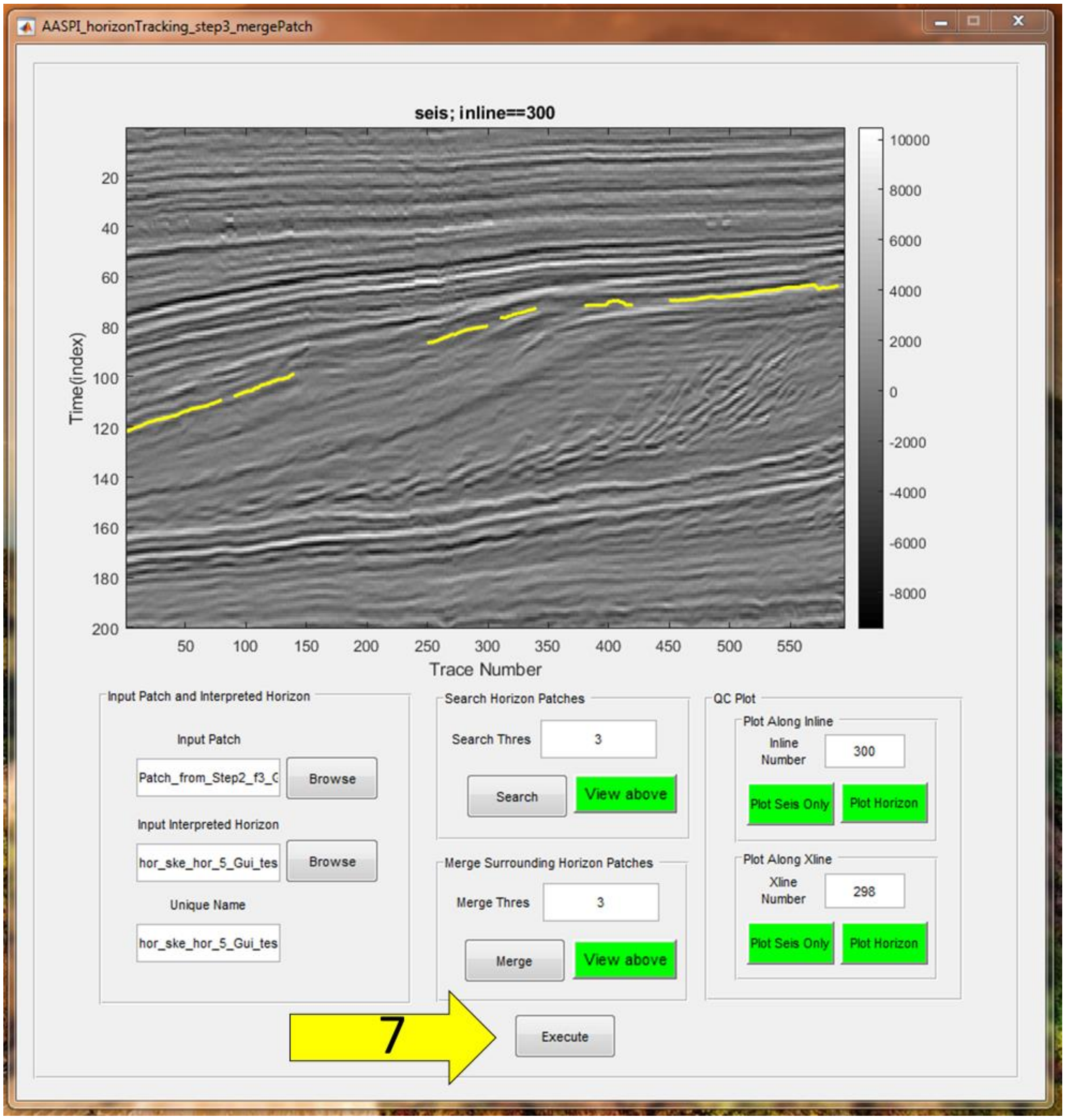
The following figure show the QC plot after the merging. The parameters for search and merging are 3 and 1, respectively.



The following figure show the QC plot after the merging. The parameters for search and merging are 3 and 3, respectively.



Interpreters finally need clicking “execute” (arrow 7) to output the merged horizon patches after we set the parameters.



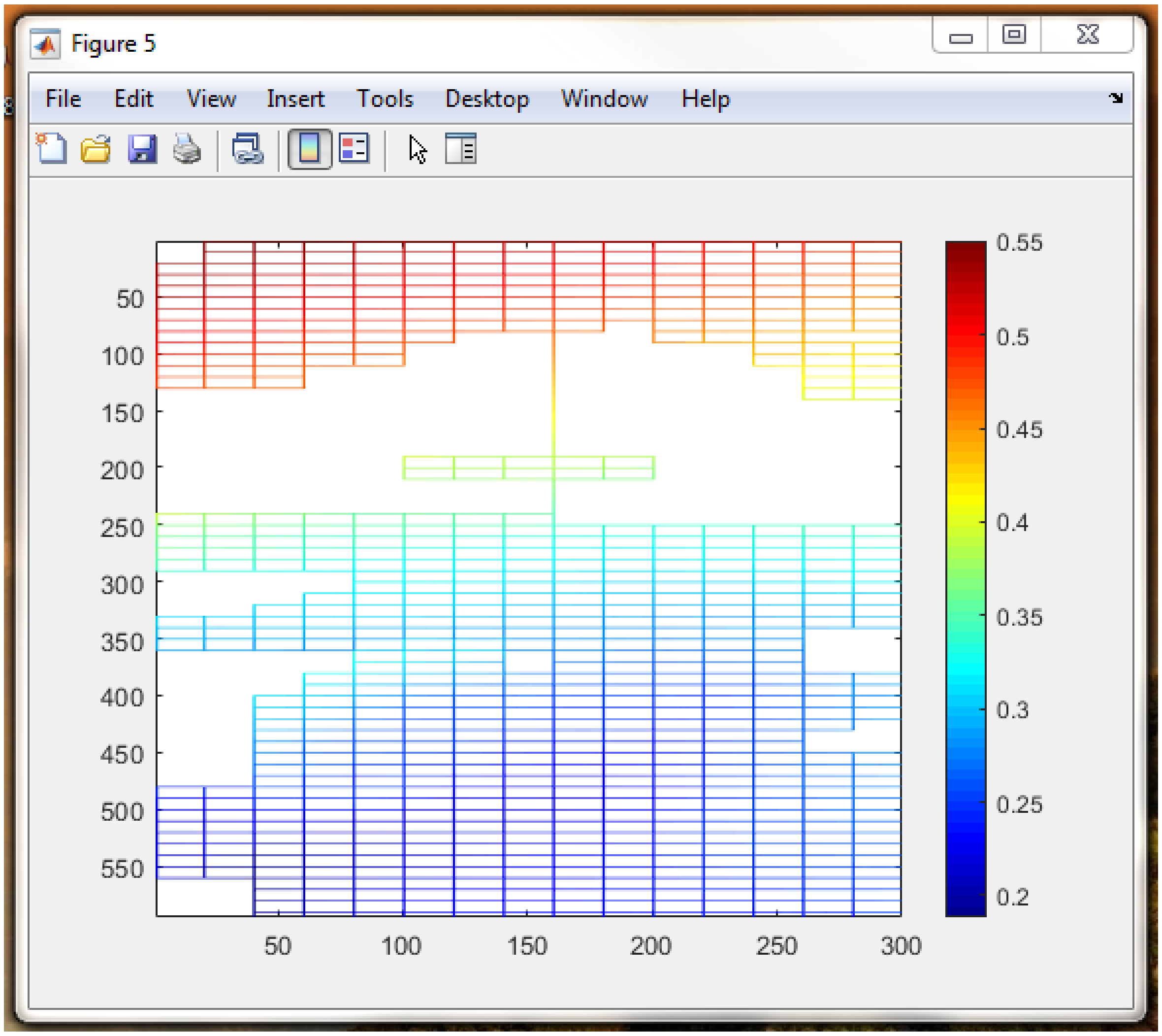
Input Patch and Interpreted Horizon
Input Patch: Patch_from_Step2_f3_C
Input Interpreted Horizon: hor_ske_hor_5_Gui_tes
Unique Name: hor_ske_hor_5_Gui_tes

Search Horizon Patches
Search Thres: 3

Merge Surrounding Horizon Patches
Merge Thres: 3

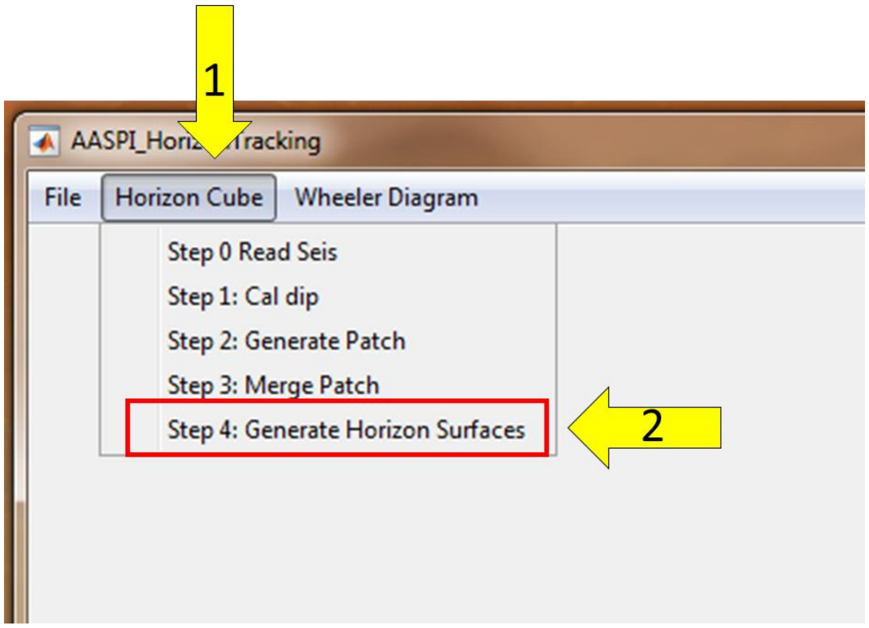
QC Plot
Plot Along Inline
Inline Number: 300
Plot Along Xline
Xline Number: 298

The merged horizon patches. The merged horizon patches have an interpreted density of 20 inlines by 10 inlines (defined in "Step 2)

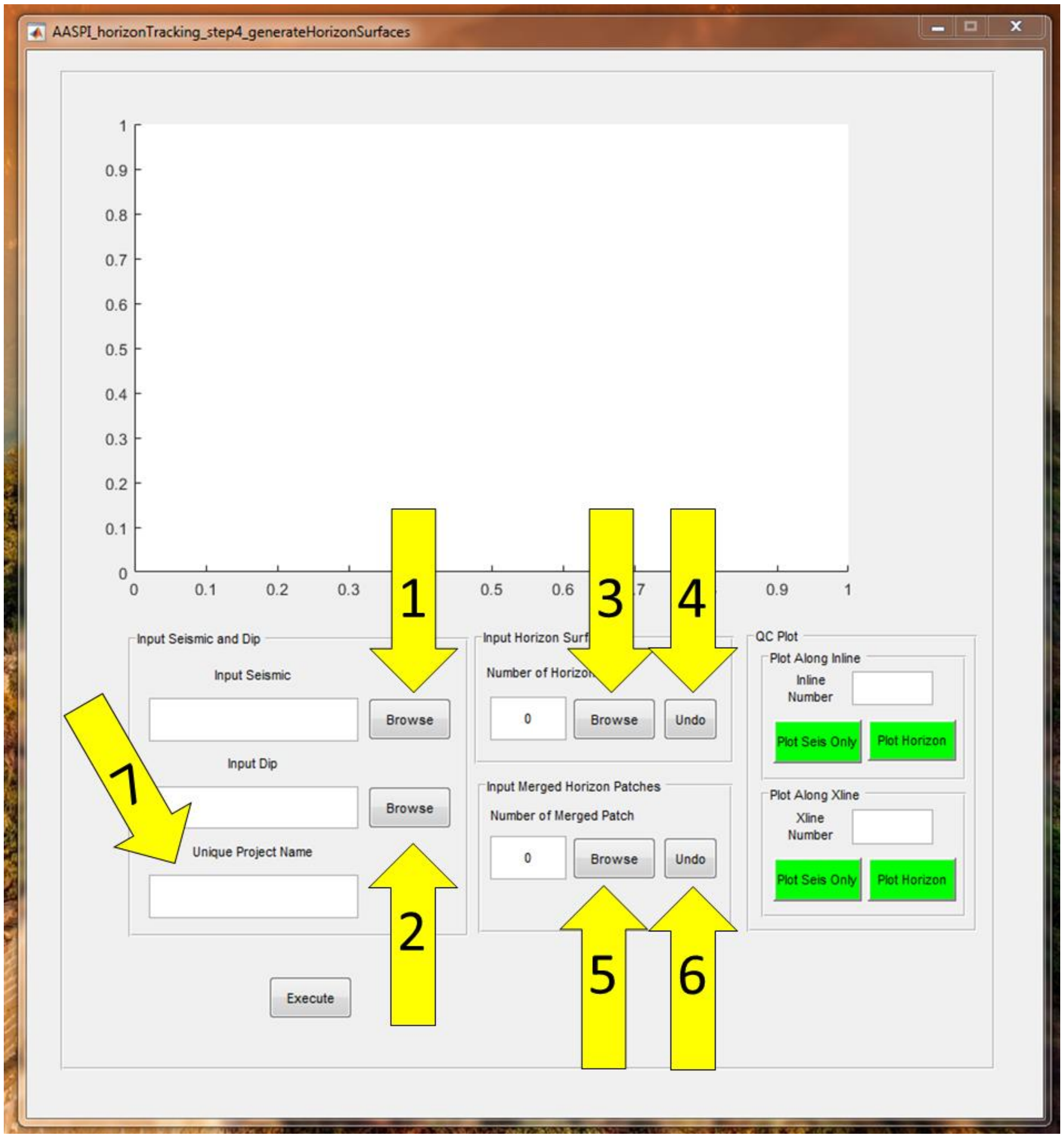


Step 4: Generate horizon surfaces

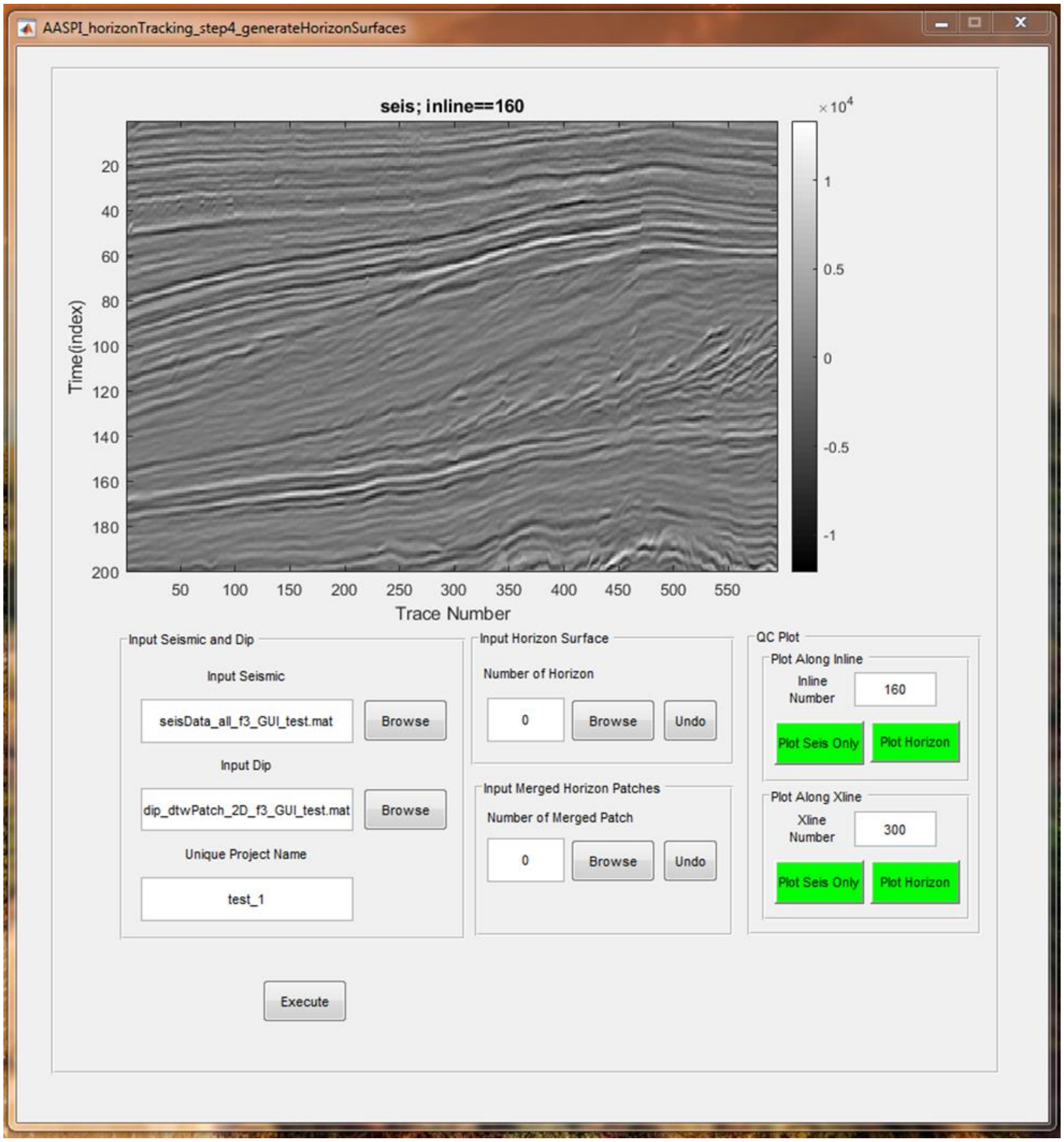
The final step is generating horizon surface (interpretation density is 1x1). The merged horizon patches in step 3 function as the hard constraints. We launch the interface of horizon surface generation by clicking “Horizon Cube” (arrow 1) and select “step 4 Generate horizon surfaces” (arrow 2).



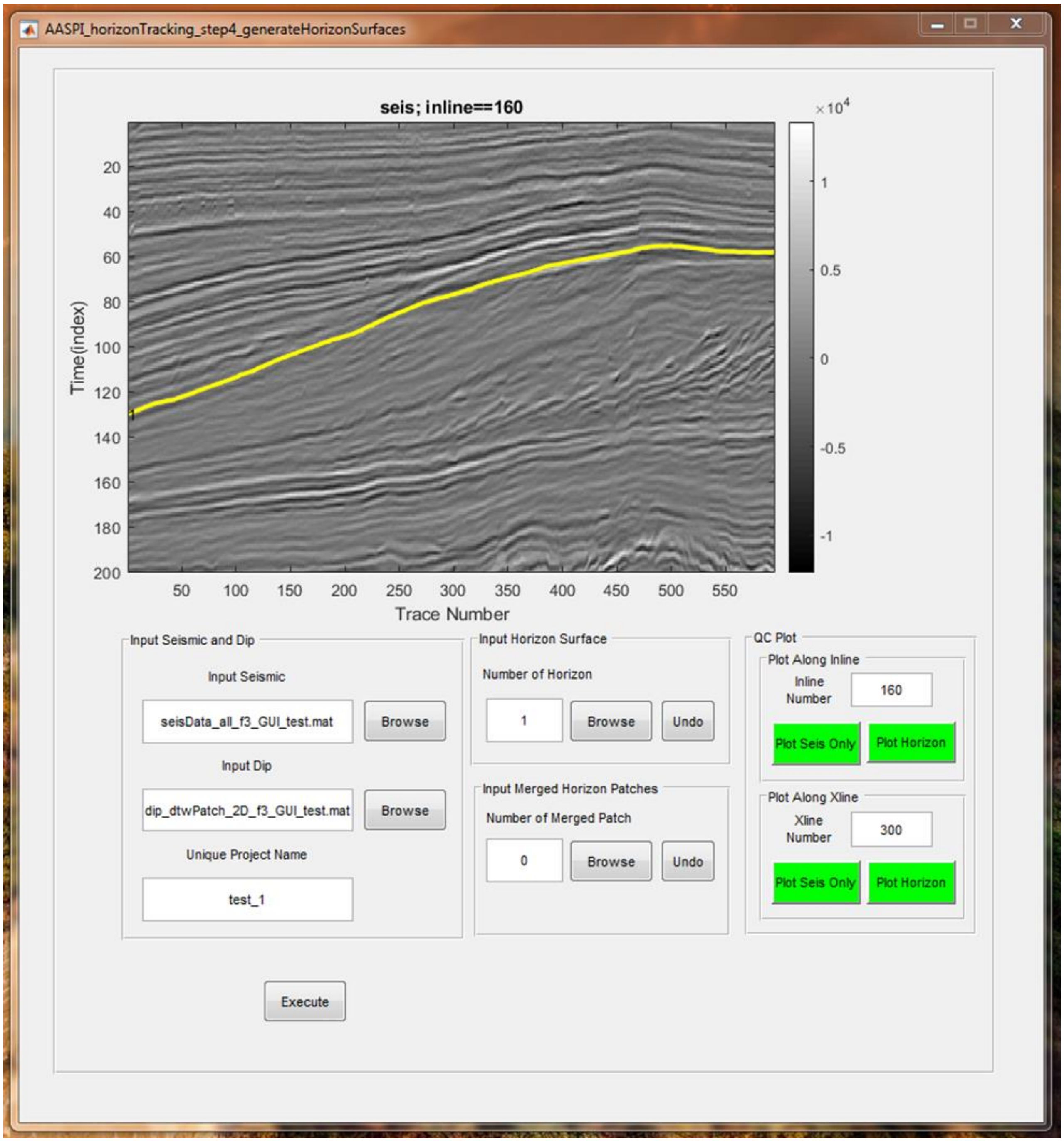
Interpreters need inputting the converted seismic data in “step 0” by clicking “browse” (arrow 1), and input seismic reflector dip in “step 1” by clicking “browse” (arrow 2). Interpreters can input several interpreted horizon surfaces (***optional***) by repeating clicking the “browse” (arrow 3) (interpreted density is **1x1**). Note that the interpreted horizons function as the constraints for current surface generating. Interpreters can click “undo” (arrow 4) to delete last input horizon. Interpreters then click “browse” (arrow 5) to select the merged horizons in “step 3”. Interpreters can click “undo” (arrow 6) to delete the merged horizon patch. Interpreters then give a unique project name for the generated horizon surface (arrow 7).



Interface after the input of seismic data and dip calculated in “step 1”.



Interface after the input of seismic data, dip, and the first horizon.



Input Seismic: seisData_all_f3_GUI_test.mat

Input Dip: dip_dtwPatch_2D_f3_GUI_test.mat

Unique Project Name: test_1

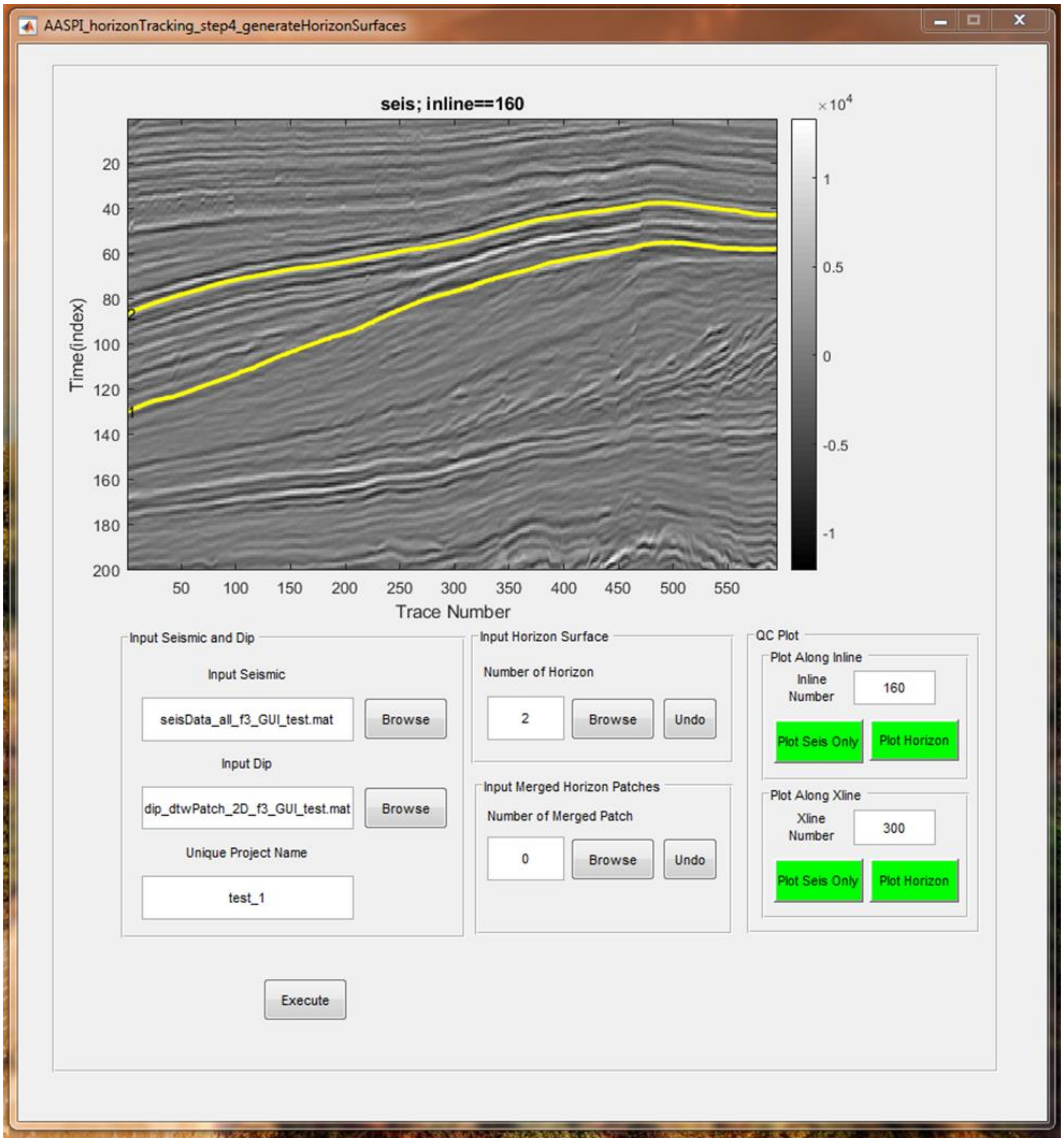
Number of Horizon: 1

Number of Merged Patch: 0

Inline Number: 160

Xline Number: 300

Interface after the input of seismic data, dip, and two horizons



Input Seismic: seisData_all_f3_GUI_test.mat

Input Dip: dip_dtwPatch_2D_f3_GUI_test.mat

Unique Project Name: test_1

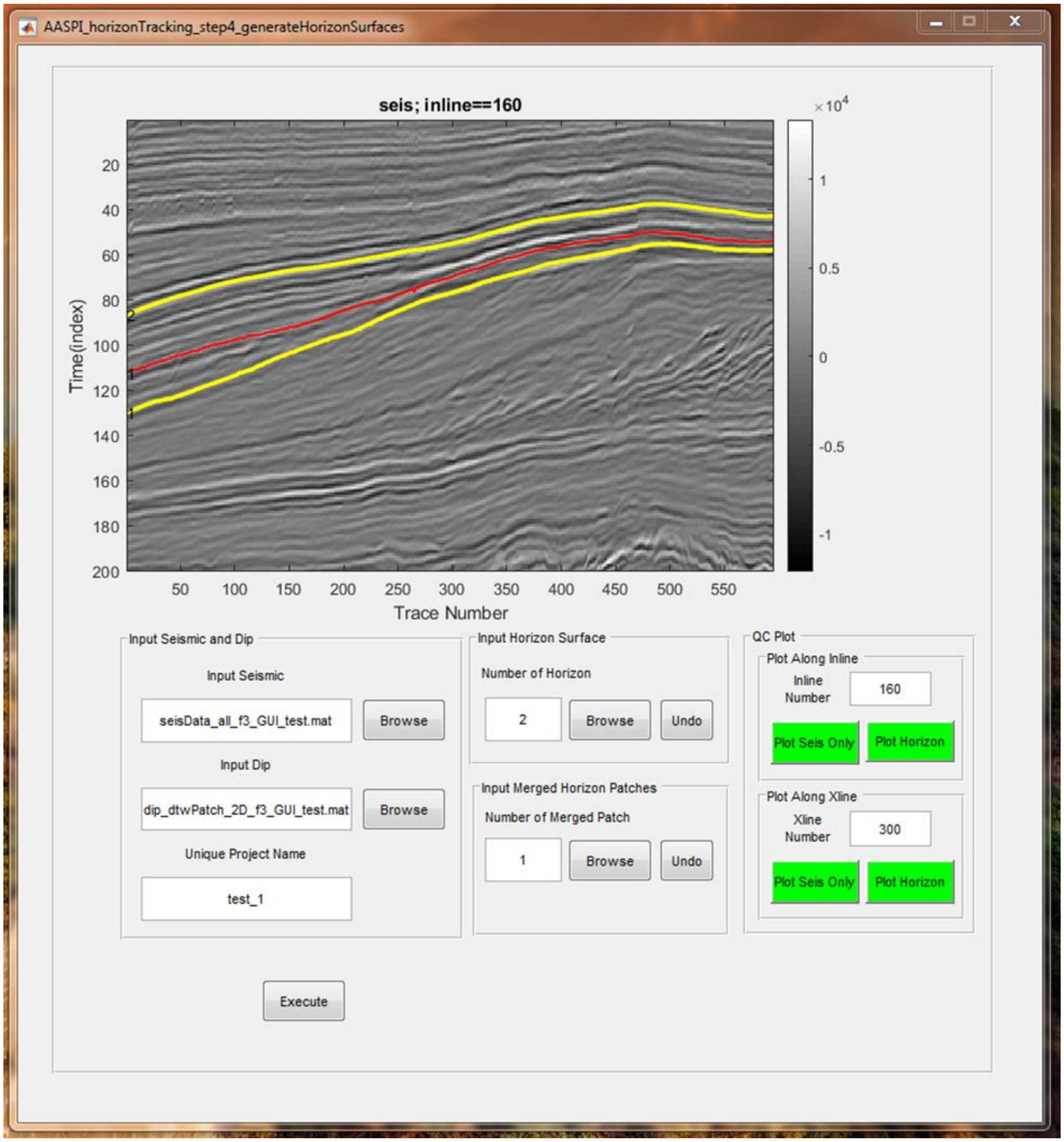
Number of Horizon: 2

Number of Merged Patch: 0

Inline Number: 160

Xline Number: 300

Interface after the input of seismic data, dip, two horizons, and the merged horizon patches (red curve).



Input Seismic and Dip

Input Seismic: seisData_all_f3_GUI_test.mat

Input Dip: dip_dtwPatch_2D_f3_GUI_test.mat

Unique Project Name: test_1

Input Horizon Surface

Number of Horizon: 2

Input Merged Horizon Patches

Number of Merged Patch: 1

QC Plot

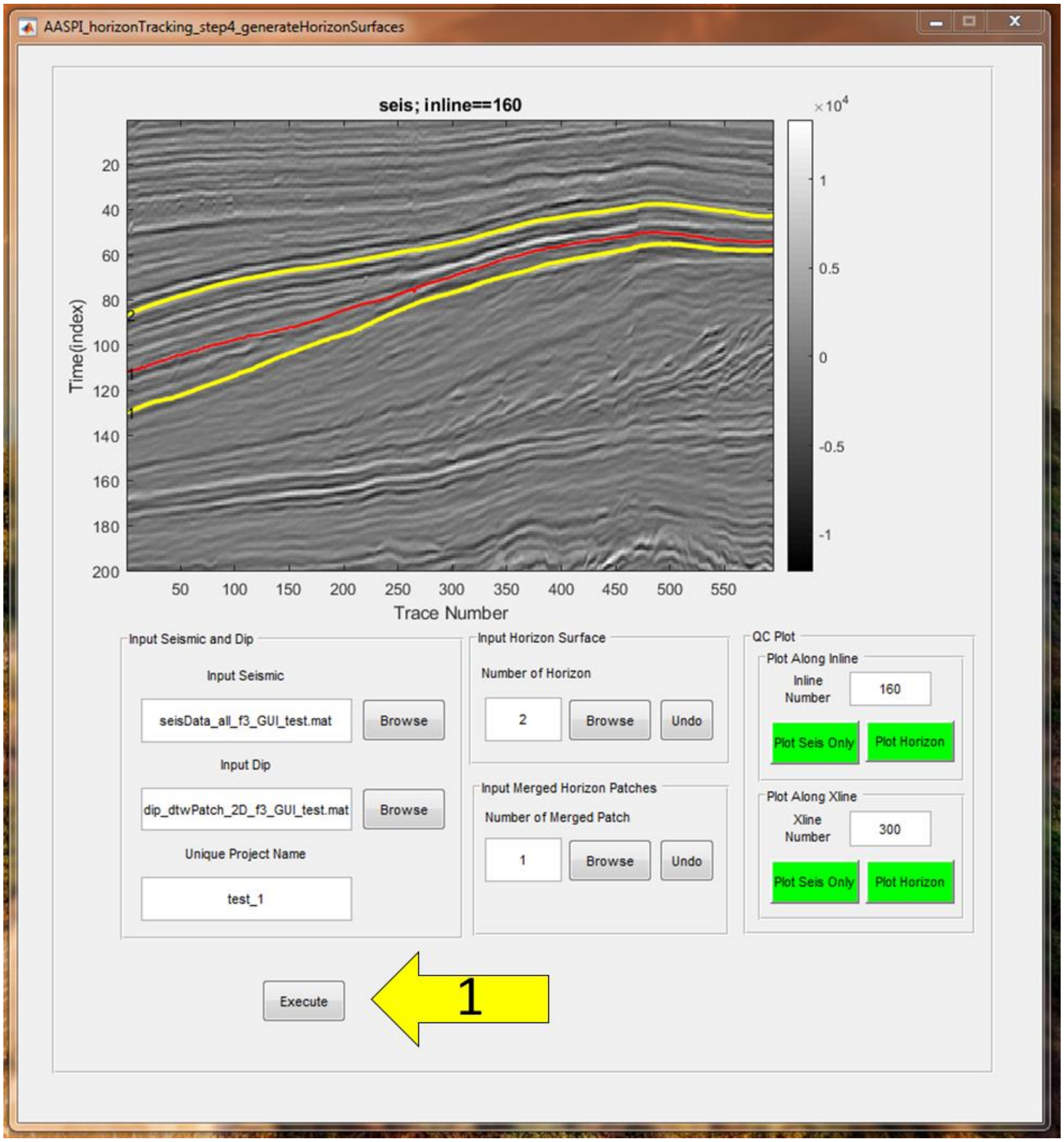
Plot Along Inline

Inline Number: 160

Plot Along Xline

Xline Number: 300

Finally, we click "execute" (arrow 1) to generate horizon surface.



Input Seismic and Dip

Input Seismic: seisData_all_f3_GUI_test.mat [Browse]

Input Dip: dip_dtwPatch_2D_f3_GUI_test.mat [Browse]

Unique Project Name: test_1

Input Horizon Surface

Number of Horizon: 2 [Browse] [Undo]

Input Merged Horizon Patches

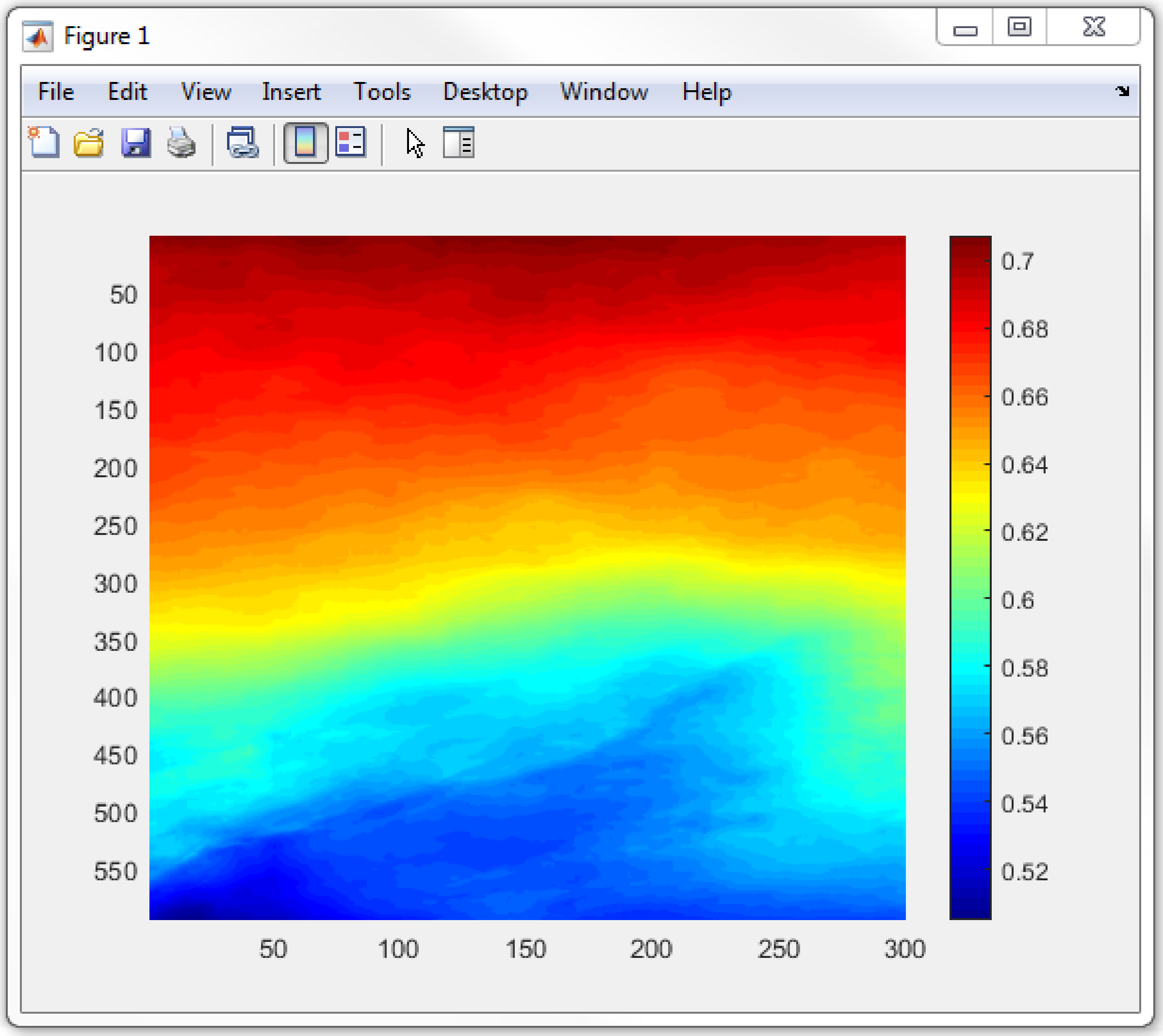
Number of Merged Patch: 1 [Browse] [Undo]

QC Plot

Plot Along Inline: Inline Number 160 [Plot Seis Only] [Plot Horizon]

Plot Along Xline: Xline Number 300 [Plot Seis Only] [Plot Horizon]

The automatically produced horizon surface.



Theory: Loop-tie checking of automatically extracted horizons

We employ an improved dynamic time warping (IDTW) to compute the seismic reflector's dip. The alignment lags between the two signals are regarded as the reflectors dips. Current dip computation algorithms usually need users defining an analysis window centered at analysis trace. However, the dip computed using an analysis window is the "average" dip between seismic traces within the analysis window. Extracted horizons using "average" dip may not strictly follow seismic events. Thus, we propose to compute two dip values: "left" and "right" dips for each sample of seismic trace.

Manual horizon picking is performed on inline and crossline vertical slices independently. Thus, one of the most important task is checking whether picked horizons of same seismic traces on inline and crossline vertical slices pass the same two-way travel time and this task is named as loop-tie. Although 3D automatic horizon extracting algorithms produce horizons meeting the loop-tie checking. However, there is no guarantee that the extracted horizon strictly follows the same seismic reflection event. Automatic horizon picking algorithms are based on a pre-computed seismic attributes such as seismic reflector's dip. Considering that signal-to-noise ratio of seismic events varies within the 3D seismic survey, it is impossible to obtain an accurate seismic attribute over the whole 3D seismic survey. Thus, it is impossible for automatic horizon picking algorithms to have the extracted horizons following the seismic events over the whole seismic survey. However, we notice that automatic extracted horizons strictly follow the seismic events if the left (right) dip of the analysis seismic trace equals to the right (left) dip of the nearby left (right) seismic trace.

Figure 1a shows a representative referred trace (blue curve) and two target seismic traces (black curves). We first obtain the left dip by aligning the referred seismic trace with the left target seismic trace and then obtain the right dip by aligning the referred seismic trace with the right target seismic trace. In this paper, we employ the IDTW to align the seismic traces. The yellow dot in Figure 1a is a representative sample of the referred trace. The red and blue arrows in Figure 1a indicate the calculated "left" and "right" reflector dips for the yellow dot, respectively. The red and blue dots in Figure 1a of target seismic traces are the corresponding aligned samples of the yellow dot. There is an obvious difference between the left and right dip. We compute the right dip of the red dot located on the trace one where the traces one and two function as the referred and target seismic traces, respectively (Figure 1b). Similarly, we compute the left dip of the blue dot located on the trace three in Figure 1b where the traces three and two function as the referred and target seismic traces, respectively. The red arrows in Figures 1a and 1b indicate the yellow and red dots function as the "corresponding" dots with each other. The red arrows in Figures 1a and 1b further indicate that the "left" reflector dip of the yellow dot equals to the "right" reflector dip of the red dot. We name the pair of yellow and red dots as the "matched pair". However, the blue arrows in Figures 1a and 1b indicate that the yellow and blue dots fail to function as "corresponding" dots with each other. The blue arrows in Figure 1a and 1b also indicate that the "right" dip of the yellow dot does not equal the "left" dip of the blue dot. We name the pair of yellow and blue dots as the "mismatched pair".

Theory: Loop-tie checking of automatically extracted horizon (continued)

$$t(j) = t_{seed} + \sum_{k+1}^{j-1} \Delta t_m^{right}\big(t(m)\big), \tag{1a}$$

$$k + 1 \leq m \leq j - 1, \tag{1b}$$

where t_{seed} is the two-way travel time of the control point at the referred seismic trace *k*; $\Delta t_m^{right}\big(t(m)\big)$ is the right dip of the trace m at two-way travel of $t(m)$. We use equation 2 to obtain the two-way travel time of trace *i* if trace *i* is located at the left side of referred seismic trace *k*

$$t(i) = t_{seed} - \sum_{k-1}^{i-1} \Delta t_n^{left}\big(t(n)\big), \tag{2a}$$

$$k - 1 \leq n \leq i - 1. \tag{2b}$$

Where $\Delta t_n^{left}\big(t(n)\big)$ is the left dip of the trace n at two-way travel of *t(n)*.

The loop tie checking of our algorithm is based on forward and backward horizon tracking and consists of horizon tracking on inline and crossline slices independently. The vertical yellow line in Figure 2a is the seed seismic trace and the blue cross mark indicates the control point. We first use equations 1 and 2 to track the horizon, which is the yellow curve in Figure 2a, on an inline vertical slice and this procedure is name as "forward horizon tracking". Then we treat the extracted two-way time, which are the pink cross marks in Figures 2a and 2b, of trace *j* (other than the seed seismic trace) as the control point. We use equation 1, if trace *j* is located at the left side of the seed trace, or equation 2, if trace *j* is located at the right side of the seed trace, to generate a backward horizon. The back tracking is performed until the initial control point is reached (blue cross marker in Figure 2a).

The red dash curves in Figures 2a and 2b show two representative backward tracked horizons. The forward tracked horizon in Figure 2a (yellow curve) and the backward tracked horizon (red dashed curve) perfectly coincides with each other, and the extracted horizon at trace *j* in Figure 2a is a loop tie meeting extraction. The backward extracted horizon in Figure 2b fails to coincide with each other and the extracted horizon at trace *j* in Figure 2b is not a loop tie meeting extraction. Loop tie checking is performed for each seismic trace of the inline slice and we only keep those extracted two-way travel time of seismic traces that are loop tie meeting extractions. The accepted portion and rejected portion of the horizon are yellow and red curves shown in Figure 2c, respectively.

Loop-tie checking is based on analyzing the relationship between extracted horizons on nearby inline and crossline slices. Figure 3a demonstrates the steps of checking whether the automatically extracted horizons between nearby inline and crossline slices meet the loop-tie. Inline seismic slice AA' and corresponding yellow curve in Figure 3a are the same seismic slice and extracted yellow horizon shown in Figure 2a. The extracted horizon between seismic trace number four and seismic trace number one is the same accepted yellow horizon shown in Figure 2c. We then automatically track the horizon along crossline BB' where seismic trace number 1 is the referred seismic trace, and the extracted two-way travel time of trace number 1 is the control point. We accept the extracted horizon between seismic traces number 1 and number 2 on crossline slice BB' using the loop-tie checking procedure illustrated in Figure 2. We next track the horizon along crossline slice DD' where the seismic trace number 4 as referred seismic trace, and extracted the two-way travel time

Theory: Loop-tie checking of automatically extracted horizon (continued)

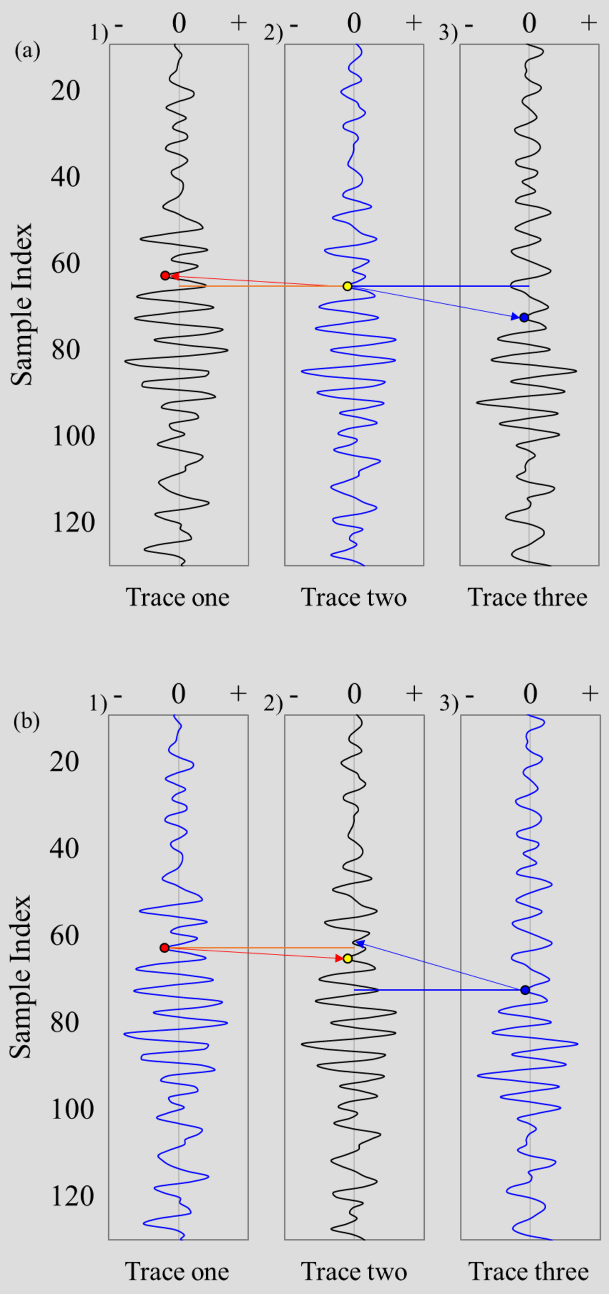


Figure 1. (a) The calculated "left" and "right" reflector dips of the representative sample (yellow dot) located on the referred trace (blue curve) using IDTW. (b) The calculated "right" dip of the red dot and "left" dip of the blue dot located on the target traces (black curves) using IDTW.

Theory: Loop-tie checking of automatically extracted horizon (continued)

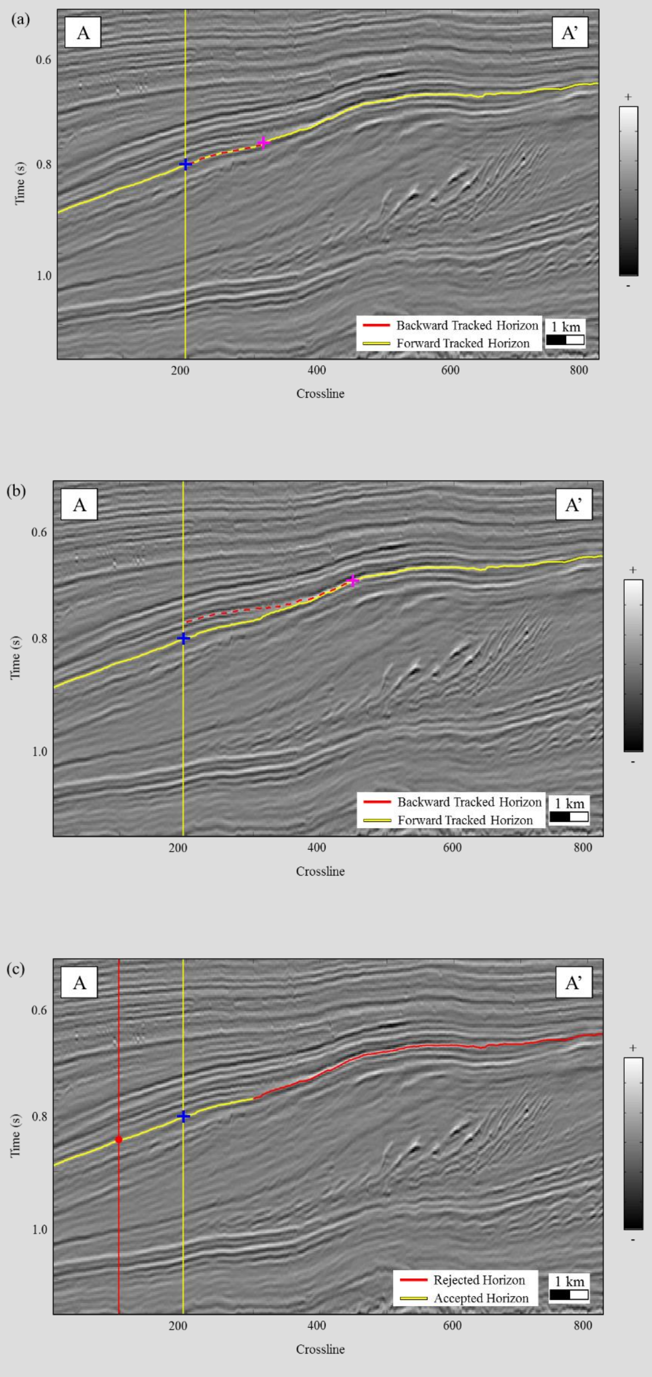
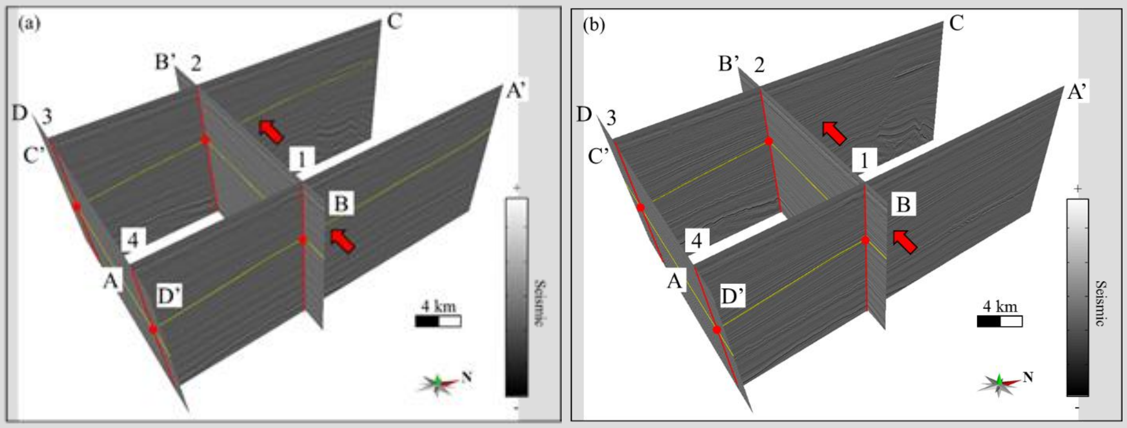


Figure 2. A representative example of the loop-tie checking applied in 2D case. The forward tracked horizon with (a) a backward tracked horizon which meets loop-tie checking, and (b) a backward tracked horizon which fails to meet loop-tie checking. (c) The result of accepted and rejected horizons after the loop tie checking.

Theory: Loop-tie checking of automatically extracted horizon (continued)

We accept the extracted horizon between seismic trace number 3 and seismic trace number 4 on crossline slice DD' using the loop-tie checking procedure illustrated in Figure 2. The accepted horizon size on crossline slice DD' is larger than that of crossline slice BB'. Thus, we choose to track horizon on inline slice crossing trace number 2. We finally track the horizon along inline slice CC' where the seismic trace number 2 as referred seismic trace and extracted two-way travel time of trace 2 as the control point. The extracted horizon on inline slice CC' exactly passes the extracted two-way travel time of trace number 3 of crossline slice DD'. We define the extracted horizon between seismic traces number 4, 1, 2, and 3 as loop-tie met horizon patch over the 3D seismic survey. In addition, we define the extracted horizon as loop-tie failed horizon patch if the extracted horizon on inline slice CC' fails to pass the extracted two-way travel time of trace number 3 of crossline slice DD'. Figure 3b shows an accepted loop-tie met horizon patch. Figure 3c shows a rejected loop-tie failed horizon patch. The trace number 5 is the last trace of accepted horizon along crossline slice DD'. Along inline slice EE', we track the horizon where the seismic trace number 5 is the as referred seismic trace and extracted two-way travel time of trace 5 is the control point. The extracted horizon on inline slice EE' fails to pass the extracted two-way travel time of trace number 6 of crossline slice BB'. We define the extracted horizon in Figure 3c between seismic traces number 4, 1, 6, and 5 as a loop-tie failed horizon patch.

Theory: Loop-tie checking of automatically extracted horizon (continued)



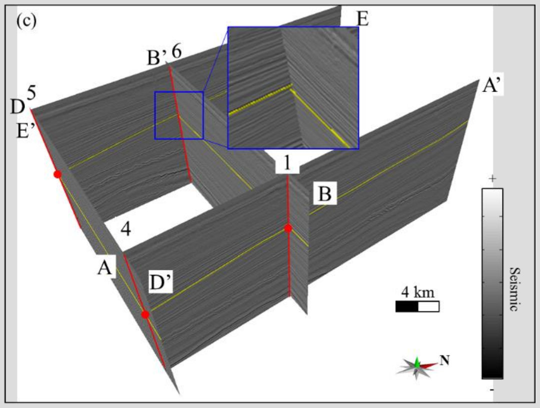


Figure 3. A representative example of the loop-tie checking applied in 3D case. (a) Four representative 2D seismic slices and corresponding tracked horizons before the loop-tie checking. (b) The accepted loop-tie met horizon patch. (c) The rejected loop tie failed horizon patch.

Theory: The generation of horizon patches that meet the defined loop-tie criteria

Three are four steps in the generation of a loop-tie mesh horizon patch over the whole seismic survey. The first step is extracting a horizon along the inline direction under the control point of the seed seismic trace. The second step is extracting horizons on user defined set of crossline slices, and the control points are the extracted two-way travel time from first step. The third step is extracting horizons on set of inline slices and the control points are the extracted two-way travel time of seismic trace of crossline slices from the second step. The final step is “cropping” the extracted horizon using the strategy illustrating in Figure 3. Figure 4 shows the base map of a seismic survey. The blue dot and line in Figure 4 are the seed seismic trace and inline slice AA’ crossing the seed seismic trace, respectively. The inline slice AA’ (Figure 4) is the same inline slice AA’ in Figure 2. We generate a set of crossline slices intersecting the inline slice AA’ with an interval of crossline slices of 10. When we project the extracted horizon of inline slice AA’ to any crossline slice, it is a single dot. The red dot in Figure 5 is the projected extracted two-way travel time on inline slice AA’. We then extract a horizon (yellow line in Figure 5) on crossline slice BB’ under the constraint of the red dot in Figure 5. We also extract horizons (Figure 6a) on other defined crossline slices under the constraints of interpretation on inline slice AA’. We then extract horizons (Figure 6b) on a user defined set of inline slices with the control points being the extracted two-way travel time of seismic trace of crossline slices. We finally extract the loop-tie met horizon patch (Figure 6c) using the strategy illustrated in Figure 3. In this study, we exam the loop-tie of an extracted horizon every 10 inline slices.

Theory: The generation of horizon patches that meet the defined loop-tie criteria (continued)

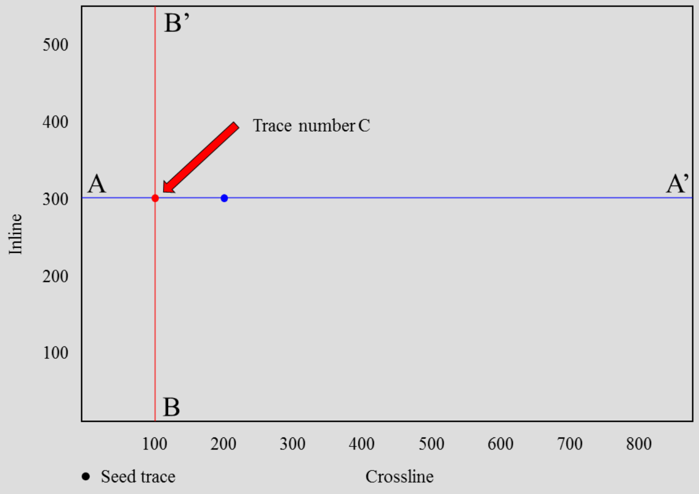


Figure 4. The base map of the seismic survey.

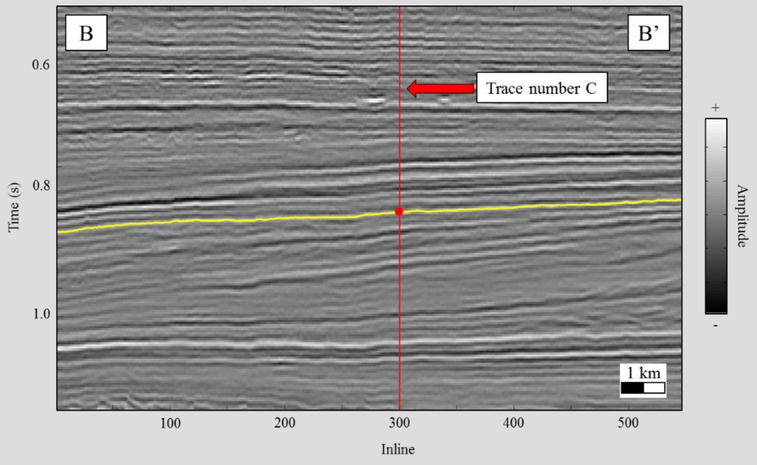


Figure 5. The tracked horizon (yellow line) on the crossline slice BB'.

Theory: The generation of horizon patches that meet the defined loop-tie criteria (continued)

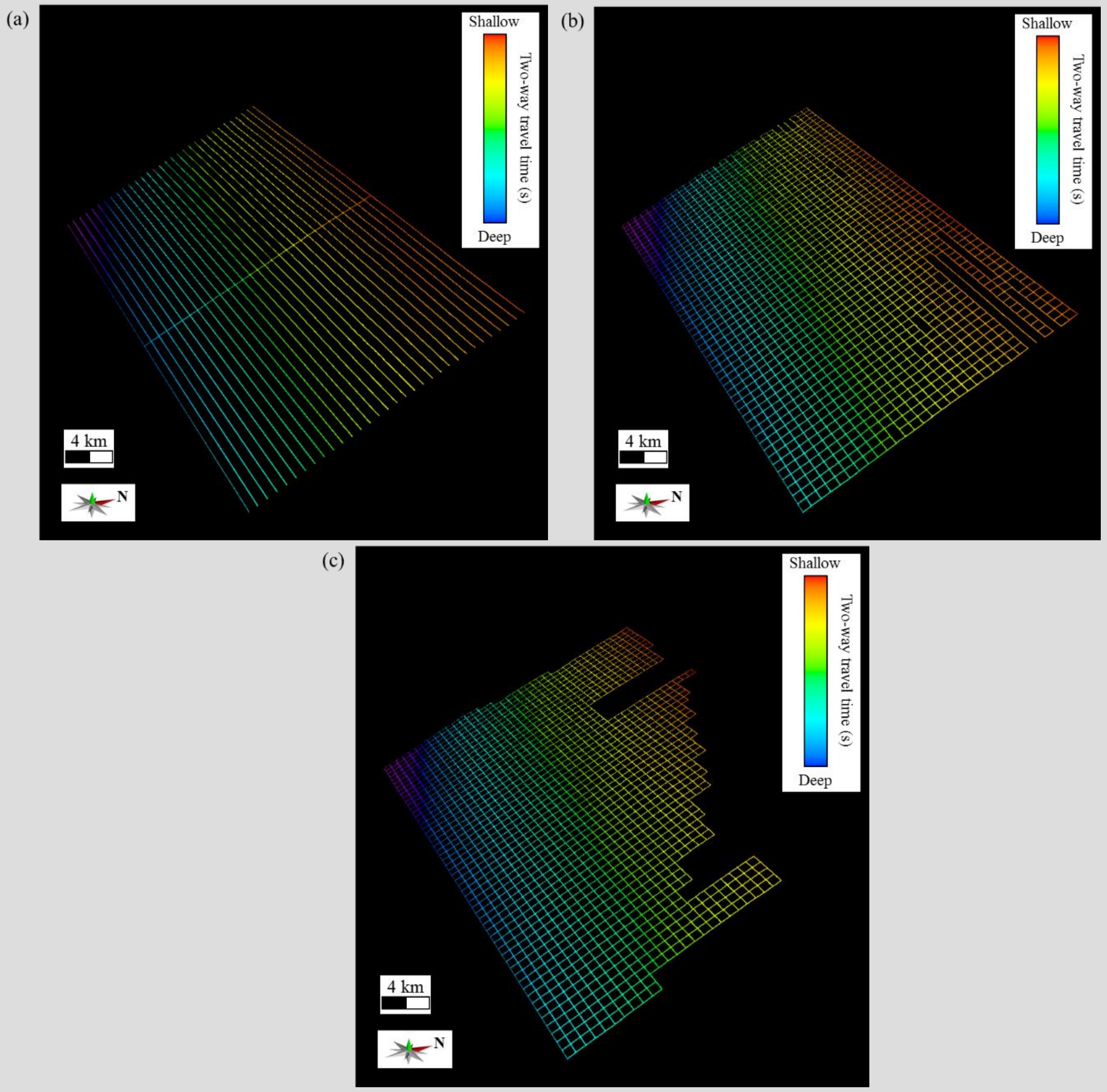


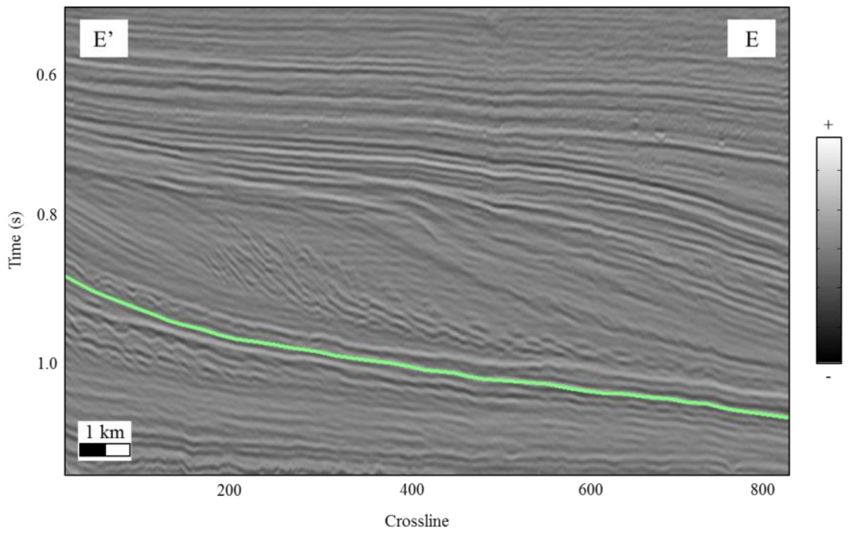
Figure 6. (a) The result after extracting horizons along crossline slices under the constraints of interpretation on inline slice AA'. (b) Before and (c) after cropping the extracted horizon using the loop-tie checking strategy illustrated in Figure 3.

Example

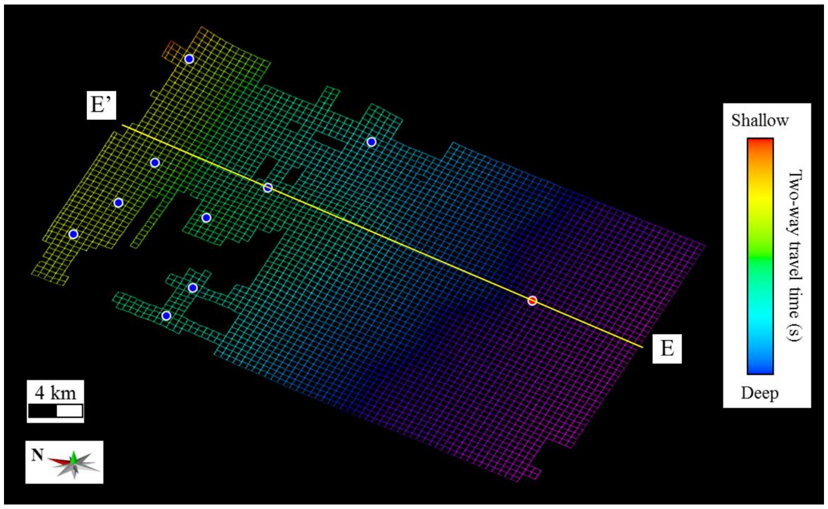
We propose the following three steps to generate horizon surfaces over the whole seismic survey after generating horizon patches:

- **Step 1:** Manual picking. Manually pick the target horizon at least on one vertical slice (inline, crossline, or arbitrary line).
- **Step 2:** Horizon patches searching and merging. Automatically search and merge horizon patches belong to the target horizon.
- **Step 3:** Horizon surface producing. Automatically generate horizon surface under the constraints of merged horizon patches.

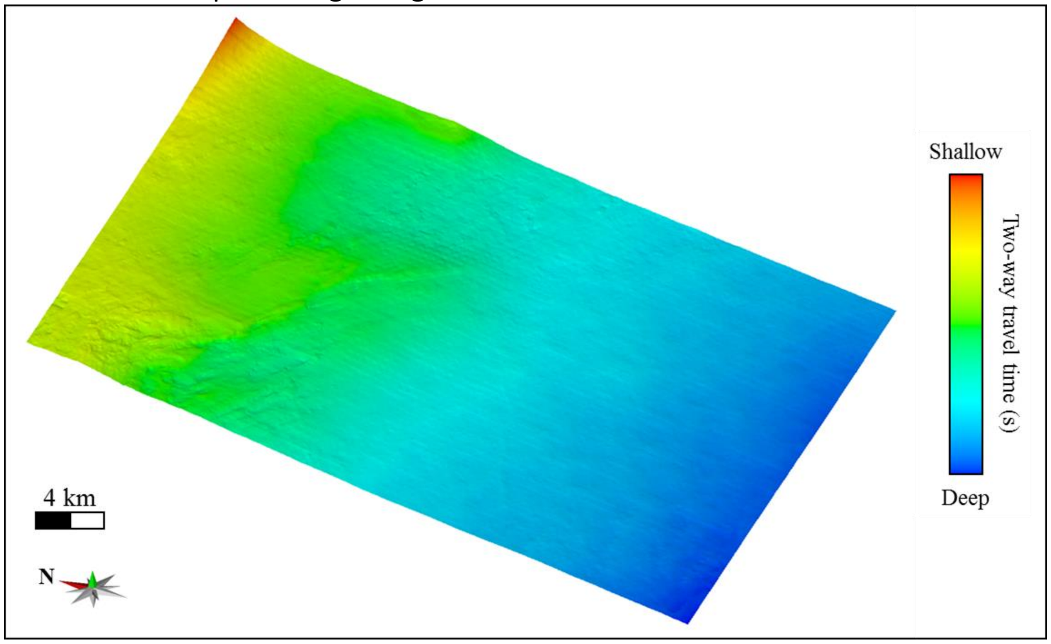
The example from F3:
Step 1: Manual picking. Manually pick the target horizon on the representative inline slice.



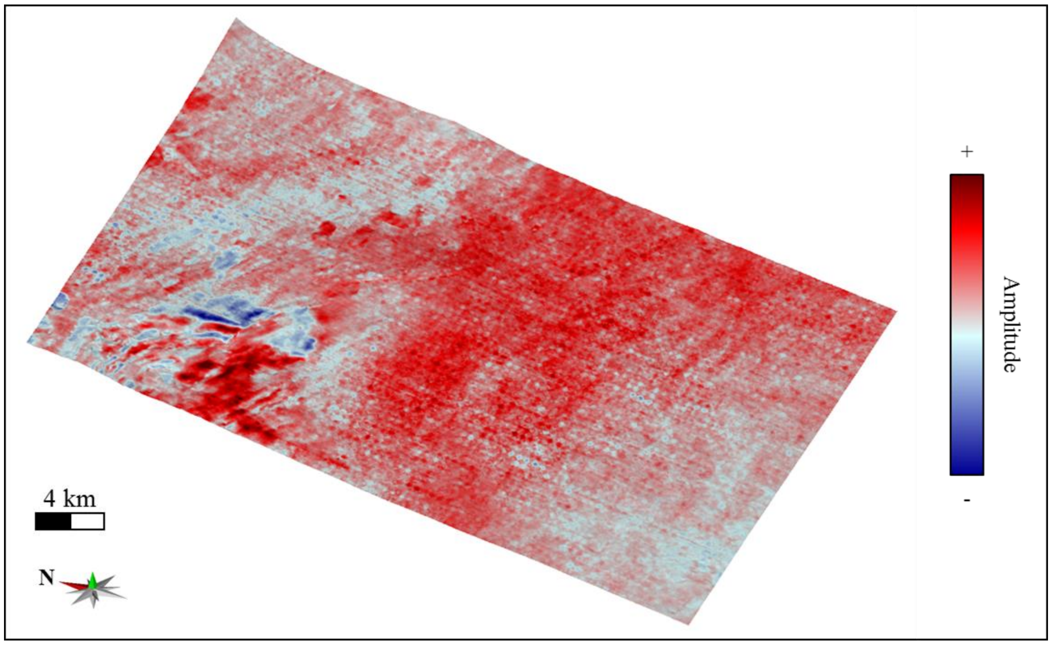
Step 2: Horizon patches searching and merging.



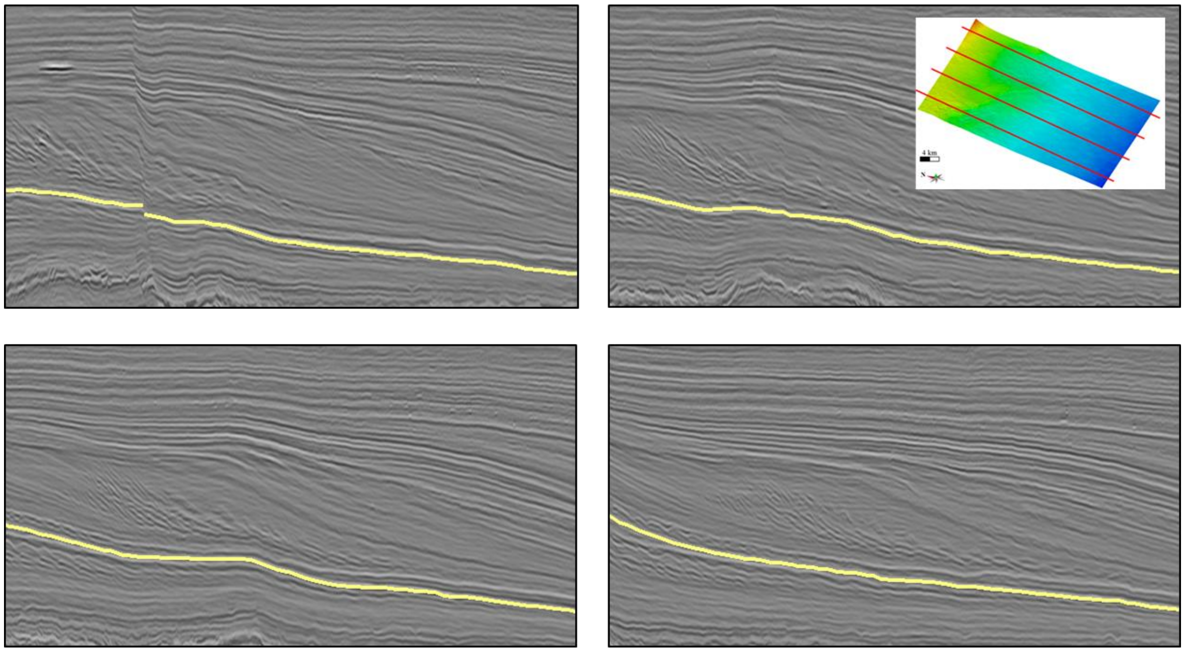
Step 3: Horizon surface producing. We generate horizon surface over the whole seismic survey.



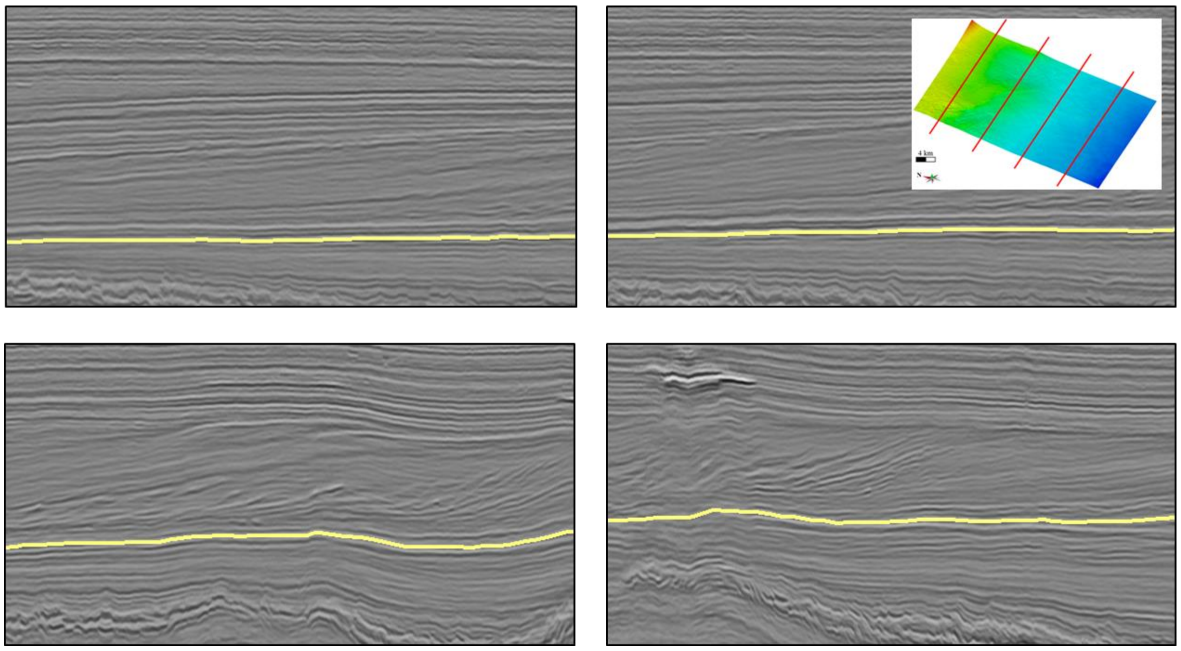
We have a uniform distribution for seismic amplitude slice expect nearby the fault zones.



The extracted horizon on representative inline slices. The constructed horizon strictly follows local seismic reflection events, even near fault zones and unconformity zones.



The extracted horizon on representative crossline slices.



References

Lou, Y., B. Zhang, H. Fang, and D. Cao, 2020, Simulating the procedure of manual seismic horizon picking: Geophysics (under review).

Lou, Y., B. Zhang, T. Lin, and D. Cao, 2020, Seismic horizon picking by integrating reflector dip and instantaneous phase attributes: Geophysics, 85(2), O37-O45.