Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Computer Science 18 (2013) 732 – 741

Procedia
Computer Science

International Conference on Computational Science, ICCS 2013

Seismic Image Restoration Using Nonlinear Least Squares Shape Optimization

Mathieu Gilardet[a,b,*], Sebastien Guillon[a], Bruno Jobard[b], Dimitri Komatitsch[c]

[a]*TOTAL, Pau, France*
[b]*LIUPPA, University of Pau, France*
[c]*LMA, CNRS UPR 7051, Université Aix-Marseille, Centrale Marseille, France*

Abstract

In this article we present a new method for seismic image restoration. When observed, a seismic image is the result of an initial deposit system that has been transformed by a set of successive geological deformations (flexures, fault slip, etc) that occurred over a large period of time. The goal of seismic restoration consists in inverting the deformations to provide a resulting image that depicts the geological deposit system as it was in a previous state. Providing a tool that quickly generates restored images helps the geophysicists to recognize geological features that may be too strongly altered in the observed image.

The proposed approach is based on a minimization process that expresses geological deformations in terms of geometrical constraints. We use a quickly converging Gauss-Newton approach to solve the system.

We provide results to illustrate the seismic image restoration process on real data and present how the restored version can be used in a geological interpretation framework.

Keywords: Seismic image restoration, least squares minimization process, Gauss-Newton solver.

1. Introduction

In this article we present a new method for seismic image restoration in order to help geophysicists in their interpretation of sedimentary deposit systems. In oil and gas exploration, seismic images are acquired by propagating, recording and processing acoustic waves underground. These images highlight geological events such as *horizons* (stratigraphic limits) or *faults* (discontinuities). Because of structural deformations of the underground that have occurred over time, such as flexure or displacements along faults, important geological structures might be difficult to detect and reconstruct (see figure 1). Our aim is to provide an easily configurable restoration tool that corrects the structural deformations to facilitate the identification of subtle geological events. Seismic restoration consists in finding a set of successive transformations between the observed image and an *unwarped* version that represents the initial sedimentary deposit system.

Two types of deformation operations might successively intervene during the restoration process. One is to split a seismic image along a geological fault and to slide one resulting block along this fault in order to get the

[*] Corresponding author. Tel.: +33-559-835518.
E-mail address: mathieu.gilardet@gmail.com.

Selection and peer review under responsibility of the organizers of the 2013 International Conference on Computational Science
doi:10.1016/j.procs.2013.05.237

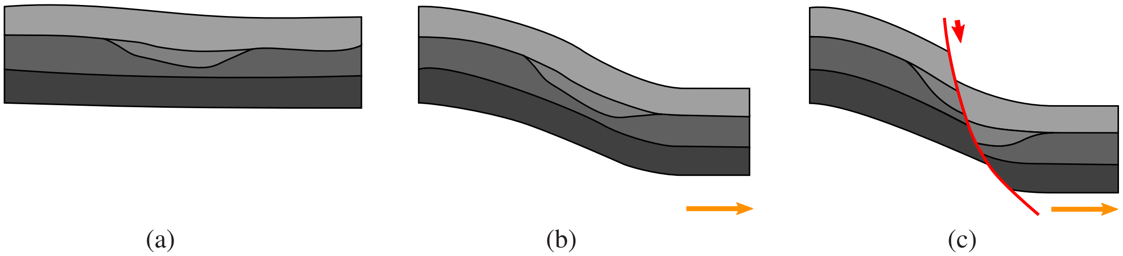


Fig. 1. **Successive geological deformations of the underground.** Over time, the structure of an initial sedimentary deposit (a) is deformed under the action of tectonic constraints (b) that might lead to fault formation and block displacements (c). To facilitate understanding the resulting structure, geophysicists apply several **restoration processes** that transforms it in the opposite order ($c \rightarrow b$: *displacement along the fault*, then $b \rightarrow a$: *horizon flattening*) to recover the initial deposit system.

horizons matching on each side of the fault (see figure 1, $c \rightarrow b$). The second operation is to deform the whole seismic image so that a selected horizon gets flattened along with the surrounding blocks, therefore restoring the initial structure of the geological deposit (see figure 1, $b \rightarrow a$).

In the domain of seismic imagery, different solutions [1], [2], [3] and [4] have been proposed to globally restore a seismic image. All these solutions are based on automatic image analysis, estimating the local dip of each horizon and then providing a global flattening process. This approach is effective as long as seismic quality is good enough over the whole domain and that no structural deformation such as faults has altered the underground configuration. This last aspect is a major limitation since geological regions of interest are very likely to present such faults.

The geomechanical aspects of fault restoration are of great interest for modelling and understanding a deformation process [5], [6]. The objective here concerns the quantification of deformation history and strain analysis. Based on geometric assumptions and mechanical rock properties, the restoration problem is solved based on a geomechanical approach. The methodology first requires defining a mesh conforming to geological interfaces (fault and horizons), associating mechanical properties to rock units and finally, based on boundary conditions, using a finite-element approach to solve the problem. These approaches does not completely suit our restoration needs, as they require to provide geomechanical properties that are not available at the early stage of the structural interpretation of seismic images. We therefore focus our attention on lighter geometrical methods that provide acceptable geological deformations without the need of injecting explicit geomechanical properties to initiate the process.

In the computer graphics field, more general purpose shape deformation techniques are able to achieve physically plausible deformation results [7], [8]. Weng et al. proposed a 2D shape deformation method that preserves the boundary curve of the shape and the local areas inside the shape [9]. Preservation of the geometrical constraints is expressed as a nonlinear least squares minimization and is resolved iteratively with a Gauss-Newton method.

Our restoration technique is an adaptation of the approach of Weng et al. whose purely geometrical local areas preservation scheme has proven to be sufficient to mimic the geological deformation behavior in the context of seismic image interpretation. We extend their shape deformation algorithm by defining two supplementary *restoration constraints* that control the fault restoration and horizon flattening operations.

The article is organized as follows. Section 2 presents the general shape deformation framework. In section 3 we then introduce the specific constraints that handle fault restoration and horizon flattening. In section 4 we finally present applications and results.

2. Shape Deformation Framework

In this section we present the global scheme of shape deformation as proposed in [9]. This scheme is based on a mesh deformation: the initial mesh is obtained by a triangulation of the input domain and is transformed in order to minimize a set of criteria.

We will introduce two families of criteria to define what points should be transformed (restoration constraints) and how the transformation is extended to other points (geometrical constraints). This combination will produce a non linear system that we will subsequently solve based on a Gauss-Newton approach.

Finally, the use of texture mapping will directly provide the transformed image on the deformed mesh.

In the following, we note $\mathcal{V}$ the set of nodes belonging to the mesh $\mathcal{M}$ and $\mathcal{V}_p$ the subset of $\mathcal{V}$ containing the nodes from the contour of $\mathcal{M}$. $\mathcal{V}_g$ is the subset of interior nodes that do not belong to $\mathcal{V}_p$. $\mathcal{V}$ has n elements.

2.1. Restoration Constraints

The basic mechanism of shape deformation consists in defining one or several constraints that are applied to different subsets of mesh vertices. Each constraint will be applied to a subset of vertices $\mathcal{V}_c = \{v_c\} \in \mathcal{V}$. Considering a function c that provides the new positions of a set of vertices depending on their current position, the transformation of the set of vertices can be expressed as:

$$CV = c(V) \tag{1}$$

where

- C is a $(card(\{v_c\}) \times n)$ matrix, such that $C_{r,c} = 1$ when v_c is the r^{th} element of $\mathcal{V}_c$ and the c^{th} element of $\mathcal{V}$; $C_{r,c} = 0$ otherwise.
- V is the $(n \times 2)$ vertices matrix, such as: $\forall v_k \in \mathcal{V}, V_k = [v_{kx}, v_{ky}]$

The previous geometrical transformation can be expressed as a geometrical constraint by considering the new positions $c(V)$ as a target to reach. The deformation process will then try to minimize the difference between the current and target positions. Therefore, we write the restoration constraint J_C to minimize as:

$$J_C = ||CV - c(V)||^2 \tag{2}$$

In section 3, we will describe how to adapt J_C to insert the two constraints responsible for *fault restoration* and *horizon flattening*.

2.2. Geometrical Constraints

After having described how to impose constraints onto specific subsets of vertices, we now describe how the deformation will be propagated into the entire mesh. It is indeed important for deformation to be as uniformly distributed as possible to avoid localized distortions. This is the role of what we call the *geometrical constraints* that tend to preserve the shape and area of each triangle. As stated in [9] this objective is achieved by preserving the *mean value coordinates*, the *contour Laplacian* and *edge lengths*. The following sections describe how we formulate these constraints.

2.2.1. Mean Value Coordinate

For each point v_i of $\mathcal{V}_g$ and $v_{i,j}$ its adjacent neighbors, its mean value coordinate [10] is defined as the set of $\omega_{i,j}$ weights such that:

$$v_i = \sum \omega_{i,j} * v_j \tag{3}$$

with

$$\omega_{i,j} = \frac{\lambda_{i,j}}{\sum_j \lambda_{i,j}} \ , \ \lambda_{i,j} = \frac{tan(\alpha_{j-1}/2) + tan(\alpha_j/2)}{|v_i - v_j|} \tag{4}$$

and α_j is the angle between $[v_i v_j]$ and $[v_i v_{j+1}]$.

The mean value coordinate over the entire shape can be written in matrix form as:

$$MV = 0 \tag{5}$$

with

- M a $(card(\mathcal{V}_g) \times n)$ matrix such that:
  - $M_{r,c} = \omega_{i,j}$ when v_i is the r^{th} element of $\mathcal{V}_g$, and v_j is the c^{th} element of $\mathcal{V}$.
  - $M_{r,c} = -\sum \omega_{i,j}$ when v_i is the r^{th} element of $\mathcal{V}_g$, and the c^{th} element of $\mathcal{V}$.
  - $M_{r,c} = 0$ otherwise.
- V the vertices matrix.

Preserving the mean value coordinate during shape deformation will consist in minimizing:

$$J_M = ||MV||^2 \, . \tag{6}$$

2.2.2. Contour Laplacian

Because mean value coordinates are limited to points which are the kernel of a star shaped polygon (as detailed in [10]), this formulation could not apply to contour points $v_i \in \mathcal{V}_p$. For this reason we estimate a Laplacian coordinate δ_i on the initial mesh:

$$\delta_i = \mathcal{L}_p(v_i) = v_i - (v_{i-1} + v_{i+1})/2 \tag{7}$$

And δ_i is introduced as a constraint to be preserved. Written for all $v_i \in \mathcal{V}_p$, we minimize:

$$J_L = \sum_{v_i \in \mathcal{V}_p} ||\mathcal{L}_p(v_i) - \delta_i||^2 = ||\Delta V - \delta(V)||^2 \tag{8}$$

with

- Δ the $(card(\mathcal{V}_p) \times n)$ matrix with
  - $\Delta_{r,c} = -1/2$ when v_i is the r^{th} element of $\mathcal{V}_p$, and v_{i-1} or v_{i+1} is the c^{th} element of $\mathcal{V}$.
  - $\Delta_{r,c} = 1$ when v_i is the r^{th} element of $\mathcal{V}_p$, and the c^{th} element of $\mathcal{V}$.
  - $\Delta_{r,c} = 0$ otherwise.
- V the vertices matrix.
- $\delta(V)$ the function that returns the $(2 \times card(V_p))$ matrix: $\delta(v_i) = T_i \delta(v_i^0)$, such that:
  - v_i^0 is the position of v_i before deformation.
  - T_i is the transformation matrix defined by: $T_i = \sum_{(i,j) \in \mathcal{E}_p} (v_j - v_i)(v_j^0 - v_i^0)^T D_i$
  - $D_i = (\sum_{(i,j) \in \mathcal{E}_p} (v_j - v_i)(v_j^0 - v_i^0)^T)^{-1}$

2.2.3. Edge Length Preservation

The mean value coordinates formulation has for objective to preserve the triangle shape, but because this measurement is invariant with scale, a constraint on edge length must be introduced to ensure the area preservation. Preserving edge lengths requires to minimize the following energy:

$$J_E = \sum_{(i,j)\in\mathcal{E}} \|(v_i - v_j) - e(v_i, v_j)\|^2 \tag{9}$$

with

$$e(v_i, v_j) = \frac{{l_{i,j}}^0}{l_{i,j}} \tag{10}$$

where $l_{i,j}$ is the current length, and ${l_{i,j}}^0$ the length before deformation.

We rewrite (9) into the matrix form:

$$J_E = \|LV - l(V)\|^2 \tag{11}$$

with

- L the $(card(\mathcal{E}) \times n)$ oriented incidence matrix.
- V the vertices matrix.
- $l(V)$ the function that returns the $(card(\mathcal{E}) \times 2)$ matrix, where the k^{th} rows represents the k^{th} edge under its vectorial form.

2.3. Solving the System

Combining all the energy terms of restoration and geometrical constraints, the global shape deformation consists in minimizing:

$$J = \alpha J_M + \beta J_L + \gamma J_E + \zeta J_C \tag{12}$$

or:

$$J = \alpha \cdot \|MV\|^2 + \beta \cdot \|\Delta V - \delta(V)\|^2 + \gamma \cdot \|LV - l(V)\|^2 + \zeta \cdot \|CV - c(V)\|^2 \tag{13}$$

The global energy can be rewritten in a simple matrix form as:

$$min\ \|AV - b(V)\|^2, \text{ with } A = \begin{bmatrix} \alpha \cdot M \\ \beta \cdot \Delta \\ \gamma \cdot L \\ \zeta \cdot C \end{bmatrix},\ b(V) = \begin{bmatrix} 0 \\ \beta \cdot \delta(V) \\ \gamma \cdot l(V) \\ \zeta \cdot c(V) \end{bmatrix} \tag{14}$$

As minimizing J is not a linear problem, we use a Gauss-Newton approach to solve the system iteratively. Thus at each iteration, we solve the following system:

$$\min_{t+1}\ \|AV^{t+1} - b(V^t)\|^2 \tag{15}$$

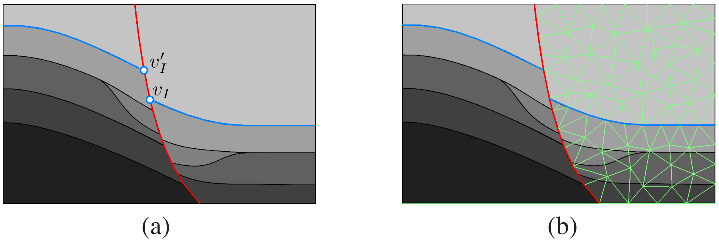
where V_t is the vector of current positions, and V_{t+1} the vector of updated positions at the next iteration. Equation 15 is then linearized using the formulation:

$$V^{t+1} = (A^T A)^{-1} A^T b(V^t)\,. \tag{16}$$

As A only depends on the mesh before deformation, the term $(A^T A)^{-1}A^T$ is computed once at the beginning and thus a single matrix multiplication is needed at each iteration. In practice, the method converges typically after a hundred iterations.

3. Fault Restoration and Horizon Flattening Workflow

We now introduce the specific constraints responsible for fault restoration and horizon flattening, which are the two restoration constraints introduced in the section 2.1. To better explain the workflow, let us apply it to the synthetic test case illustrated in figure 2.a. In this case, we have identified a fault (in red) and a horizon (in blue) and the objective is to first recover from the displacement along the fault and then, in a second restoration stage, flatten the horizon.



(a)

(b)

Fig. 2. **Restoration workflow applied to a seismic image.** (a) Initial model (b) Meshed model.

3.1. *Fault Restoration Procedure*

To restore a fault displacement, a mobile block is defined on one side of the fault while the other block on the other side of the fault remains static. We then need to define two constraints controlling the displacement of the mobile block. The first constraint keeps the mobile block in contact with the fault while the second is responsible for its displacement until the blue horizon matches on both sides of the fault.

3.1.1. *Meshing the mobile block*

We start meshing the mobile block from its contour (figure 2.b.). Before meshing we make sure to insert the intersection point v_I of the blue horizon in the mobile block within the fault because we will need it later to stitch the horizon of the mobile block with the corresponding point v'_I in the static block.

3.1.2. *Fault Sliding Constraint*

This first constraint must guarantee that the moving block will remain in contact with the fault during its deformation. Let $\mathcal{V}_F = \{v_F\}$ be the subset of $\mathcal{V}$ containing the vertices of the moving block belonging to the fault. We define $f(v_F)$ their projection target on the fault polyline. The cumulated distance to the fault that we need to minimize is:

$$J_{F1} = \sum_{v_F \in \mathcal{V}_F} (v_F - f(v_F))^2 = ||C_F V - c_F(V)||^2 \tag{17}$$

where

- C_F is a $(card(\mathcal{V}_F) \times n)$ matrix such that $C_{F\,r,c} = 1$ when v_F is the r^{th} element of $\mathcal{V}_F$ and the c^{th} element of $\mathcal{V}$; $C_{F\,r,c} = 0$ otherwise.
- V is the vertices matrix
- $c_F(V)$ is the function that returns the $(card(\mathcal{V}_F) \times 2)$ matrix that contains $[f(v_F)_x, f(v_F)_y]$ in its k^{th} row.

Let us mention that we exclude from $\mathcal{V}_F$ the vertex v_I that will be handled by the horizon stitching constraint in the next section.

3.1.3. Horizon Stitching Constraint

This constraint aims at moving the mobile block so that its v_I point gets as close as possible to the v'_I point on the edge of the static block. The energy term is:

$$J_{F2} = (v_I - v'_I)^2 = ||C_I V - c_I||^2 \tag{18}$$

with

- C_I the $(1 \times n)$ matrix such that $C_{I0,c} = 1$ when v_I is the c^{th} element of $\mathcal{V}$; 0 otherwise.
- V, the vertices matrix.
- c_I the (1×2) target position $c_I = [v'_{I_x}, v'_{I_y}]$.

3.1.4. Gathering the Fault Restoration Constraints

From the equation 12 the global energy that has to be minimized to restore the fault is expressed by:

$$J1 = \alpha J_M + \beta J_L + \gamma J_E + \zeta J_{F1} + \lambda J_{F2} \, . \tag{19}$$

Figure 3.a shows the result obtained at the end of the fault restoration process. As expected, the mobile block is still in contact with the fault and the blue horizon is now continuous across the fault.

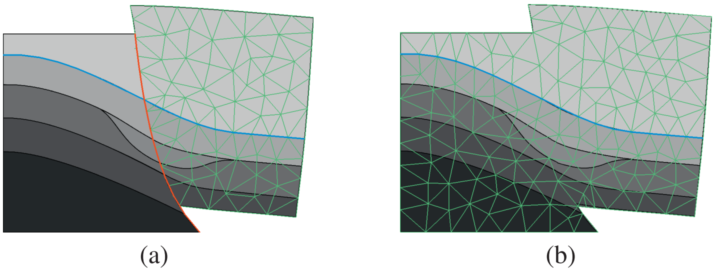


Fig. 3. **Fault restored model.** (a) After fault restoration (b) The remeshed domain that will be used to start the horizon flattening process.

3.2. Horizon Flattening Procedure

This part presents the second deformation process in our restoration workflow: the flattening of a horizon. Its aim is to provide an unfolded representation of the sedimentary deposit system. For this purpose, a horizon is selected and adequate constraints will be introduced in order to flatten it. An additional constraint will take care of preserving the horizon length during the operation.

3.2.1. Whole Domain Remeshing

At this stage, the fault displacement has been restored and the selected horizon is continuous across the whole domain. To start the next deformation step, we discard the current mesh, which only covers a limited part of the domain. We therefore remesh the whole domain from its contour and from the selected horizon. However the fault have been disabled with the first restoration step, we do not constraint the new mesh to its geometry.

3.2.2. Horizon Flattening Constraint

Let $\mathcal{V}_H = \{v_{Hk} \ (x_k, y_k)\}$ be the set of vertices belonging to the selected horizon. This constraint consists in displacing the v_{Hk} vertices on the selected horizon toward an horizontal line of equation $y = y_H$. Flattening the horizon will require minimizing the energy:

$$J_{H1} = \sum_{v_k \in \mathcal{V}_H} (y_k - y_H)^2 = ||C_H V = c_H(V)||^2 \tag{20}$$

where

- C_H is a $(card(\mathcal{V}_H) \times n)$ matrix such that $C_{H\,r,c} = 1$ when v_k is the r^{th} element of $\mathcal{V}_F$ and the c^{th} element of $\mathcal{V}$; $C_{F\,r,c} = 0$ otherwise.
- V is the vertices matrix
- $c_H(V)$ is the function that returns the $(\mathcal{V}_F \times 2)$ matrix that contains $[v_{k_x}, y_H]$ in its k^{th} row: .

3.2.3. Horizon Length Constraint

Expressing the length variation of the edge of the horizon consists in expressing the sum of length changes of its edges $\mathcal{E}_H$. Similarly to equation 11, we write:

$$J_{H2} = \|L_H V = l_H(V)\|^2 \tag{21}$$

where this time, L_H and $l_H(V)$ have $card(\mathcal{E}_H)$ rows.

However, since we want to explicitly formulate the horizon length energy, the regular length expression $\|LV - l(V)\|^2$ must now exclude the set of horizon edges $\mathcal{E}_h$.

3.2.4. Gathering the Horizon Flattening Constraints

From equation 12 the global energy that has to be minimized to flatten the horizon is expressed by:

$$J2 = \alpha J_M + \alpha J_L + \gamma J_E + \zeta J_{H1} + \lambda J_{H2} \tag{22}$$

Figure 4 shows the flattened version of figure 3.b.

The complete restoration process shows how the particular geological feature that was split by the fault on figure 2.a has been recovered. Its visual identification is thus now easier.

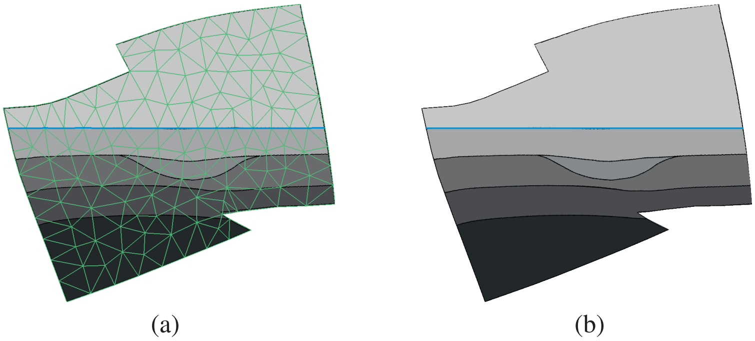


Fig. 4. **Restored model after flattening.** (a) Meshed (b) Plain.

4. Results

In this section we present the effects of the restoration processes on the real seismic image depicted in figure 5. On this image, the objective will consist in restoring the main fault (discontinuity in the middle of the image) and unfolding the dome structure. Restoration will allow for better understanding of the initial deposit system and will clarify horizon continuity through the actual fault.

Figure 6 shows the successive deformations of the seismic image going through the global restoration framework. First, an horizon and a fault are selected on the seismic image (blue and red lines in figure 6.a). The block on the right of the fault is then restored as explained in section 3.1 and the result is displayed in figure 6.b. Finally

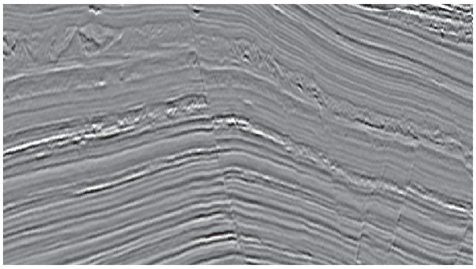
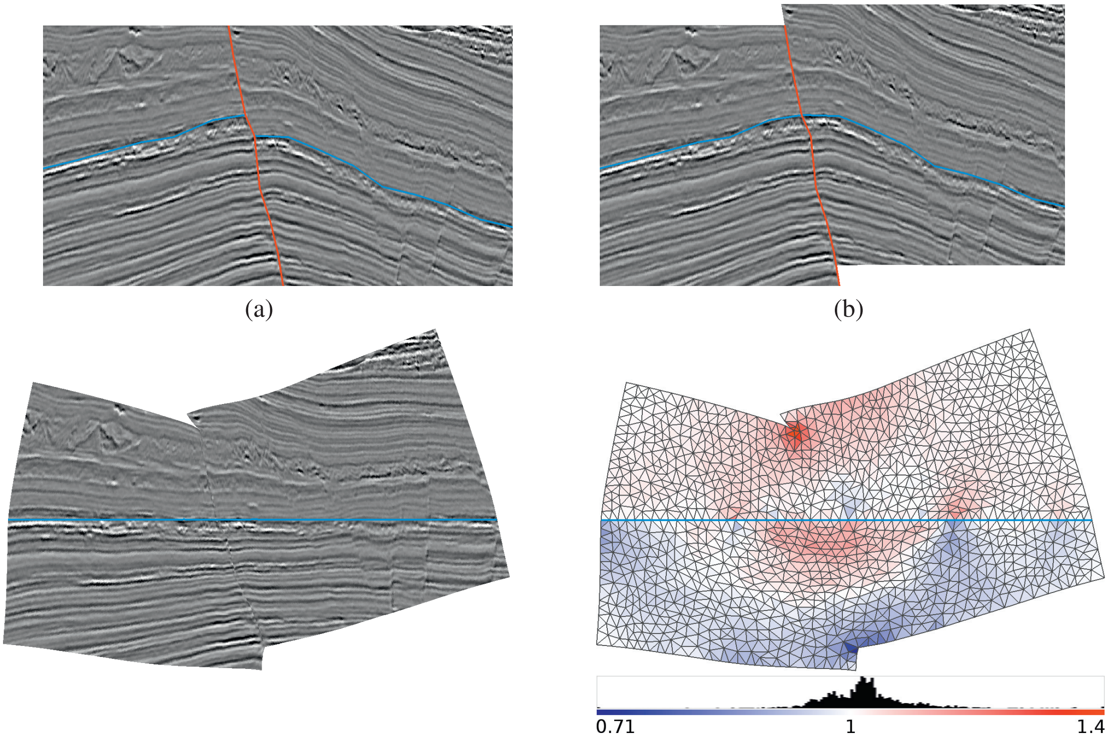


Fig. 5. **The initial seismic image** that has to be restored.



(a)

(b)

(c)

(d)

Fig. 6. **Restoration workflow applied to a seismic image.** (a) Interpreted image (b) Fault restoration (c) Final image (d) Deformation map and histogram: red shows compression, blue shows dilatation.

the blue horizon is flattened as explained in section 3.2 and the resulting image is presented in figure 6.c. At this stage, interpreters can more easily track continuous and flat horizons and ultimately identify geological features.

As a consequence of the competing constraints, the least squared minimization method usually cannot fully honor all of them. Figure 6.d shows the deformation ratio of each triangle (initial area over final area). The map thus depicts the typical regions of compression (red) and dilatation (blue) that inevitably occur during such restoration operations. The histogram distribution of deformations attests that the minimization behaves well since most of the triangles undergo only small distortions.

It appears through our experiments that imposing weighting coefficients in a ratio of 10 between restoration constraints and geometrical constraints was a good balance for our data. It means that in equations 19 and 22 we select, $\alpha, \beta, \gamma = 1$ and $\zeta, \lambda = 10$. Moreover we have observed good stability of the overall restoration when applying local variations to these coefficients. We conclude that for realistic cases, the geometrical constraints are met satisfactorily despite the strong weighting on the restoration constraints.

Dealing with the deformation of triangular meshes implies that the restoration process is intrinsically bijective and therefore that the inverse transformation can be defined. This leads to an interesting tool for interpreters because they can perform their picking in the restored domain and their work can automatically be transformed back in the original domain. We illustrate this possibility in figure 7, in which, we have transformed the yellow and green horizons picked in the restored domain (figure 7.a) in the original domain (figure 7.b).

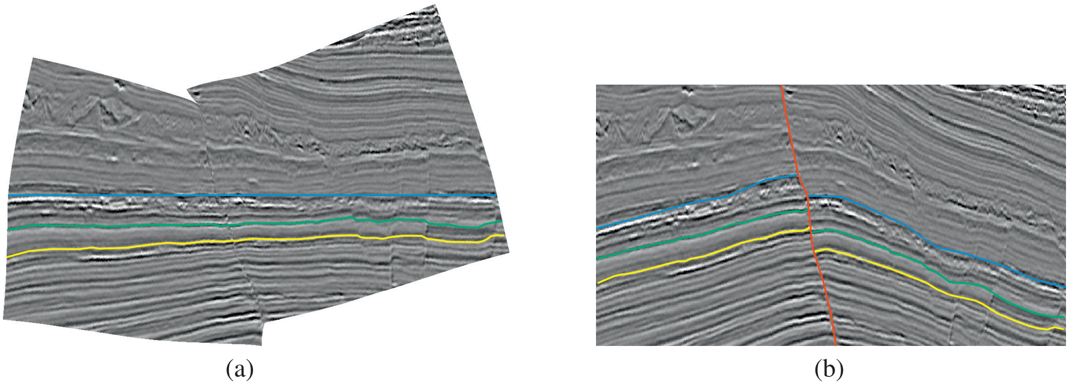


Fig. 7. **Interpreting simultaneously on both restored and original domains.** The original and restored images are connected through a bijective transformation. Therefore picking horizons on the restored domain (a) can interactively be displayed on the original domain (b).

Conclusions and future work

In this article we have presented a complete shape deformation framework that achieves seismic image restoration based on a non linear least squares optimization. Due to the fact that deformations are expressed as purely geometrical constraints, our method is much simpler to parameterize than geomechanical-based restoration approaches that require providing rock properties. In spite of this assumption, we obtain results that are good enough to perform seismic interpretation tasks directly on the restored images.

The method that we have introduced is fast enough to be used in interactive interpretation sessions and thus a large number of restoration hypotheses can be tested rapidly. This is particularly useful when restoring a fault, to try several feature matching positions.

In future work we plan to extend this framework to 3D image restoration since the underlying equations are not restricted to the 2D case.

References

[1] N. Keskes, S. Guillon, M. Donias, P. Baylou, F. Pauget, Method of chronostratigraphic interpretation of a seismic cross section or block (2004).

[2] J. Lomask, A. Guitton, S. Fomel, J. Claerbout, A. Valenciano, Flattening without picking, Geophysics 71 (4) (2006) 13–20.

[3] V. Toujas, M. Donias, D. Jeantet, S. Guillon, Y. Berthoumieu, A robust framework for geotime cube, in: ICIP'08, 2008, pp. 1880–1883.

[4] G. Zinck, M. Donias, S. Guillon, O. Lavialle, Discontinuous seismic horizon tracking based on a poisson equation with incremental dirichlet boundary conditions., in: B. Macq, P. Schelkens (Eds.), ICIP, IEEE, 2011, pp. 3385–3388.

[5] I. Moretti, F. Lepage, M. Guiton, Kine3d: a new 3d restoration method based on a mixed approach linking geometry (2006).

[6] P. Durand-Riard, G. Caumon, P. Muron, Balanced restoration of geological volumes with relaxed meshing constraints, Comput. Geosci. 36 (4) (2010) 441–452.

[7] S. Gibson, B. Mirtich, A survey of deformable modeling in computer graphics, Technical Report TR-97-1 9.

[8] T. Igarashi, T. Moscovich, J. Hughes, As-rigid-as-possible shape manipulation, in: ACM Transactions on Graphics (TOG), Vol. 24, ACM, 2005, pp. 1134–1141.

[9] Y. Weng, W. Xu, Y. Wu, K. Zhou, B. Guo, 2d shape deformation using nonlinear least squares optimization, Vis. Comput. 22 (9) (2006) 653–660.

[10] M. S. Floater, Mean value coordinates, Comput. Aided Geom. Des. 20 (1) (2003) 19–27.