Topological reconstruction of oil reservoirs from seismic surfaces

GEOVAN TAVARES¹, ROGERIO SANTOS², HÉLIO LOPES¹, THOMAS LEWINER¹,³ AND ANTONIO WILSON VIEIRA¹,⁴

¹ Department of Mathematics — Pontifícia Universidade Católica — Rio de Janeiro — Brazil
² Petrobras — Brazilian Oil Company — Rio de Janeiro — Brazil
³ Géométrica Project — INRIA — Sophia Antipolis — France
⁴ CCET — Universidade de Montes Claros — Brazil
{tavares, awilson, lopes, tavares, tomlew, awilson}@mat.puc-rio.br.
r_santos@petrobras.com.br.

Abstract. This paper describes the main aspects of Project Geosis. It is an ongoing three year project between the Brazilian oil company Petrobras and the Pontifical Catholic University of Rio de Janeiro, Brazil. Its main objective is to extract information from seismic data through the use of geometric and topological modeling, as well as scientific visualization.

Keywords: Topological surface reconstruction. Surface simplification.

Figure 1: 3D topological reconstruction, from seismic data, of a deep water oil reservoir (2000 meters).

1 Introduction

The characterization of oil reservoirs has as one of its components the study of their geometry and topology. While the geometry deals with spatial distribution of points and its associated properties, e.g. permeability and porosity, the topology handles its connectivity. To determine the topology of the reservoir is then an essential ingredient for its characterization. One of the main characteristics of seismic processing is the generation of large data sets. Computationally intensive powerful techniques are needed to extract mean-
ingful information from these data. The main objective of
the paper is to indicate how to reconstruct geological objects
(channels, lobes, . . .) directly from seismic data. The data is
piled up in offset groups in such a way as to get information
from the seismic aiming at the geological modeling.

This paper focuses on: 1. Topological surface reconstruction
from seismic data and its corresponding visualization; 2. Surface simplification in order to deal with large data sets.
Techniques for surfaces simplification for handling large
data sets will be discussed. Several techniques have been im-
plemented and will be presented.

This paper describes the main aspects of Project Geosis.
It is an ongoing three year project between the Brazilian oil
company Petrobras and the Pontifical Catholic University
of Rio de Janeiro, Brazil. Its main objective is to extract
information from seismic data through the use of geometric
and topological modeling, as well scientific visualization.

2 Isosurface extraction

(a) Overview

The seismic data obtained by physical measures can be
interpreted, after the signal processing step, as a
sampling of a continuous function \( f : (x, y, z) \in \mathbb{R}^3 \mapsto f(x, y, z) \in \mathbb{R} \) at some points in the space
\( \{(x_i, y_j, z_k) : i = 1..m, j = 1..n, k = 1..p\} \). The function \( f \)
describe a geophysical property, such as the permeability,
the porosity, . . . The reservoir corresponds to the portion of
space \( R \) where the value of the property is included in a
certain range \([v, w] : R = f^{-1}([v, w])\).

Any method that computes this volume \( R \) needs to ex-
trapolate from the sampled data. For example, we could
induce the function \( f \) from the sampled data, and then com-
pute analytically the pre–image \( R \). This extrapolation would
involve a specific and complete geological model of the ter-
rain. However, this method is computationally expensive,
renders interpreted data and does not guarantee the topol-
ogy of the reservoir in general. We will use a different but
classical strategy, and enhance its robustness.

We will compute the reservoir surface directly from the
sampled data. We aim at controlling and minimizing the
artifacts induced by the extrapolation. In particular, we are
concerned with the topology of the resulting surface, i.e.

preserving the connections between or inside the reservoirs.
Moreover, in order to improve the quality of the result,
we will include in our process the global geological information
we already know, such as the data representation described
in the next section.

(b) Incorporation of geological models

We will suppose from now on that the points where the
property \( f \) is estimated are arranged in a cuberille manner.
This means that we are able to connect those points in order
to form adjacent convex solid polytopes with 8 faces.
Actually, if we do not look for the boundary region, there
are several ways of connecting the measure points into a
cuberille grid. For example on an infinite cubic grid, we
could connect the cube with diagonal strips.

Ideally, the cubes should follow the geological structure
of the reservoir. This can be partially achieved by a prepro-
cessing step, identifying the main curves of the geological
points. In particular, submarine reservoir are often locates in
fracture zone, and the reservoir itself often follows the frac-
ture lines. During this preprocessing step, we connect the
measure points in order to form a cuberille grid which fol-

ows those main curves.

This preprocessing offers different advantages. First, the
resulting reservoir surface is more accurate: the isosurface
extracted from the measures \( f \) is tiled within each cube. If
an edge of the grid has one vertex inside the reservoir and
one outside, then the isosurface will intersect this edge. If
the grid were simply parallel to the axes, those edges crossed by
the surface can be arranged in a very steep manner, leading
to less accurate surface. Second, we can rectify the grid, i.e.
moving the grid points to form a cubic grid parallel to the
axis, without modifying the value of the property \( f \) at that
point. Then, we can work in the original grid, which gives a
realistic view of the reservoir, or in the rectified grid, which
often gives a clearer view of the reservoir.

(c) Missing data

The main problem we encountered with the seismic data
is its incompleteness: some of the grid points \((x_i, y_j, z_k)\) do
not have an associated property value \( f(x_i, y_j, z_k) \). Those
points can be isolated, or form entire volumes inside the grid.
We implemented three strategies to handle this deficiency,
which corresponds to different quality/reliability and qual-
ity/computational costs trade–off:

- No interpolation: the grid point is discarded, and none
  of the triangles of the final surface will intersect a
cube containing a discarded point. This ensures a more
  reliable output, but leads to many holes in the surface.
- Linear interpolation: the value of a missing grid point
  is computed as the barycenter of its nearest valid
  grid points.
- Radial–basis interpolation: all the valid point of the
  grid contributes to the missing value proportionally
  to their distance. This gives nice results, but induces
  a model of the data which is not always accurate.
  Radial–basis methods have been extensively studied,
  and would offer many possibilities of including accu-
  rate geo-physics models [1].

(d) Surface reconstruction

We will use an extension of the Marching–Cubes’ algo-

rithm [3] to extract the surface of the reservoir from the
preprocessed seismic data. The Marching Cubes method
produces a triangular mesh of the preimage \( g^{-1}(0) \), given
by samples over a cuberille grid. To convert the test
\( f(x_i, y_j, z_k) \in [v, w] \) into \( f(x_i, y_j, z_k) \geq 0 \), we will test a
grid point with \( g = (f - v) \cdot (w - f) \). The original method

The corresponding work was published in the proceedings of the [International Association for Mathematical Geology 2003]
sweeps the grid, and tiles the surface cube per cube. Each point of the grid is classified into positive and negative vertices. Thus, there are $2^8 = 256$ possible configurations of a cube. The usual implementation stores those 256 in a lookup table that encodes the tiling of the cube in each case (see Figure 3).

However, the original implementation can lead to cracks (see Figure 2) and could not respect the topology of the tri-linear interpolation of $f$. We thus need to add further test on ambiguous cubes. This distinction has been done by Chernyaev in its Marching Cubes’ 33 algorithm [2], and lead to the completed lookup table of 730 cases.

This enhanced algorithm has been implemented in [7]. However, their implementation needs a re-computation when tiling an ambiguous cube, which is computationally expensive. Their computation allows a more accurate geometry inside each cube by the addition of extra vertices inside some of the cubes. For our purpose, we needed a faster algorithm which would not produce too many vertices. Therefore, we implemented the 730 cases of Chernyaev’s lookup table [3][5]. The tiling of a cube is then done by only a consultation of the lookup table.

The Marching Cubes’ algorithm offer many extensions, in particular in previsualization, view-dependant rendering, hardware acceleration ... Some of those enhancements need an expensive pre-computation ($n \cdot \log(n)$ for topology pre-visualization for example), other pre-suppose parts of the results (the ability of having one seed per connected component in order to avoid parsing all cubes). All those extensions are possible to implement with our enhancements. We will not introduce them here as they are not always adapted to our problem.

(e) Visualization of seismic surfaces

Given a seismic cube, parallel slices are extracted and piled up in such a way as to allow the 3D reconstruction of the reservoir using the amplitude of the seismic wave in a given interval. In the figures below domains on parallel slices were chosen. In Figure 3 and Figure 4, isosurfaces were extracted and geological structures like channels and lobes, were identified in a deep water reservoir.

3 Surface Simplification

The surface reconstruction discussed in the previous section will generate a triangular mesh that represents the boundary of the oil reservoir. According to the sampled data resolution, the data set for handling such mesh may be extremely large. Then, in order to represent the same surface with lower data cost, we have developed a simplification process to obtain a simplified mesh with lower number of triangles that preserves the topology type of the original surface and uses an Error Metric in order to control its geometrical proprieties, as volume, by minimizing the geometrical distortion from the original surface.

Our simplification algorithm is based on local topology preserving operators so that the simplified mesh has the same topology type to the original mesh. The main objective of these operators is removing elements as vertices, edges and faces from the mesh without changing its topologic type. Each element to be removed from the mesh is chosen as the one with lower geometrical cost to the original surface. This cost is computed using the Quadric Error Metric by Garland and Heckbert [4]. We will point out how does this operators and metric works before discussing the simplification algorithm.

(a) Mesh simplification operators

The operators we will present were implemented using a data structure called Corner-Table [6] that is a compact version of the Half-Edge representation of triangular meshes. Using this data structure we implemented several local operators and techniques for mesh simplifications [9] as follow.

Edge–Flip: This operation does not remove any element from the mesh and consists in transforming a two-face cluster into another two-face cluster by swapping its common edge.

Let the edge $e = (u, v)$ and $s$ and $t$ the two vertices opposed to $e$ (see Figure 4). The Edge-Flip operation will replace $e$ by $(s, t)$, and replace the 2 triangles incident to $e$ by $(u, s, t)$ and $(v, t, s)$.

Edge–Collapse: This operator consists on removing an edge $e = (u, v)$ from the mesh, identifying its vertices to a unique vertex $w$. From a combinatorial viewpoint, this operator will remove 1 vertex, 3 edges and 2 faces from original
Figure 3: Slices of seismic data of a deep water oil reservoir (2000 meters).

mesh, without changing its Euler characteristic. From a geometric viewpoint, the new position of the vertex $w$ can be computed with the geometry around $u$ and $v$.

![Figure 5: Edge-Collapse.](image)

Let $s$ and $t$ be the vertices opposite to $e = (u, v)$, which is the edge to be collapsed (see Figure 5). By collapsing the edge $e$ we will remove the faces $(u, v, s)$ and $(u, t, v)$ from the mesh. The topological consistency of this operations is guaranteed by the following link condition [3].

*Link Condition* Let $S$ be a combinatorial 2–manifold. The contraction of an edge $e = (u, v) \in S$ preserves the topology of $S$ if and only if $\text{link}(u) \cap \text{link}(v) = \text{link}(e)$.

Figure 6 shows, in (a), an edge $(u, v)$ whose contraction is topology preserving while in (b) it is not.

Other operators as edge-weld, face-collapse and star-collapse were implemented as a combination of these two first operators.

The corresponding work was published in the proceedings of the International Association for Mathematical Geology 2003.
(b) Geometric cost evaluation

For evaluation of the geometrical cost of performing simplification operators we used the Quadric Error Metric from Garland and Heckbert \[4\] as follow.

Consider the problem of finding the distance of a point \( w \) to the plane \( p_f \) support of a face \( f = (v_1, v_2, v_3) \) whose unitary normal vector is \( \vec{n} \).

Given a point \( p = (p_x, p_y, p_z) \in p_f \), the distance \( d \) from the origin to the plane \( p_f \) is

\[ d = p \cdot \vec{n} = p_x n_x + p_y n_y + p_z n_z \]

Then, the distance from a point \( w \in \mathbb{R}^3 \) to \( p_f \), is the distance \( d_w \) from the origin to the plane parallel to \( p_f \) touching \( w \) minus \( d \).

\[ d_w = w \cdot \vec{n} - d = w_x n_x + w_y n_y + w_z n_z - d \]

Making \( w = (w_x, w_y, w_z, 1) \) and \( \vec{n} = (n_x, n_y, n_z, -d) \), \( d_w \) can be computed as a dot product in dimension 4.

\[ d_w = w \cdot \vec{n} \left( n_x n^t + w_x n_x + w_y n_y + w_z n_z + 1(-d) \right) \]

The quadratic distance \( d(w) \) from a point \( w \in \mathbb{R}^3 \) to the plane \( p_f \) is, than, given by

\[ d(w) = (w \cdot \vec{n})^2 = (w \cdot n)^t w = \bar{w}^t (n, n^t) w \]

The product \( n, n^t \) above origins a 4x4 symmetric matrix, which Garland and Heckbert called the fundamental quadric \( Q_f \) associated to the face \( f \).

\[ Q_f = n, n^t = \begin{pmatrix} n_x^2 & n_x n_y & n_x n_z & n_x d \\ n_x n_y & n_y^2 & n_y n_z & n_y d \\ n_x n_z & n_y n_z & n_z^2 & n_z d \\ n_x d & n_y d & n_z d & d^2 \end{pmatrix} \]

Let \( v \) a vertex on a mesh \( M \), and \( f_i \in \text{star}(v) \) the faces incidents to \( v \). The distance \( d(w, v) \) from a point \( w \) to the vertex \( v \) is given, by the Quadric Error Metric, as the sum of the quadratic distances \( d_i(w) \) from \( w \) to the plane support of each face \( f_i \). Using the quadrics \( Q_i \) associated with each face \( f_i \) we have

\[ d(w, v) = \sum u^t (Q_i) u = w^t (\sum Q_i) w \]

The sum \( \sum Q_i \) origins a new 4x4 matrix called the fundamental quadric associated with the vertex \( v \), which is noted as \( Q_v \).

On performing the edge-collapse operation for an edge \( e = (u, v) \) to a resultant vertex \( w \) we have the cost \( C \) (geometric distortion) given by the sum of the distances \( d(w, v) \) and \( d(w, u) \)
integers for representing the connectivity at each level. This connectivity can also be encoded in a compressed manner (with less than 2 bits per triangle) using the Edgebreaker compression scheme [6].

Figure 7: Original Marching Cubes lookup table.

References


Figure 8: Chernyaev’s lookup table.

Preprint MAT. 22/03, communicated on May 7th, 2003 to the Department of Mathematics, Pontifícia Universidade Católica — Rio de Janeiro, Brazil.