



**IDENTIFICACIÓN DE REFLECTORES EN IMÁGENES DE SECCIÓN SÍSMICA
APLICANDO MÉTODOS ESTADÍSTICOS Y MÉTODOS SINTÁCTICOS**

CARLOS ANDRÉS PÉREZ SOLANO

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS
ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y
TELECOMUNICACIONES
BUCARAMANGA**

2008



**IDENTIFICACIÓN DE REFLECTORES EN IMÁGENES DE SECCIÓN SÍSMICA
APLICANDO MÉTODOS ESTADÍSTICOS Y MÉTODOS SINTÁCTICOS**

CARLOS ANDRÉS PÉREZ SOLANO

**Trabajo de grado presentado como requisito parcial para optar al título de
Ingeniero Electrónico**

Director

CESAR ANTONIO DUARTE GUALDRÓN, MSc

Codirectores

GERMÁN OJEDA, PhD

RAFAEL ISAACS

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS
ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y
TELECOMUNICACIONES
BUCARAMANGA**

2008

AGRADECIMIENTOS

A la Universidad Industrial de Santander y al Instituto Colombiano del Petróleo por facilitar los medios para la creación y sostenimiento de los grupos de investigación. Al profesor Carlos Piedrahita por su compromiso con el crecimiento del grupo Petrosísmica. A mis directores Cesar Duarte, Germán Ojeda y Rafael Isaacs por el apoyo brindado.

A mis padres Francisco y Luz Mar,
y a mi hermano Holguer Fernando
por apoyarme siempre.

A mis amigos
por plantearme siempre nuevos retos.

Carlos Andrés Pérez Solano

CONTENIDO

RESUMEN	12
ABSTRACT	13
INTRODUCCIÓN	14
OBJETIVOS	16
OBJETIVO GENERAL	16
OBJETIVOS ESPECÍFICOS	16
1. EL MÉTODO GEOFÍSICO: SÍSMICA	17
1.1. Teoría básica del método sísmico	17
1.2. Adquisición de datos sísmicos	19
1.3. Procesamiento de datos sísmicos	22
1.4. Interpretación de datos sísmicos	24
1.4.1. Reflectores discontinuos	26
2. RECONOCIMIENTO DE PATRONES	28
2.1. Reconocimiento de patrones por métodos estadísticos	28
2.1.1. Coeficiente de Coherencia.....	29
2.1.2. Coeficiente de semblanza.....	31
2.2. Reconocimiento de patrones por métodos sintácticos	33
2.2.1. Reconocimiento sintáctico usando autómatas de estado finito.....	34
2.2.2. Reconocimiento sintáctico usando criterio de mínima distancia entre cadenas	40
3. DISEÑO DE LOS ALGORITMOS	43
3.1. Algoritmo de coherencia.....	43
3.1.1 Núcleo del algoritmo	43
3.1.2. Identificación con coherencia.....	45

3.2. Algoritmo de semblanza	47
3.2.1. Núcleo del Algoritmo.....	48
3.2.2. Identificación con semblanza.....	48
3.3. Algoritmo con Autómatas	49
3.3.1. Inferencia del autómata de clasificación de patrones	51
3.3.2. Identificación con autómatas	58
3.4. Algoritmo de Levenshtein	59
3.4.1. Núcleo del algoritmo	60
3.4.2. Identificación usando criterio de Levenshtein	61
4. EVALUACIÓN DEL DESEMPEÑO DE LOS ALGORITMOS.....	65
4.1. Conjunto de datos	65
4.2. Elección de los parámetros de entrada	69
4.3. Resultados en datos sintéticos.....	70
4.3. Prueba de rapidez computacional	80
4.4. Resultados en datos reales	82
5. NUEVAS APLICACIONES DE LOS ALGORITMOS	86
5.1. Aplicación en identificación de primeros arribos.....	86
5.2. Aplicación en corrección de velocidades por NMO	88
5.3. Aplicación en identificación de curvas de emisividad	89
6. CONCLUSIONES Y RECOMENDACIONES.....	92
6.1. Conclusiones.....	92
6.2. Recomendaciones.....	94
6.3. Trabajo Futuro.....	96
REFERENCIAS	98

ANEXO. MANUAL DE LA HERRAMIENTA COMPUTACIONAL *ARI*..... 102

LISTA DE FIGURAS

Figura 1. Ondas generadas por la fuente, reflejadas y registradas en los sensores	18
Figura 2. Imagen de disparo en una sísmica terrestre	21
Figura 3. Clases de ondas con igual tiempo de viaje.	22
Figura 4. Flujo característico de procesamiento en sísmica.....	23
Figura 5. Imagen procesada de una línea sísmica del Campo Escuela de la UIS.	24
Figura 6. Imagen de sección sísmica.....	25
Figura 7. Imagen sísmica con horizontes interrumpidos por fallas geológicas.	26
Figura 8. Diagrama de bloques de un sistema estadístico de reconocimiento de patrones..	29
Figura 9. Coherencia entre dos señales aplicando desfase.....	31
Figura 10. Conjunto o ventana de trazas en el cual se evalúa semblanza.	32
Figura 11. Diagrama de bloques de un sistema sintáctico de reconocimiento de patrones .	34
Figura 12. Sistema de clasificación de muestras usando gramática de estado finito	35
Figura 13. Señal de evaluación y patrón de entrada.....	36
Figura 14. Autómata determinista de estado finito que representa una GEEF con 91 reglas de producción, 5 estados (símbolos no terminales) y 9 símbolos terminales.	38
Figura 15. Sistema de clasificación usando mínima distancia medida entre cadenas	40
Figura 16. Señales de entrada para los algoritmos de identificación.	44
Figura 17. Señales guía o patrones en una imagen determinada.....	47
Figura 18. Hipermatriz de costos para códigos de Freeman.	56
Figura 19. Matriz de recorrido para hallar distancia entre cadenas	60
Figura 20. Imagen sísmica sintética con 3 reflectores primarios	66
Figura 21. Imagen sísmica sintética con 5 reflectores primarios	66
Figura 22. Imagen sísmica sintética con 3 reflectores primarios	67

Figura 23. Imagen sísmica sintética con 2 reflectores primarios	67
Figura 24. Imagen sísmica sintética con 2 reflectores primarios	68
Figura 25. Imagen sísmica de Boonsville con 2 reflectores primarios	68
Figura 26. Imagen sísmica sintética con 2 reflectores primarios	69
Figura 27. Resultado de elegir más de un reflector como parámetro de entrada	70
Figura 28. Resultado de rapidez de los algoritmos	82
Figura 29. Línea 1440 del Campo Escuela interpretada por un experto	83
Figura 30. Reflectores identificados por tramos usando Semblanza y Levenshtein.....	83
Figura 31. Primeros arribos identificados en una imagen de disparo.	87
Figura 32. Identificación de un reflector en una imagen para NMO	89
Figura 33. Nueva aplicación en curvas de emisividad de gases	90
Figura 34. Curva identificada en la grafica de emisividad.....	90
Figura 35. Cargar una imagen en la herramienta <i>ARI</i>	102
Figura 36. Imagen sísmica cargada en la interfaz <i>ARI</i>	103
Figura 37. Interfaz <i>SelReflector</i> para seleccionar la ventana del reflector deseado.....	104
Figura 38. Ventana seleccionada limitada por líneas negras en la interfaz <i>ARI</i>	105
Figura 39. Selección del algoritmo en la interfaz <i>ARI</i>	106
Figura 40. Imagen con el resultado encontrado por <i>ARI</i>	106

LISTA DE TABLAS

Tabla 1. Costos de producción usando el autómata de la Figura 14.....	39
Tabla 2. Codificación de Freeman normalizada.....	51
Tabla 3. Costo de sustitución con símbolos terminales de Freeman.....	54
Tabla 4. Resultados en las fallas verticales de la Figura 20.....	71
Tabla 5. Resultados en las fallas con inclinación positiva de la Figura 20.....	72
Tabla 6. Resultados en las fallas con inclinación negativa de la Figura 20.....	73
Tabla 7. Resultados en las fallas verticales de la Figura 21.....	74
Tabla 8. Resultados en las fallas con inclinación positiva de la Figura 21.....	74
Tabla 9. Resultados en las fallas verticales de la Figura 22.....	75
Tabla 10. Resultados en la falla con inclinación negativa de la Figura 22.....	75
Tabla 11. Resultados en las fallas verticales de la Figura 23.....	76
Tabla 12. Resultados en la falla con inclinación positivo de la Figura 23.....	76
Tabla 13. Resultados en la falla con inclinación negativa de la Figura 23.....	77
Tabla 14. Resultados en la falla con inclinación negativa de la Figura 24.....	77
Tabla 15. Resultados en las fallas con inclinación positiva de la Figura 25.....	78
Tabla 16. Resultados en las fallas con inclinación negativa de la Figura 25.....	78
Tabla 17. Resultados de los algoritmos ante distintos tipos de fallas.....	79
Tabla 18. Resultados totales de los algoritmos en imágenes sintéticas.....	79
Tabla 19. Resultados de rapidez de los algoritmos.....	81
Tabla 20. Parámetros de entrada en identificación de reflectores sobre la línea 1440 del Campo Escuela.....	83

RESUMEN

TITULO:

IDENTIFICACIÓN DE REFLECTORES EN IMÁGENES DE SECCIÓN SÍSMICA APLICANDO MÉTODOS ESTADÍSTICOS Y MÉTODOS SINTÁCTICOS.*

AUTOR:

CARLOS ANDRÉS PÉREZ SOLANO**

PALABRAS CLAVES:

Tratamiento de imágenes sísmicas, reconocimiento de patrones, métodos sintácticos, métodos estadísticos, autómatas de estado finito, coherencia, programación de algoritmos, método sísmico.

DESCRIPCIÓN:

En interpretación de datos sísmicos, la identificación de reflectores constituye el primer paso a realizar. La identificación de los reflectores primarios permite a los expertos inferir estructuras geológicas en el subsuelo del lugar donde se hizo la adquisición. Esta información, unida a otros datos, permite presentar pruebas sobre la presencia de hidrocarburos.

Los métodos estadísticos (coeficiente de coherencia y coeficiente de semejanza) y los métodos sintácticos (autómatas de estado finito y criterio de Levenshtein) para reconocimiento de patrones mostraron resultados satisfactorios en la identificación semiautomática de reflectores. Los algoritmos diseñados permitieron seguir reflectores continuos. En reflectores con discontinuidades como fallas verticales e inclinadas los algoritmos diseñados permitieron seguir la trayectoria de aquellos en los cuales la discontinuidad no presentó un desplazamiento mayor al espesor del reflector elegido. Los métodos estadísticos mostraron una rapidez en crecimiento linealmente proporcional al número de muestras del reflector patrón, mientras que los métodos sintácticos mostraron una rapidez en crecimiento exponencial.

Los métodos diseñados constituyen una herramienta útil en la identificación semiautomática de reflectores. Los algoritmos ayudarán a los expertos en la tarea interpretativa de datos sísmicos, quienes podrán encontrar resultados con calidad de acuerdo a los parámetros que sean ingresados a la entrada de cada función.

* TRABAJO DE GRADO

** FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS. ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y DE TELECOMUNICACIONES. Programa de Ingeniería Electrónica. Director César Antonio Duarte Gualdrón.

ABSTRACT

TITLE:

REFLECTOR IDENTIFICATION IN SEISMIC SECTION IMAGES APPLYING STATISTICS METHODS AND SYNTACTIC METHODS.*

AUTHOR:

CARLOS ANDRÉS PÉREZ SOLANO**

KEYWORDS:

Seismic images processing, pattern recognition, statistics methods, syntactic methods, finite state automata, coherency, algorithms programming, seismic method.

DESCRIPTION:

In seismic interpretation, identification of reflectors is the first step to do. Primary reflectors identification leads the experts to infer geologic structures in the land where the acquisition was made. That information, along with other data sets, is sufficient to conclude about the presence of hydrocarbons.

Statistics methods (coherence coefficient and semblance coefficient) and syntactic methods (finite-state automaton and Levenshtein distance) for pattern recognition demonstrated satisfactory results on semiautomatic reflector identification. The algorithms were designed for reflectors without discontinuities. In reflectors with discontinuities, like vertical falls and inclined falls, the algorithms were able to follow reflectors trajectory in low displacement falls. Computational speed in statistics methods have a lineally performance, but syntactic methods have exponential performance.

The methods here designed perform as a good tool in semiautomatic identification of reflectors. The algorithms will help experts to find satisfactory results by only selecting well input parameters.

* TRABAJO DE GRADO

** FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS. ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y DE TELECOMUNICACIONES. Programa de Ingeniería Electrónica. Director César Antonio Duarte Gualdrón.

INTRODUCCIÓN

En la actualidad, el mundo experimenta un proceso de desarrollo que demanda cantidades cada vez mayores de energía. En gran parte, los hidrocarburos cubren esta demanda y lo seguirán haciendo al menos hasta la primera mitad de este siglo. Por tal motivo, las inversiones en exploración y producción petrolífera, deben estar acompañadas de métodos cada vez más eficientes que justifiquen estos procesos.

Pensando en la automatización de los procesos actuales de identificación de reflectores en imágenes sísmicas se han planteado métodos estadísticos para reconocimiento de patrones, incluyendo redes neuronales, cubo de coherencia, lógica difusa y algoritmos genéticos, entre otros [3], [8], [13], [20].

Además del enfoque estadístico, también existen métodos para reconocimiento sintáctico de patrones. Los métodos sintácticos son basados en la teoría de lenguajes formales y autómatas. Esta teoría es ampliamente utilizada en problemas de reconocimiento estructural de patrones. Entre ellos, se involucran problemas en los que se desean analizar estructuras como reconocimiento de caracteres, reconocimiento de huellas dactilares, análisis de texturas, entre otras [5], [21], [22].

Sin embargo, en la mayoría los trabajos citados no se observa un flujo claro para la identificación semiautomática de reflectores. En este trabajo se plantean métodos que analizan una imagen sísmica y brindan al geólogo una herramienta útil en su búsqueda por una correcta interpretación sísmica. Se ha estudiado el comportamiento y los alcances de los métodos estadísticos basados en coeficiente de coherencia temporal, coeficiente de semblanza y los métodos sintácticos basados en autómatas de estado finito y criterio de distancia entre cadenas de Levenshtein. El comportamiento es adecuado en reflectores continuos

y en reflectores discontinuos para los cuales el reflector no se desplaza una distancia mayor a su espesor.

El método geofísico “Sísmica” es explicado en detalle en el *capítulo 1*. Allí se observan los pasos que son llevados a cabo como prueba de existencia de hidrocarburos en determinado lugar del subsuelo. Una vez comprendido el método sísmico es mostrada la teoría que soportó el diseño de los algoritmos. Esta teoría comprende los métodos para reconocimiento de patrones de forma estadística y sintáctica que son mostrados en el *capítulo 2*.

El planteamiento y diseño de los algoritmos en pseudo código es el contenido del *capítulo 3*. Estos algoritmos fueron programados en una herramienta utilizada para analizar imágenes sísmicas apiladas generadas de forma sintética e imágenes reales de zonas de exploración en Colombia. Los resultados generados son mostrados en el *capítulo 4*. Además en este capítulo se presenta un análisis del comportamiento de los algoritmos en reflectores que presentan discontinuidades geológicas.

En el desarrollo del trabajo se encontró que el seguimiento de reflectores no sólo es aplicable en interpretación de datos sísmicos, por esta razón se presenta un capítulo adicional en cual se muestran algunas otras aplicaciones en sísmica. De esta forma el *capítulo 5* contiene algunas propuestas para aplicar los métodos en procesamiento de datos sísmicos.

Finalmente se plantean las conclusiones y recomendaciones del trabajo en el *capítulo 6*. Estas servirán como guía para la continuación de la investigación en trabajos futuros que incluirán posiblemente herramientas para interpretación semiautomática total o parcial de las zonas de exploración de hidrocarburos.

OBJETIVOS

OBJETIVO GENERAL

Estudiar el alcance de los métodos estadísticos y sintácticos en el seguimiento de horizontes continuos y discontinuos programando una herramienta de software que ayude a la interpretación semiautomática de datos sísmicos en exploración de hidrocarburos.

OBJETIVOS ESPECÍFICOS

- ✓ Comparar el desempeño de los métodos en estudio para reconocer y caracterizar reflectores continuos y discontinuos en líneas sísmicas.

- ✓ Desarrollar una herramienta computacional que aplique métodos sintácticos y estadísticos en datos de sección sísmica generados sintéticamente y en datos reales tomados de la zona en estudio del Piedemonte en Colombia.

1. EL MÉTODO GEOFÍSICO: SÍSMICA

La exploración terrestre y oceánica usando métodos Geofísicos ha tenido una profunda influencia en el entendimiento de la Tierra y su comportamiento a través del tiempo [4]. El desarrollo de nuevas tecnologías ha permitido el rápido desarrollo de la Geofísica y de otras ramas de la ciencia. Por tal razón, en la actualidad se apoya el planteamiento de nuevas teorías, que mediante el uso de herramientas computacionales, permitirán seguir nuevos horizontes en la búsqueda del progreso científico.

1.1. Teoría básica del método sísmico

Sísmica es un método geofísico cuyo objetivo es reconocer la estructura del subsuelo en el lugar donde ésta se realice. En palabras simples, las imágenes sísmicas son 'ecografías' del subsuelo. El método sísmico se basa en la reflexión y refracción de ondas debidas a los cambios de densidad de las rocas y de velocidad de propagación entre las diferentes capas terrestres. En campo, los componentes de un sistema de adquisición sísmica incluyen fuentes de energía y sensores de presión en la superficie que registran la forma de onda luego de ser reflejadas en capas del subsuelo de diferentes litologías. Dicha señal es amplificada y registrada digitalmente para luego ser procesada y presentada en una forma interpretable en términos geológicos de estructura, estratigrafía y a veces de contenido de fluidos [1].

En el método sísmico se pueden analizar las ondas debidas a refracción o reflexión. Para el estudio actual se trabaja con sísmica de reflexión, debido a que los objetivos son orientados a la identificación de reflectores. Este consiste en medir los tiempos de llegada de eventos atribuibles a reflexión de las ondas en interfaces donde se presentan cambios en la impedancia acústica (Figura 1). Identificar tales interfaces conduce a la descripción de la estratigrafía y permite inferir la estructura del lugar. El éxito en la identificación de las estructuras del

subsuelo depende de la efectividad con que se realicen los pasos fundamentales: *adquisición, procesamiento e interpretación* [1], [28].

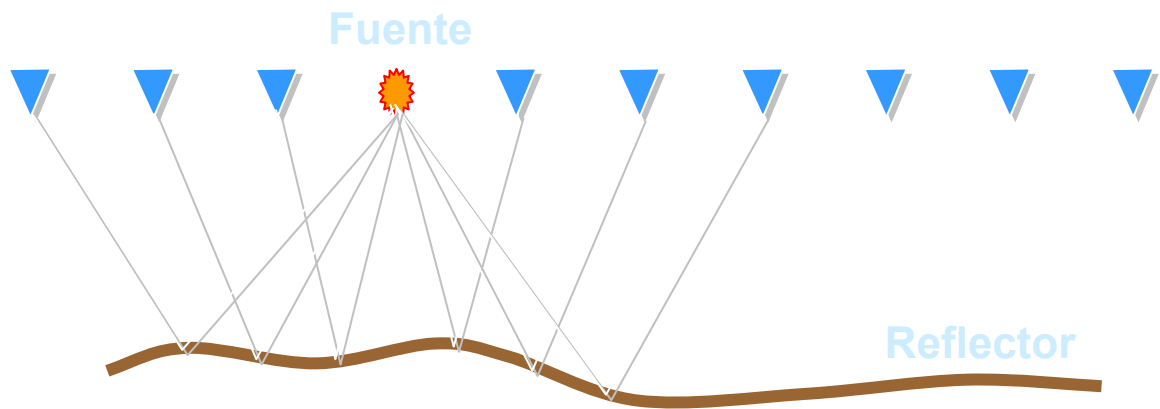


Figura 1. Ondas generadas por la fuente, reflejadas y registradas en los sensores

En la adquisición, los sensores deben estar calibrados para no introducir ruido del dispositivo, la ubicación de las fuentes y de los sensores debe ser planeada con anterioridad para generar los resultados esperados. En procesamiento, se aplican diferentes técnicas a los datos registrados que llevan a presentar imágenes compuestas de trazas sísmicas en 2D o en 3D. Las técnicas que se aplican se orientan a realizar las siguientes correcciones [1]:

1. Correcciones en tiempo: Estáticas o dinámicas.
2. Correcciones en amplitud: Se corrige el decaimiento en amplitud.
3. Correcciones de distorsiones: Múltiples, fantasmas, etc.
4. Correcciones de posición de los eventos: Demultiplexado.

En la interpretación, los geólogos y geofísicos tienen como objetivo dar un significado estructural de la zona de estudio. En este importante paso son requeridos mapas sísmicos, mapas estructurales y en la actualidad estaciones computarizadas de interpretación. Finalmente y luego de analizar una gran cantidad de líneas sísmicas los interpretes pueden asegurar la existencia de un pozo contenedor de aceite o minerales de importancia [1].

1.2. Adquisición de datos sísmicos

La exploración sísmica es llevada a cabo usando lo siguiente [4]:

1. Una fuente de energía sísmica.
2. Objetos para la detección de ondas elásticas viajando a través de columnas de agua (sísmica marina) o capas someras de tierra (sísmica terrestre).
3. Un método para la grabación e ilustración de ondas sísmicas en una forma legible e interpretable.

La fuente de ondas sísmicas ideal es aquella que produce un solo pulso de gran energía que sea fácilmente detectable en presencia de ruido después de haber viajado hasta la zona más profunda de tierra que requiera ser estudiada. Esta señal de salida cubre un rango infinito de frecuencias; puede ser usada para discriminar interfaces cercanamente espaciadas en capas de tierra a poca profundidad, para lo cual son necesarias las frecuencias altas, y para sísmica profunda probando bajas frecuencias. La naturaleza de los transductores, la absorción de la onda elástica dependiente de la frecuencia y la presencia de fronteras de propiedades entre los medios hacen que la forma de onda a la salida sea más compleja que un simple pulso [4].

Las fuentes sísmicas son divididas en dos: *las impulsivas*, que producen pulsos de corta duración, y las de *barrido en frecuencia*, que producen una señal de baja amplitud, aproximadamente sinusoidal y progresivamente varía su frecuencia. La elección en la fuente de energía depende de la resolución sísmica y la penetración requerida.

Las ondas sísmicas son detectadas por las variaciones en la presión que producen dentro de las columnas de agua o la tierra. Las forma de ondas generadas son captadas por hidrófonos o geófonos, estos sensores son utilizados en sísmica marina y terrestre respectivamente [1]. Por lo general se trata de

piezoeléctricos conectados a una bobina y una resistencia. Los elementos para grabado de ondas son ordenados en arreglos lineales (para sísmica 2D) o en forma de cuadrícula (para sísmica 3D).

La grabación de ondas final se presenta en forma digital gracias al sistema de los sensores que además de detectar la señal, también incluyen un sistema de digitalización y filtrado con los siguientes componentes [4]:

(i) Preamplificadores y filtros: las señales capturadas son primero amplificadas y luego filtradas analógicamente. Un filtro de pasabajos para eliminar la interferencia generada por el movimiento de las olas. Un filtro pasaltos antialiasing para prevenir el sobremuestreo en la etapa de digitalización. Y frecuentemente un filtro notch para atenuación de ruido característico de 50 o 60 Hz (interferencia de la línea alimentadora).

(ii) Unidad de Multiplexado: es una unidad electrónica trabajando a gran velocidad con objetivo de enviar la señal a un amplificador digital y al convertidor analógico-digital.

(iii) Amplificador de ganancia: amplifica la señal de entrada y graba esta ganancia digitalmente con el objetivo de recuperar la ganancia original en etapas posteriores.

(iv) Convertidor analógico-digital: cambia de formato la señal y digitaliza por lo general a 24 bits.

(v) Grabadora digital: la señal digital obtenida es grabada en un formato estándar como SEG-D de 4 bytes. La presentación en una cinta magnética de 13 mm dentro de un cartucho compacto.

(vi) Demultiplexador: demultiplexa la señal y le da el formato internacional SEG-Y, para su posterior uso en procesamiento.

(vii) Unidades de graficación: unidades de visualización de los datos, usualmente pueden ser estaciones de trabajo conectadas a un *cluster*.

Por cada disparo de la fuente los sensores y el sistema asociado entran en funcionamiento hasta obtener una imagen de disparo como la mostrada en la Figura 2. En la imagen se observa una grabación de aproximadamente 7000 muestras que pueden equivaler a 7 segundos según la tasa de muestreo. Cada columna en la imagen corresponde a la grabación realizada por un geófono a la vez (Fuente: ECOPETROL – ICP). Una gran cantidad de imágenes de disparo son necesarias para obtener una sola imagen sísmica apilada. Los pasos necesarios para convertir estas imágenes iniciales en imágenes interpretables son conocidos como procesamiento de datos sísmicos.

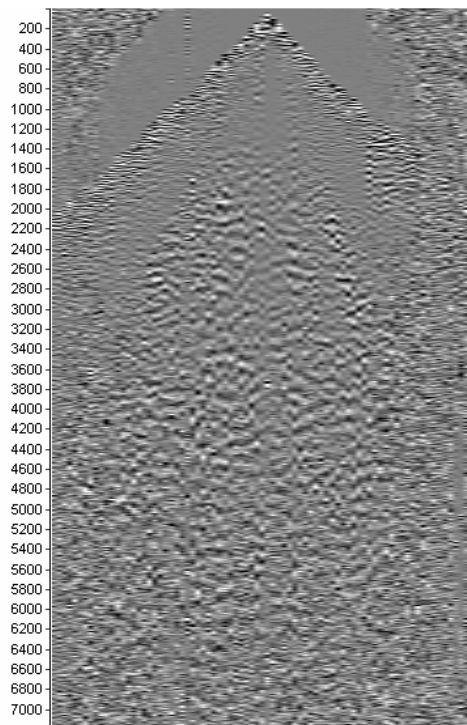


Figura 2. Imagen de disparo en una sísmica terrestre

1.3. Procesamiento de datos sísmicos

Los registros de disparos obtenidos en la adquisición deben ser procesados para obtener imágenes sísmicas interpretables. El objetivo principal del procesamiento de datos es incrementar la señal reflejada en comparación con el ruido de fondo. Usualmente esta medida se denomina relación señal a ruido **SNR**. Los eventos considerados como señal son únicamente los de interés geológico y geofísico, estos son: las reflexiones primarias, pues ayudan en la definición de capas y las difracciones que ayudan a determinar discontinuidades [24]. En el registro de disparo usualmente se tiene la reflexión primaria acompañada de una serie de ondas no necesarias, consideradas como ruido. En la Figura 3 se observan algunas de estas ondas.

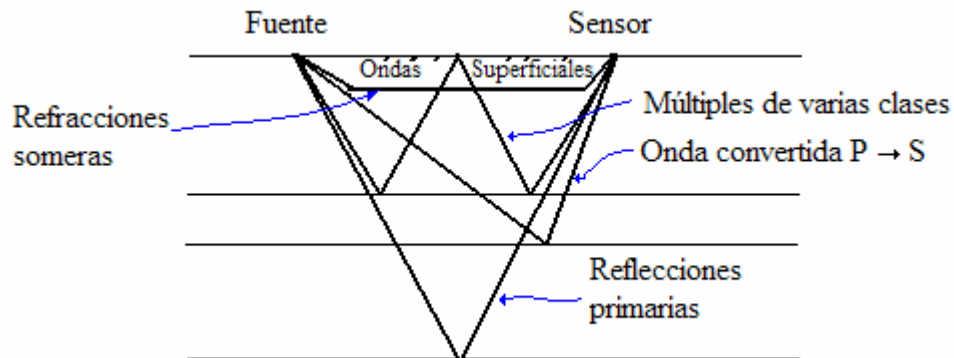


Figura 3. Clases de ondas con igual tiempo de viaje.

Los procesos que deben aplicarse al conjunto de sismogramas de disparo deben estar orientados a cumplir el objetivo principal del procesamiento. Estos procesos se organizan en un flujo que debe contener dentro de sí mismo algunas iteraciones que conduzcan a mejores resultados, luego de realizar pruebas a los datos y estudiar las propiedades del subsuelo, como presencia de anisotropía, densidad de fallas, velocidad de propagación, entre otras. El flujo de procesamiento varía de acuerdo a las necesidades de cada zona en estudio, dependiendo de sus características geológicas y estructurales. De igual forma existen procesos que deben ser realizados de forma obligatoria, pues de otro

modo no se llegaría a un correcto procesamiento. En la Figura 4 se observa un diagrama de flujo característico para procesamiento de líneas sísmicas (2D). Los procesos encerrados en rectángulos se realizan en forma obligatoria, mientras que los encerrados en elipses se realizan de acuerdo a la necesidad [4].

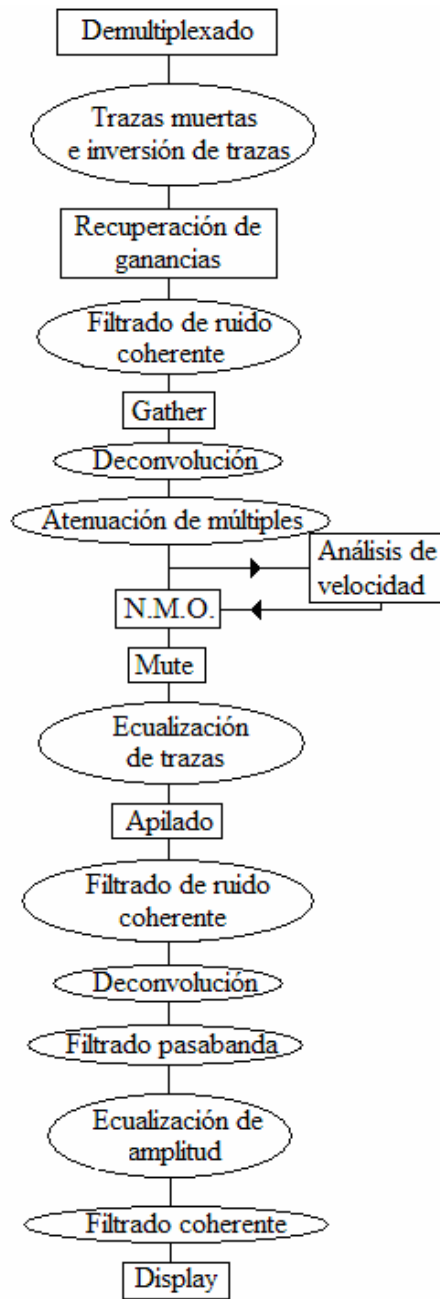


Figura 4. Flujo característico de procesamiento en sísmica.

Las imágenes de disparo, como la mostrada en la Figura 2, son convertidas en imágenes más claras del subsuelo después de aplicar el flujo de procesamiento. En la Figura 5 se muestra la imagen de la línea sísmica 1440 del campo Escuela de la UIS luego de ser procesada. El eje vertical está en milisegundos y el horizontal en número de trazas. Esta imagen esta lista para ser interpretada, aunque en la actualidad y con el apoyo de la UIS y el ICP estas líneas serán reprocesadas para obtener imágenes mas claras.

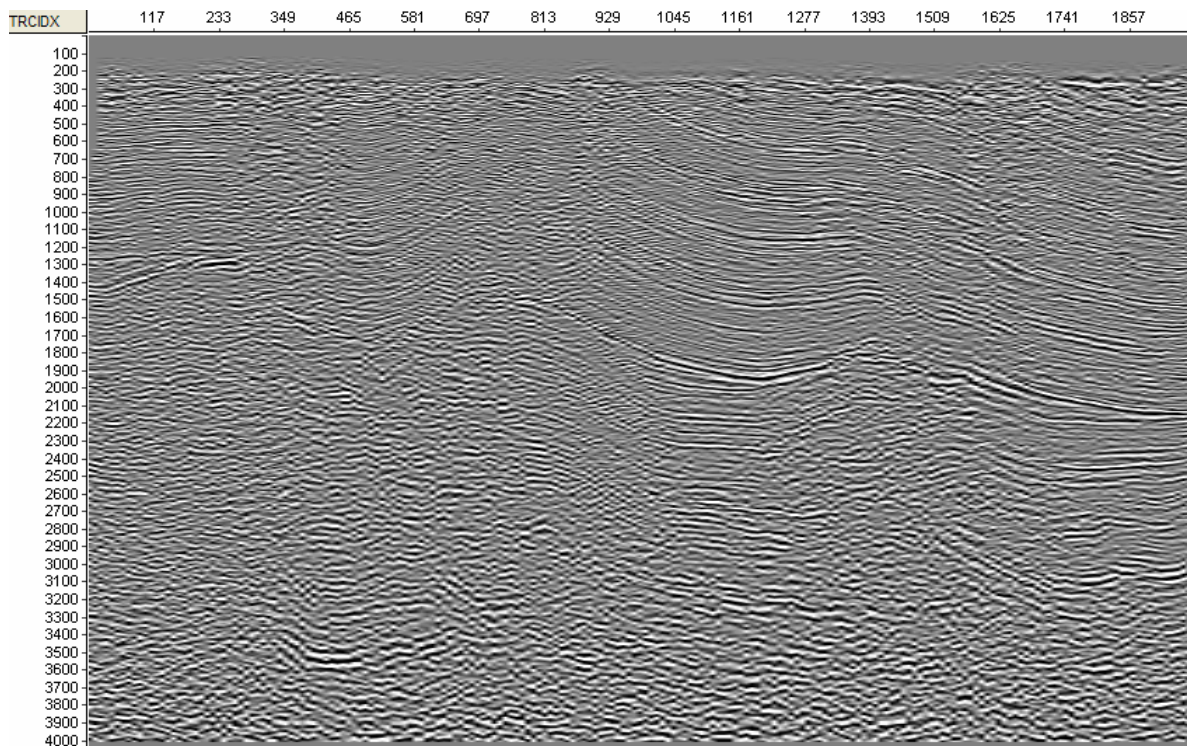


Figura 5. Imagen procesada de una línea sísmica del Campo Escuela de la UIS.

1.4. Interpretación de datos sísmicos

La versión interpretada de datos sísmicos es el objetivo geológico y producto final del trabajo sísmico. Ésta implica la determinación del significado geológico de los datos sísmicos e incluye la reducción de datos, la selección de eventos que se toman por reflexiones primarias, y la localización de los reflectores con que están asociadas [18].

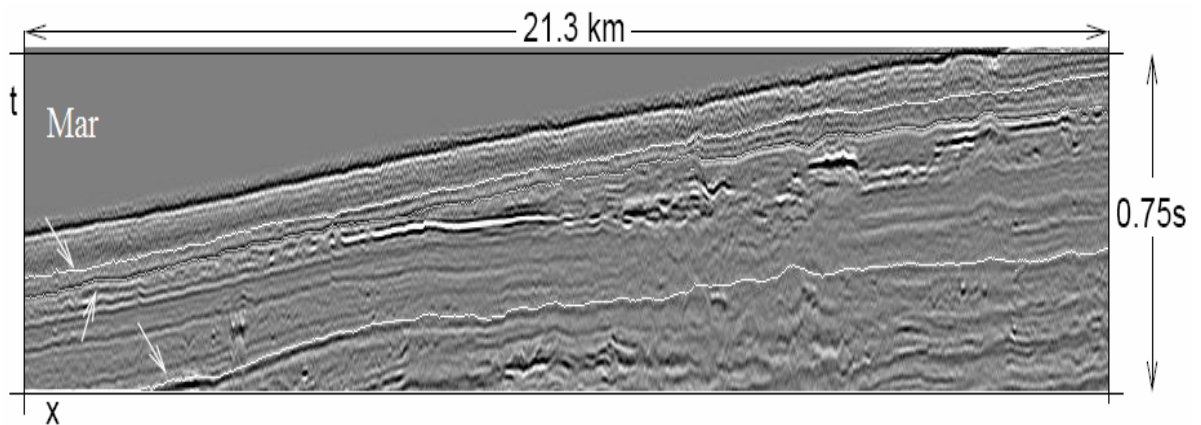


Figura 6. Imagen de sección sísmica.

Uno de los objetivos más importantes, es la construcción de mapas estructurales que combinados con rasgos geológicos brindan congruencia espacial en términos de interpretación. Los rasgos que se examinan pueden ser fallas, estructuras plegadas y de flujo, arrecifes, discordancias, canales y trampas estratigráficas [1].

En términos simples una sección sísmica puede entenderse como una ecografía del subsuelo. Estas imágenes usualmente muestran un pastel de capas en el que se presentan cambios abruptos en las propiedades físicas de estas en sentido vertical, pero solo cambios graduales lateralmente a lo largo de cada capa [19]. Por lo tanto, la identificación de capas guía (reflectores) es indispensable en interpretación sísmica (Figura 6).

En la Figura 6, por ejemplo se observa una línea sísmica de 21.3 km de longitud y 0.75 segundos de penetración vertical [26], en cuya adquisición se usaron elementos y procesos diseñados con anterioridad por los geólogos. Después de aplicar el procesamiento sísmico se llegó a la imagen ‘apilada’ que se observa. En ésta se identificaron tres horizontes o reflectores que a posteridad servirán para definir la estructura del lugar y correlacionando estos datos con los de otras líneas cercanas se podría predecir la existencia o no de hidrocarburos o algún otro mineral de interés.

1.4.1. Reflectores discontinuos

Algunas zonas de exploración suelen estar acompañadas de una densidad de discontinuidades o discordancias en los reflectores que los interrumpen. Geológicamente estas discontinuidades, usualmente formadas a lo largo de millones de años, son de interés pues allí pudieron ocurrir trampas en las cuales están contenidos los hidrocarburos.

A grandes rasgos las discontinuidades pueden ser fallas y adelgazamientos (*pinch-out*). Las fallas geológicas en imágenes sísmicas se visualizan como una interrupción del reflector y pueden ser inversas o normales. Además, las capas que sufren la falla pueden desplazarse varios metros como producto del movimiento tectónico terrestre. Este proceso que puede tardar millones de años por lo general hace que sea identificable porque los reflectores no se desplazan una distancia mayor a su propio espesor. Aunque en algunas zonas en donde la actividad tectónica es elevada, los reflectores se visualizan con un desplazamiento bastante mayor a su espesor. En la Figura 7 se puede observar el fenómeno de las fallas presentes en una imagen sísmica [20].

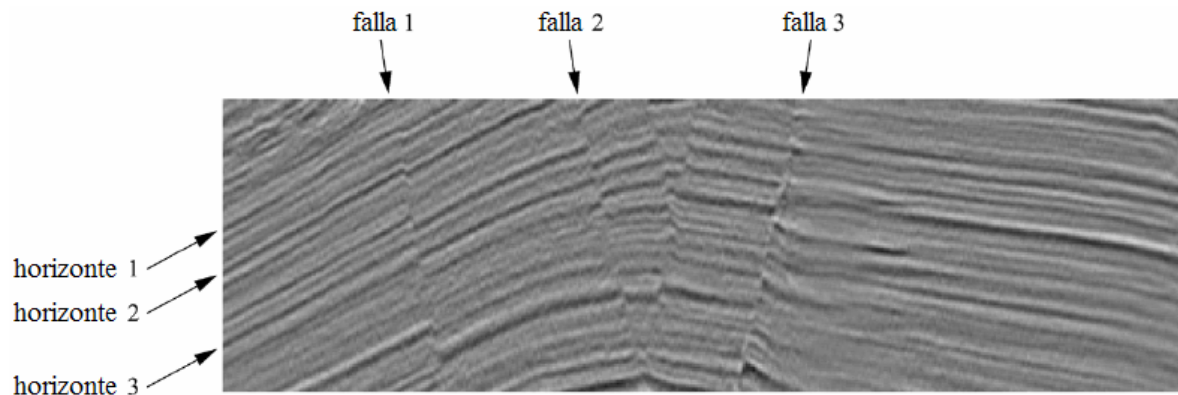


Figura 7. Imagen sísmica con horizontes interrumpidos por fallas geológicas.

Los pinchamientos son fenómenos que sufren los reflectores que se adelgazan y forman uno solo luego de varios kilómetros. En términos de imágenes se visualiza

como una combinación de las reflexiones que da como resultados la fusión de dos o mas reflectores en uno sólo.

En resumen, las imágenes sísmicas son el medio más usado en la interpretación de las litologías terrestres. Por tal razón, identificar reflectores en ellas es de interés en la exploración de hidrocarburos. Los reflectores correctamente identificados son el primer paso para la definición de las estructuras geológicas. En los capítulos siguientes se observará cómo las metodologías planteadas para reconocimiento de estadístico y sintáctico de patrones permiten identificar reflectores continuos y reflectores con cierto tipo de discontinuidades de manera semiautomática.

2. RECONOCIMIENTO DE PATRONES

El uso del reconocimiento de patrones en exploración sísmica de hidrocarburos ha llegado a ser una de las líneas de investigación con mayor importancia en este método geofísico. La interpretación de datos sísmicos de gran tamaño representa un trabajo desafiante. Para resolver este problema se han planteado métodos estadísticos que han explorado diferentes principios en la medición de similitudes entre trazas sísmicas [3], [11], [13], [20]. Por esta razón es importante tomar los trabajos más significativos y evaluar el desempeño frente a diferentes contextos geológicos.

Por otra parte, los métodos para reconocimiento sintáctico de patrones han planteado alternativas innovadoras en el estudio de señales sísmicas. Aunque estos métodos han sido estudiados en detalle durante las últimas tres décadas, en el libro "*Syntactic pattern recognition for seismic oil exploration*" escrito por *Kuo-Yuan Huang* [2], se presenta la primera aplicación clara de la teoría para reconocimiento sintáctico en sísmica. Por esta razón, algunos de los algoritmos que componen los sistemas para reconocimiento sintáctico planteados en el presente trabajo, han sido una selección y adaptación de los originales tomados del citado libro. De esta forma se propone evaluar el desempeño de los métodos sintácticos, adicionando algunas mejoras que en su momento serán explicadas.

2.1. Reconocimiento de patrones por métodos estadísticos

Los métodos estadísticos se basan en medidas de probabilidad aplicadas a las trazas sísmicas que generan como resultado la selección de aquellas muestras que se correlacionan en mayor medida con el patrón de entrada. En la actualidad la teoría utilizada en sísmica es conocida como "cubo de coherencia" (*The coherence cube*) [3]. Esta teoría se orienta al análisis de datos 3D (volúmenes sísmicos), sin embargo también es aplicable a datos 2D (imágenes de línea

sísmica). En la Figura 8 es mostrado un sistema típico para reconocimiento de patrones basado en medidas de similitud.

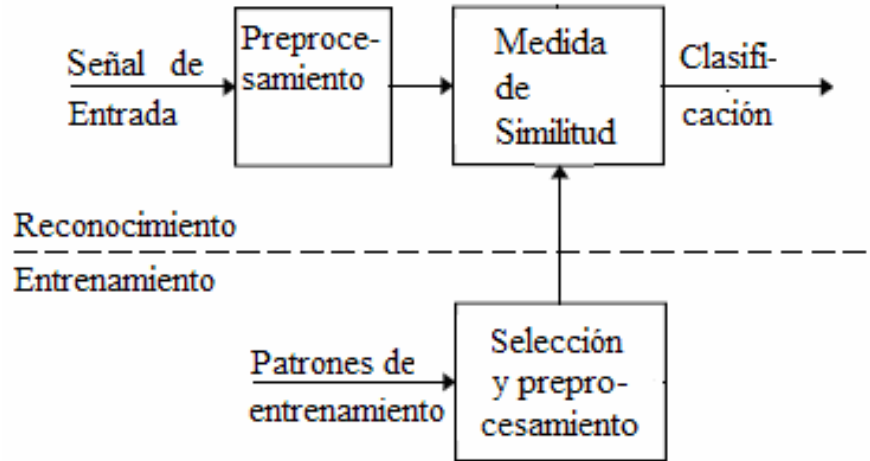


Figura 8. Diagrama de bloques de un sistema estadístico de reconocimiento de patrones

2.1.1. Coeficiente de Coherencia

La medida de coherencia está dada por la correlación cruzada normalizada $C(\tau)$ entre dos trazas sísmicas $s_1(t)$ y $s_2(t)$ [3], [9], [10], [11].

$$C(\tau) = \frac{\langle s_1(t)s_2^*(t+\tau) \rangle}{\sqrt{\langle |s_1(t)|^2 \rangle \langle |s_2(t+\tau)|^2 \rangle}} \quad (1)$$

Donde $\langle . \rangle$ representa promedio temporal. Y se define así:

$$\langle s_1(t)s_2^*(t+\tau) \rangle = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-T/2}^{T/2} s_1(t)s_2^*(t+\tau) dt \quad (2)$$

La ecuación (1) corresponde a *coherencia temporal*. Es posible obtener ecuaciones para *coherencia espectral*, *coherencia espacial* y *coherencia mutua*

considerando combinaciones entre ellas (Ejemplo: *coherencia espectral-temporal*) [11].

Las trazas sísmicas siempre son discretas, por tal razón las ecuaciones (1) y (2) deben ser discretizadas para su uso en identificación de reflectores. Las ecuaciones (3) y (4) corresponden a tales las fórmulas. Para aproximar las integrales inherentes a la ecuación de coherencia, se utilizó el método numérico del trapecio [12].

$$C(k) = \frac{\langle s_1[n]s_2^*[n+k] \rangle}{\sqrt{\langle |s_1[n]|^2 \rangle \langle |s_2[n+k]|^2 \rangle}} \quad (3)$$

$$\langle s_1[n]s_2^*[n+k] \rangle = \sum_{n=T_1}^{T_2} \left(\frac{1}{2} (s_1[n] \cdot s_2^*[n+k] - s_1[n-1] \cdot s_2^*[n+k-1]) \right) \quad (4)$$

Utilizando las ecuaciones (3) y (4) un sistema de identificación de similitudes puede ser planteado. Mediante el siguiente ejemplo es mostrada la forma de operación del sistema para reconocimiento de coherencia entre trazas sísmicas

Ejemplo 2.1:

Considerando las señales s1 y s2 de la Figura 9, se puede calcular el coeficiente de coherencia para encontrar la parte de s2 que es más coherente con s1.

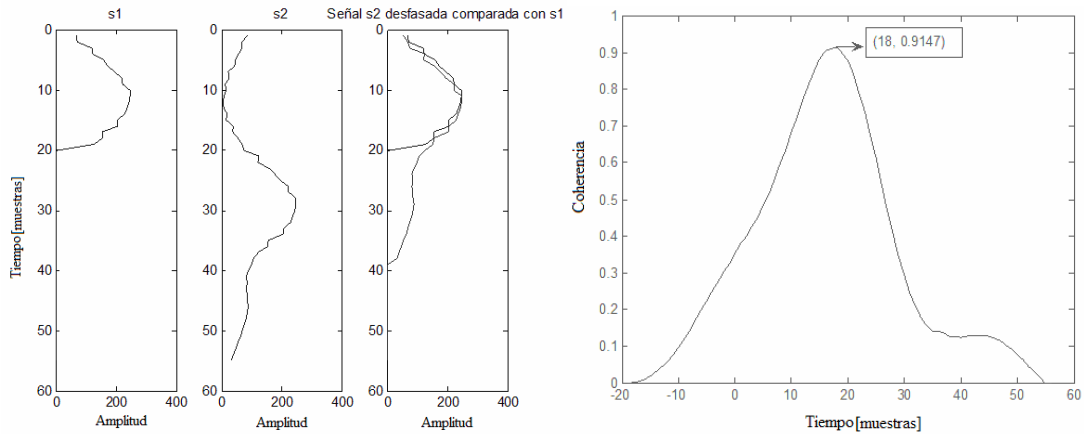


Figura 9. Coherencia entre dos señales aplicando desfase.

Solución.

La ecuación (3) puede ser utilizada para determinar el valor de desfase k de la señal s2 para el cual se obtiene máxima coherencia.

El resultado que se observa en la Figura 9 (derecha) muestra que la parte más coherente de s2 con s1 es la que empieza en la muestra número 18.

2.1.2. Coeficiente de semblanza

El término semblanza fue introducido en 1971 [8] y desde entonces se ha convertido en una de las herramientas más usadas para el procesamiento sísmico. Su definición se presenta como una medida de semejanza entre ventanas de datos en imágenes de sección sísmica (puede ser cualquier tipo de imagen). Estas ventanas representan matrices de datos y el coeficiente de semblanza según [13] se halla mediante la siguiente ecuación:

$$R = \frac{\sum_{t=T1}^{T2} F^2[n] - \sum_{i=1}^N \sum_{n=T1}^{T2} s^2_i[n]}{(N-1) \sum_{i=1}^N \sum_{n=T1}^{T2} s^2_i[n]} \quad (5)$$

Donde $F[n]$ representa la traza apilada, es decir, la suma de todas las trazas de la

ventana $F[n] = \sum_{i=1}^N s_i[n]$. N es el número total de trazas en la ventana, las

constantes $T1$ y $T2$ determinan el intervalo de muestras para el cual se desea encontrar el coeficiente de semblanza (Figura 10). El valor de semblanza para una ventana de imagen sísmica (N trazas entre $T1$ y $T2$) se encuentra en el intervalo

$$\frac{-1}{N-1} \leq R \leq 1.$$

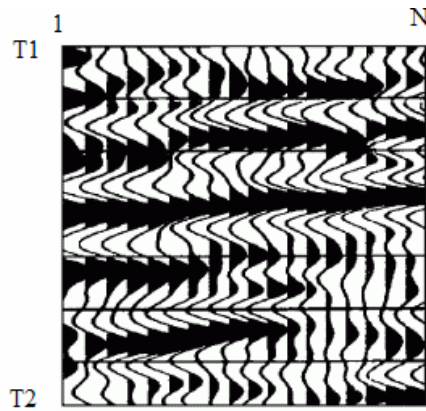


Figura 10. Conjunto o ventana de trazas en el cual se evalúa semblanza.

La ecuación (5) corresponde a una medida de similitud en una ventana de trazas sísmicas. La cualidad de esta ecuación consiste en tener en su numerador la multiplicación entre todos los pares de muestras de cada fila en la ventana. Para llegar a este principio se comienza por la definición del **SNR** en una ventana de trazas [13]:

$$R = \frac{\sum_{t=T1}^{T2} F^2[n]}{(N-1) \sum_{i=1}^N \sum_{n=T1}^{T2} s_i^2[n]} \quad (6)$$

En el numerador de la fracción presente en la ecuación (6) cada fila de muestras en la ventana de trazas es elevada al cuadrado y se suman estos resultados. Sin embargo, se obtienen términos cuadráticos que no ofrecen ninguna ventaja en la

diferenciación de ruido respecto de la señal. Lo anterior es demostrable desarrollando la siguiente operación:

$$(a + b + c)^2 = a^2 + b^2 + c^2 + 2ab + 2ac + 2bc \quad (7)$$

Si se considera a los tres elementos (a, b, c) como una fila en la ventana de trazas, el desarrollo del polinomio permite observar que los tres primeros términos a la derecha del signo igual no se relacionan con otros términos. Observando este hecho, si se quiere obtener en el numerador solamente las parejas de muestras multiplicadas, los términos cuadráticos tales como (a^2, b^2, c^2) deben ser restados y así obtener una expresión en la cual se encuentren las sumas de las parejas de muestras multiplicadas [13]. Por esta razón para obtener el coeficiente de semejanza mostrado en la ecuación (5), la expresión (8) debe ser restada en el numerador del **SNR** mostrado mediante la ecuación (6).

$$R = \sum_{n=1}^N \sum_{t=T1}^{T2} f^2(n, t) \quad (8)$$

El valor de normalización $(N - 1)$ presente en el numerador de la ecuación (5) es necesario ya que existe tal cantidad de multiplicaciones en un grupo de N datos multiplicados por parejas [13].

2.2. Reconocimiento de patrones por métodos sintácticos

Los métodos sintácticos se presentan como consecuencia de un uso específico de la teoría conocida como lenguajes formales. Estos métodos son basados en el análisis de vectores numéricos (señales discretas) que se asocian a cadenas de caracteres estableciendo reglas para tal asociación. Una vez se tienen las diferentes cadenas, son usadas para definir gramáticas y lenguajes que identifican grupos con similitudes definidas [2].

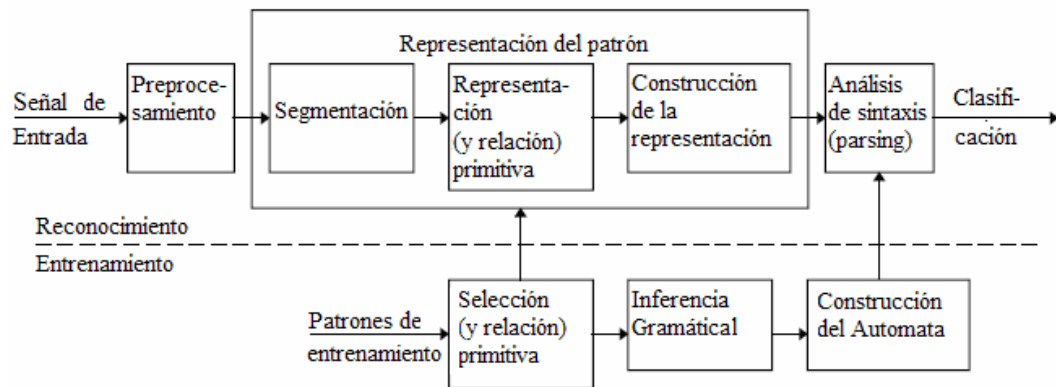


Figura 11. Diagrama de bloques de un sistema sintáctico de reconocimiento de patrones

Una red típica de reconocimiento sintáctico es mostrada en la Figura 11. En ésta se observa por una parte la red de entrenamiento, donde se encuentran los lenguajes que identifican al patrón y por otra parte el clasificador, donde llega la muestra y se analiza su pertenencia a un lenguaje u otro [2]. En la actual sección del libro sólo se presenta el camino a seguir en el uso de los métodos sintácticos. Los procesos que involucran codificación de Freeman, inferencia de gramáticas, *parsing* y distancia de Levenshtein son explicados en detalle mediante los algoritmos del capítulo 3.

2.2.1. Reconocimiento sintáctico usando autómatas de estado finito

Usando gramáticas regulares o de estado finito, a las cuales se asocia un autómata finito, se puede crear una red de reconocimiento sintáctico como se muestra en la Figura 12. Ésta se caracteriza por procesos definidos como *codificación dependiente de amplitud*, *inferencia de gramática de estado finito* y *gramática expandida restringida*, y *Parsing de estado finito* [2].

Gramática: Se define como una cuádrupla de elementos $G = \{V_N, V_T, S, P\}$, donde:

- V_N : Conjunto de elementos no-terminales.
- V_T : Conjunto de elementos terminales.
- S : Conjunto de estados iniciales.

- P: Conjunto de reglas de producción o sustitución.

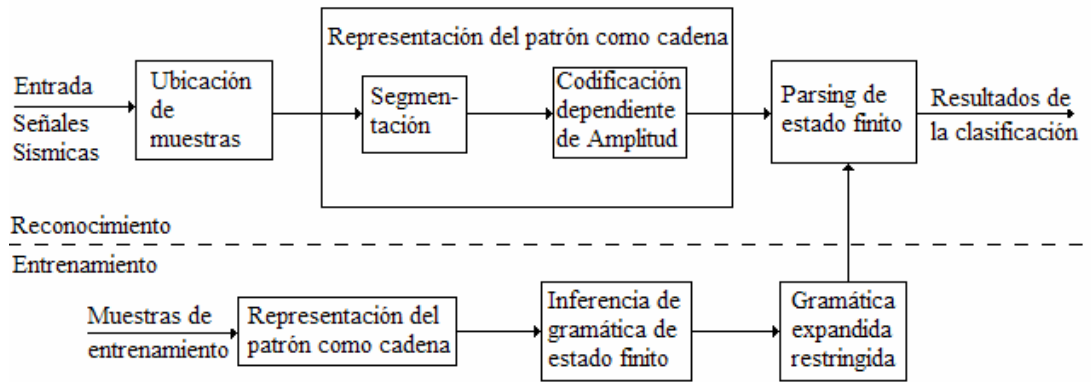


Figura 12. Sistema de clasificación de muestras usando gramática de estado finito

En el ejemplo 2.2 se observa la clasificación de una porción de señal dentro de una traza determinada, según el sistema de la Figura 12. De esta manera se ilustran las definiciones relevantes de la teoría de gramáticas regulares o de estado finito, con el fin de presentar una explicación en términos prácticos.

Ejemplo 2.2:

Se considera la señal $S1$ mostrada en la Figura 13 (izquierda) con amplitud pico a pico igual a 0.6, de la cual se extrae una porción de longitud $n2 = 5$, a partir de la muestra 7. A las muestras de esta porción de $S1$ se les agrega ruido para producir la señal $S2$ (Figura 13 derecha). El cambio en los valores se realizó sumando 0,005 a las muestras 9 a 11 de $S1$. Se desea entonces ubicar a $S2$ dentro de la traza $S1$ usando gramáticas de estado finito.

$$S1 = [0.268, 0.26, 0.104, -0.004, -0.008, -0.107, -0.130, -0.125, -0.120, -0.110, -0.085, -0.09, 0.011, 0.015, 0.0, -0.3]$$

$$S2 = [-0.130, -0.125, -0.115, -0.105, -0.08]$$

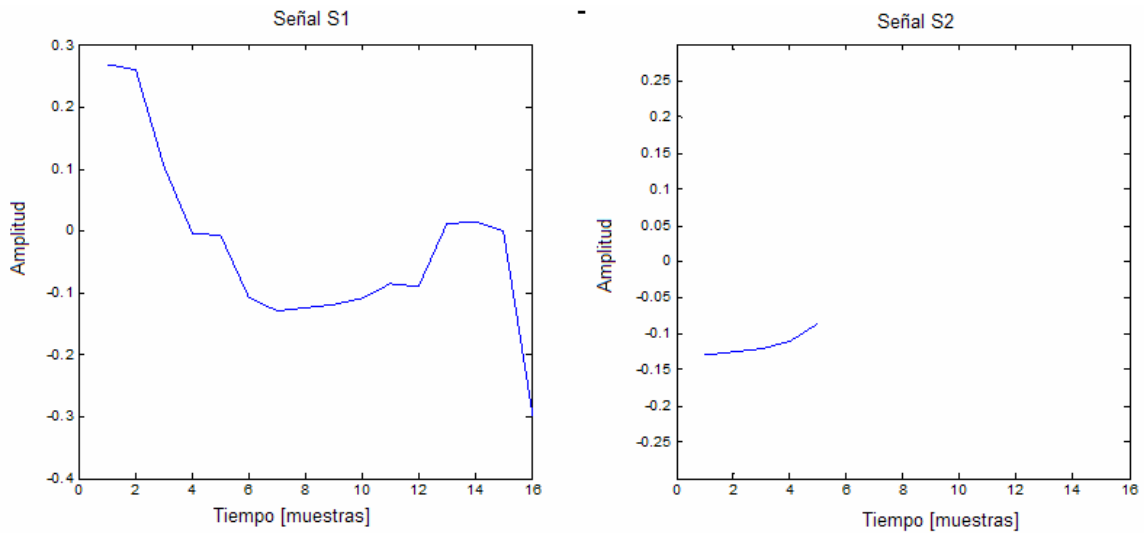


Figura 13. Señal de evaluación y patrón de entrada

Solución.

Para dar solución al problema se utiliza el sistema de la Figura 12. El primer paso es representar las señales propuestas como cadenas de caracteres. Para tal fin, se usa la codificación de cadenas de Freeman. Este es el método usado en [2] y ha sido usado en el presente trabajo por constituir una de las partes del método sintáctico a evaluar. Esta codificación se basa en 9 letras que se asignan según el valor de la diferencia entre dos puntos de la señal numérica [2], [6]. A la diferencia $d_i = y_{i+1} - y_i$ entre las coordenadas verticales de dos puntos adyacentes, (t_i, y_i) y (t_{i+1}, y_{i+1}) , se le asigna un símbolo terminal de acuerdo a la siguiente información:

$$\begin{aligned}
 w_i &= d, & \text{si } 0.05027 < d_i, \\
 w_i &= c, & \text{si } 0.01497 < d_i \leq 0.05027, \\
 w_i &= b, & \text{si } 0.00668 < d_i \leq 0.01497, \\
 w_i &= a, & \text{si } 0.00199 < d_i \leq 0.00668, \\
 w_i &= o, & \text{si } -0.00199 \leq d_i < 0.00199, \\
 w_i &= A, & \text{si } -0.00668 \leq d_i < -0.00199, \\
 w_i &= B, & \text{si } -0.01497 \leq d_i < -0.00668,
 \end{aligned}$$

$$w_i = C, \quad \text{si } -0.05027 \leq d_i < -0.01497,$$

$$w_i = D, \quad \text{si } d_i < -0.05027.$$

Los valores anteriores cambian si la amplitud de la señal no se encuentra en el rango $[-0.3, 0.3]$. Para la señal $S1$ la cadena correspondiente es: $C1 = 'B D D A D C a a b c A d a C D'$. Y la señal $S2$ queda representada mediante: $C2 = 'a b b c'$. Nótese que la diferencia entre $C2$ y su correspondiente en $C1$ (resaltado en negrita) es una letra, en lugar de una a existe una b .

El siguiente paso es inferir una gramática de estado finito para $C2$. Esto se hace mediante inferencia canónica [2], que se caracteriza por llegar a tener tantas reglas de producción como sumen en total las longitudes de las cadenas de entrada. La inferencia canónica es presentada en [2] y hace parte del flujo de prueba en los métodos sintácticos. La gramática inferida G_c debe tener como símbolos terminales todos los que componen el código de Freeman.

$$V_T = \{D, C, B, A, o, a, b, c, d\}$$

Las reglas de producción de acuerdo con la inferencia canónica se definen como P : (0) $S_N \rightarrow aA_N$, (1) $A_N \rightarrow bB_N$, (2) $B_N \rightarrow bC_N$, (3) $C_N \rightarrow c$. Y el conjunto de símbolos no terminales $V_N = \{S_N, A_N, B_N, C_N\}$. Se observa que $C2$ se obtiene partiendo del estado inicial S y aplicando las reglas (0) a (3) en secuencia de la siguiente forma:

$$S_N \xrightarrow{(0)} aA_N \xrightarrow{(1)} abB_N \xrightarrow{(2)} abbC_N \xrightarrow{(3)} abbc$$

La única cadena que se puede formar con la gramática G_c es ' $abbc$ ', pero si se quiere se puede definir la gramática **expandida** de estado finito GEEF a partir de la gramática de estado finito inferida [2]. Su cualidad es que permite formar cualquier cadena a partir del conjunto de símbolos terminales

$V_T' = \{D, C, B, A, o, a, b, c, d, \lambda\}$ (la diferencia entre V_T y V_T' es que en el segundo se tiene en cuenta el símbolo λ que es el terminal de longitud nula). En este tipo de gramática se consideran tres tipos de transformaciones de error: sustitución, inserción y borrado [2], las cuales permiten sustituir, insertar y borrar símbolos terminales asignando un valor específico para cada incidencia en el error. El costo total para generar una cadena particular (de cualquier longitud) usando el conjunto de símbolos terminales V_T' , a partir de la GEEF, es igual a la suma total del costo de cada transformación de error cometida. La GEEF para G_c puede ser representada, de igual forma, mediante el autómata determinista de estado finito mostrado en la Figura 14. En este autómata el costo de cada transformación de error es especificado en cada pareja (i, j), donde i es el símbolo terminal y j es el costo. Es importante aclarar que un autómata de estado finito no corresponde a un dibujo, en realidad es una quintupla de elementos $A = \{\Sigma, Q, \delta, q_0, F\}$ cuyas componentes representan un conjunto especial de propiedades [2]. En el presente trabajo no es necesario realizar operaciones con las componentes de un autómata, ya que las gramáticas planteadas son suficientes para diseñar un sistema de reconocimiento sintáctico.

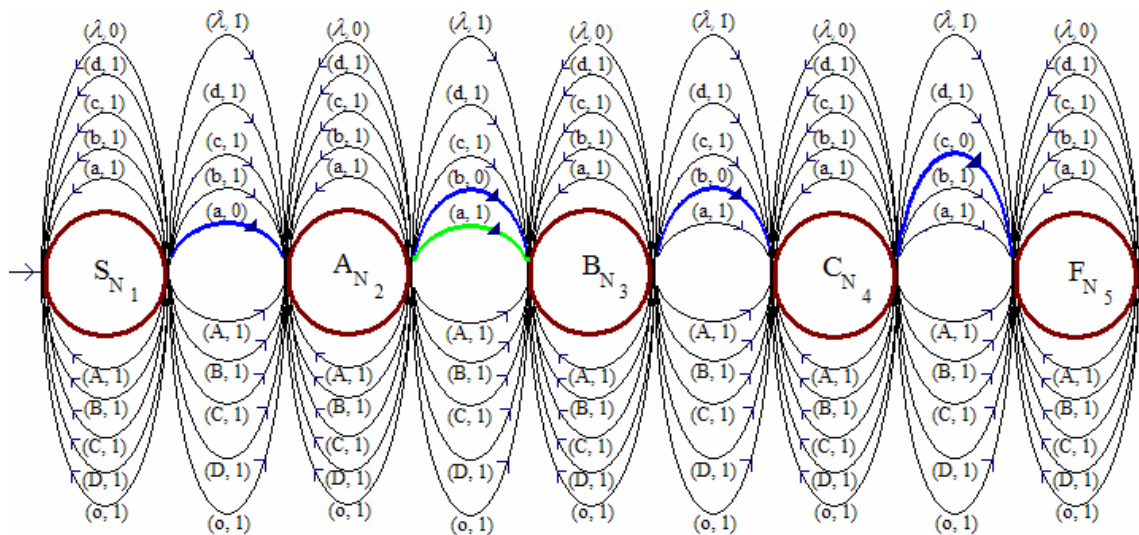


Figura 14. Autómata determinista de estado finito que representa una GEEF con 91 reglas de producción, 5 estados (símbolos no terminales) y 9 símbolos terminales.

En la Figura 14 se observa que si se sigue el camino construido con las reglas marcadas en color azul, el costo total es 0 (cero). La razón es que se formaría la cadena 'a b b c' que es la que dio origen a la gramática G_c (ésta es la única cadena con costo de producción igual a cero). Como el estado inicial es S_N , en el autómata siempre se debe comenzar en el estado número 1. Desde allí, se puede formar cualquier palabra siguiendo el camino que se requiera. El costo total de la cadena será la suma total de los costos que se especifican en cada pareja cuando se sigue el camino que comienza en el estado número 1 (S_N) y termina en el estado número 5 (F_N).

Usando el autómata de la Figura 14, se puede encontrar el costo de producción de cada cadena sobre $C1$ con la GEEF generada a partir de la cadena $C2$. De esta manera se puede encontrar la solución. En la tabla 1 se encuentra el resultado de realizar la función de *Parsing de estado finito* especificado en el diagrama de bloques de la Figura 12, que consiste en encontrar el costo de producción de cada cadena usando el autómata de estado finito.

Iteración	Cadena sobre $C1$	Costo total	Iteración	Cadena sobre $C1$	Costo total
1	'B D D A'	4	7	'a a b c'	1
2	'D D A D'	4	8	'a b c A'	2
3	'D A D C'	4	9	'b c A d'	4
4	'A D C a'	4	10	'c A d a'	4
5	'D C a a'	4	11	'A d a C'	4
6	'C a a b'	3	12	'd a C D'	4

Tabla 1. Costos de producción usando el autómata de la Figura 14.

De la Tabla 1 se concluye que el mínimo costo de producción fue 1 y se obtuvo en la iteración 7, en la Figura 14 se observa que este resultado se obtiene cuando se comete solamente la transformación de error marcada en color verde. La solución del ejemplo 2.2 usando autómatas de estado finito se obtiene de la información anterior. La señal S2 corresponde a la parte de la señal S1 que comienza en la muestra número 7 y que tiene longitud igual a 5.

2.2.2. Reconocimiento sintáctico usando criterio de mínima distancia entre cadenas

Criterios de mínima distancia pueden usarse como clasificadores siguiendo la red de reconocimiento de la Figura 15. La diferencia con la red de autómatas finitos es el bloque de *parsing* y la inferencia de gramática de estado finito que conduce a una gramática expandida. En lugar de estos bloques se usa el criterio de mínima distancia entre cadenas definiendo costos para las transformaciones de error [2].

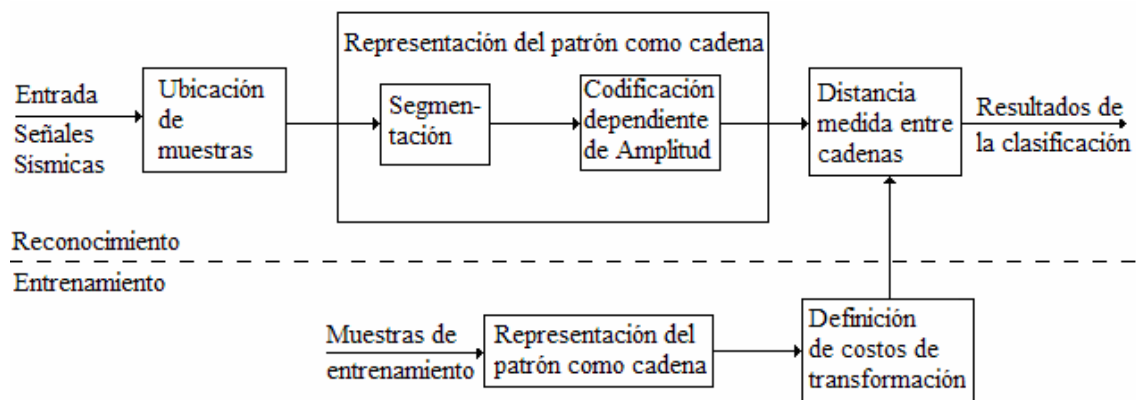


Figura 15. Sistema de clasificación usando mínima distancia medida entre cadenas

Distancia entre cadenas: La distancia entre cadenas da una medida de qué tan parecidas son dos cadenas. Si se tienen en cuenta errores de sustitución, de inserción y de borrado, cada uno con un costo establecido, la mínima distancia entre dos cadenas es aquella para la cual se llegue de una cadena a la otra cometiendo una combinación entre estos errores, tal que el costo total sea

mínimo. El anterior principio se conoce como *distancia entre dos cadenas de Levenshtein* [2], [5], [7].

Definición. Transformaciones de error [5]. Para dos cadenas $x, y \in V_T^*$, se pueden definir tres transformaciones de error en $T : V_T^* \rightarrow V_T^*$ tal que $y \in T(x)$.

(1) T. de error de Sustitución: $w_1aw_2 \xrightarrow{T_s} w_1bw_2$, para todo $a, b \in V_T$. $a \neq b$

(2) T. de error de Borrado: $w_1aw_2 \xrightarrow{T_s} w_1w_2$, para todo $a \in V_T$.

(3) T. de error Inserción: $w_1w_2 \xrightarrow{T_s} w_1bw_2$, para todo $b \in V_T$, donde $w_1, w_2 \in V_T^*$.

Definición. La distancia mínima medida entre dos cadenas $x, y \in V_T^*$, $d^W(x, y)$, es definida como el menor costo de errores de transformación requerido para derivar y de x [7].

En este capítulo se plantearon los conceptos que hacen parte de los sistemas de reconocimiento de patrones que serán utilizados en identificación de reflectores. Las medidas de similitud estadísticas consisten en coeficientes encontrados mediante fórmulas. Usando coherencia se halla la correlación entre pares de trazas, mientras que con semblanza la medida de similitud se halla encontrando la multiplicación normalizada entre las parejas de muestras de cada fila en una ventana de trazas sísmicas.

Los sistemas para reconocimiento sintáctico son más complejos. La asociación de señales numéricas con cadenas de caracteres se realiza mediante codificación de Freeman. Ésta no es la única forma de asociación que existe, pero ha sido implementada por ser la única planteada en el enfoque sísmico [2]. La inferencia canónica, inferencia de la gramática expandida y el *parsing* de estado finito son explicados en detalle en el Capítulo 3. En el mismo capítulo, el método que usa

criterio de Levenshtein es expuesto completamente por medio de algoritmos. Las variaciones y adaptaciones realizadas al método original [2] son el valor agregado en cada método. Estas mejoras son determinadas de acuerdo a la necesidad. En los capítulos que siguen se observarán las versiones finales de los algoritmos y los resultados que estos generaron en imágenes sintéticas y reales.

3. DISEÑO DE LOS ALGORITMOS

A continuación se muestran los pseudo códigos diseñados y la implementación llevada a cabo. Estos algoritmos permiten manipular imágenes sísmicas en forma de matrices bidimensionales. Para todos los casos la primera traza de una imagen sísmica corresponde a la primera columna a la izquierda de la matriz. Todas las trazas de la imagen contienen la misma cantidad de muestras, siendo estas el número de filas de la matriz.

Los algoritmos han sido diseñados con el objetivo de identificar reflectores continuos y superar discontinuidades de corto desplazamiento (no mayor al espesor del reflector).

3.1. Algoritmo de coherencia

En un trabajo anterior se implementó el cubo de coherencia para análisis de datos 3D [3]. Sin embargo, la aplicación en 2D muestra resultados que cumplen con el objetivo de identificar similitudes mediante un coeficiente en el rango de 0 a 1, siendo 1 el mayor grado de similitud. Los algoritmos presentados en la sección 3.1 han sido tomados de [25].

3.1.1 Núcleo del algoritmo

La función diseñada se denomina *Coherencia entre dos señales*, mediante esta se evalúa la coherencia temporal entre dos señales de entrada utilizando la ecuación (3). La señal guía es $s_1[n]$, un vector de número de muestras w igual al espesor del reflector, la señal $s_2[n]$ contiene $3w$ muestras. La señal sometida al desfase k es la de mayor longitud $s_2[n]$.

En la Figura 16 se observa el intervalo al cual se limita la búsqueda en la columna de evaluación. Siendo el vector marcado en azul $s_1[n]$ la parte de esa columna en

la cual se sabe con certeza que está el reflector y la señal $s_2[n]$ el espacio de la columna de evaluación en el cual se halla el valor de coherencia para vectores consecutivos de longitud w .

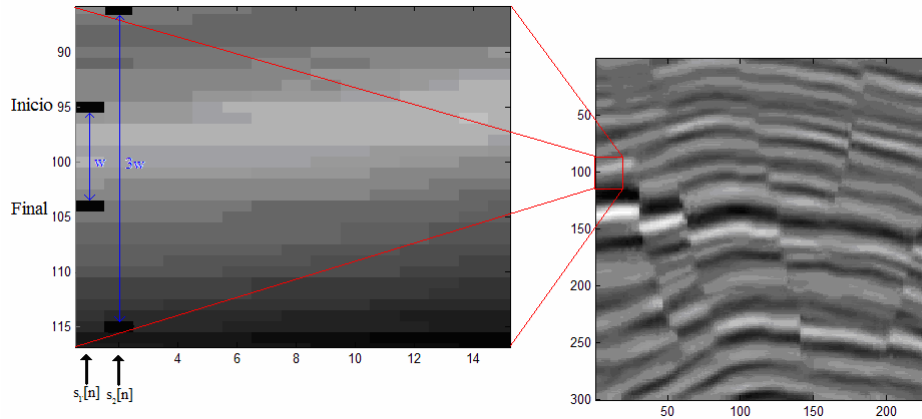


Figura 16. Señales de entrada para los algoritmos de identificación.

Algoritmo 3.1. *Coherencia entre dos señales*

Entrada: Una señal guía $s_1[n]$ con longitud w y la señal de evaluación $s_2[n]$ con longitud $3w$.

Salida: El vector de coherencia temporal $C[k]$ y el valor de k para el cual se produce máxima coherencia.

Método:

- 1) Con las señales discretas $s_1[n]$ y $s_2[n]$, encontrar el valor del denominador de la ecuación (3) utilizando métodos numéricos de integración [12]. Este valor es el mismo para cualquier valor de k .
- 2) Para cada valor de k entre $-w$ y $3w$ encontrar el valor del numerador de la ecuación (3), mediante un método numérico que resuelva la ecuación (4) y encontrar el valor $C[k]$ correspondiente usando el denominador encontrado en el paso anterior.

- 3) Encontrar el valor máximo de $C[k]$ y mostrar como salida la información del desfase k correspondiente a este valor.

El ejemplo 2.1 puede ser solucionado utilizando el algoritmo *coherencia entre dos señales*. La salida del algoritmo es $k = 18$ y el vector de coherencia temporal $C(\tau)$ es graficado en la Figura 9 (Sección 2.1.1). El resultado significa que desfasando 18 muestras la señal de evaluación $s_2[n]$ se obtiene máxima coherencia. La interpretación que debe darse es que la señal guía $s_1[n]$ es más coherente con la parte de $s_2[n]$ que comienza en la muestra 18 y tiene longitud $w = 19$, por lo tanto el reflector que representa $s_1[n]$ comienza en la muestra 18 de $s_2[n]$.

3.1.2. Identificación con coherencia

Obtener una respuesta para toda la imagen implica iterar de tal forma que la función *coherencia entre dos señales* se aplique en cada una de estas repeticiones. La forma de hacerlo es escoger cada cierto intervalo de trazas (ancho de la imagen) una señal guía $s_1[n]$ que identifique la totalidad del reflector en la traza seleccionada, representar con la señal $s_2[n]$ a una de las trazas siguientes en las cuales aun no se ha identificado el reflector y finalmente resaltar la parte de cada traza que fue encontrada como resultado en cada iteración. Cambiar de patrón guía cada cierto número de trazas hace que los resultados sean mejores en discontinuidades de poco desplazamiento. En pseudo código el algoritmo diseñado es el siguiente:

Algoritmo 3.2. Identificación de reflectores usando coeficiente de coherencia.

Entrada: Una imagen sísmica en escala de grises, el inicio 'I' y el final 'F' del reflector en la primera traza de la imagen, y un módulo 'M' para cambio de patrón.

Salida: La imagen sísmica con el reflector identificado.

Método:

- 1) Representar la imagen sísmica como una matriz de datos en la cual la amplitud de cada píxel (i, j) se ubique en la posición (i, j) de la matriz de datos.

Realizar el siguiente paso para todo h entre 1 y el módulo para cambio de patrón M .

- 2) Para cada traza h desde la primera hasta la M -ésima (M es el módulo para cambio de patrón), aplicar el algoritmo *coherencia entre dos señales* tomando como señal guía $s_1[n]$ al vector que comienza en la ubicación $(I,1)$ y termina en la $(F,1)$ (nótese que $s_1[n]$ pertenece a la primera traza de la imagen), la longitud es $w = F - I$. La señal a evaluar $s_2[n]$ comienza en la muestra $I - w$ y termina en la muestra $F + w$ (longitud $3w$) de la traza h -ésima. Luego, con el resultado k , resaltar en la imagen al píxel $I - w + k$ y al $I + k$ como el inicio y el final del reflector en la traza h . Además, cambiar el valor de I por el resultado $I - w + k$ y a F por $I + k$ en cada iteración, este valor representa el inicio del reflector en cada una de las trazas analizadas.

Realizar el siguiente paso para todos los múltiplos l de M , hasta que se hayan analizado todas las trazas.

- 3) Para cada traza h desde la $l \cdot M$ hasta la $(l+1) \cdot M$, aplicar el algoritmo *coherencia entre dos señales* tomando como señal guía $s_1[n]$ a la parte de la traza $l \cdot M$ que pertenece al reflector (ya se había identificado en una iteración anterior). La señal a evaluar $s_2[n]$ comienza en la muestra $I - w$ y termina en la muestra $F + w$ de la traza k -ésima (recordar que los valores de l y de F se han cambiado en cada iteración). Y con el resultado τ , resaltar en la imagen al píxel $I - w + k$ y al $I + k$ como el inicio y el final del reflector en la traza k .

Además, cambiar el valor de I por el resultado $I - w + k$ y a F por $I + k$ en cada iteración.

4) Mostrar como resultado la imagen con el reflector identificado.

Una ilustración del uso del parámetro M es mostrado en la Figura 17. En la imagen mencionada se observa que la discontinuidad de corto desplazamiento presentada entre las trazas 125 y 129 es superada tomando señales guía cada 10 trazas, esto es $M=10$. Si este parámetro no se tuviese en cuenta, la función podría tomar falsas soluciones en las señales donde se encuentra la discontinuidad, pasando a otro reflector diferente al inicial. De la misma manera, se observa que usando el algoritmo *identificación de reflectores*, las medidas de coherencia entre las trazas 110 y 120 se hacen respecto a la traza 110 (señal guía). Este parámetro hace más versátil al algoritmo de coherencia y en general los demás algoritmos que usan éste principio.

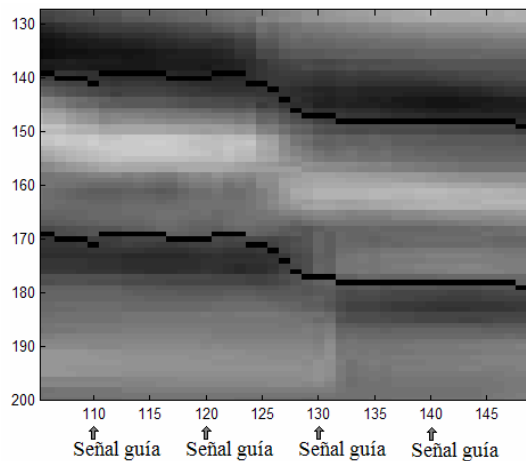


Figura 17. Señales guía o patrones en una imagen determinada.

3.2. Algoritmo de semblanza

Las medidas de similitud son en la actualidad ampliamente usadas en procesamiento de imágenes. Específicamente han sido utilizadas en el método sísmico para determinar velocidades, identificar variables coherentes, obtener

medidas **SNR**, entre otros. La medida de similitud se basa en una solución por mínimos cuadrados obtenida partir del coeficiente de correlación [13]. A partir de la ecuación (5) en este estudio se diseñó un algoritmo para identificación semiautomática de reflectores.

3.2.1. Núcleo del Algoritmo

La función programada se denomina *semblanza entre señales* y encuentra el coeficiente de semblanza entre una señal discreta de entrada y una ventana de señales discretas, todas con igual número de muestras (filas). Se puede visualizar como una función que encuentra el valor de R en la ecuación (5). Por ser una función sencilla el pseudo código no es mostrado. Esta función ha sido enumerada como algoritmo 3.3.

3.2.2. Identificación con semblanza

Encontrar una solución que permita identificar un reflector a partir de algunos datos de entrada implica iterar sobre la imagen de entrada, encontrando los coeficientes de semblanza a lo largo de ésta. El algoritmo diseñado es descrito a continuación:

Algoritmo 3.4. Identificación de reflectores usando coeficiente de semblanza

Entrada: Una imagen sísmica en escala de grises, el inicio 'I' y el final 'F' del reflector en la primera traza de la imagen.

Salida: La imagen sísmica con el reflector identificado.

Método:

- 1) Representar la imagen sísmica como una matriz de datos para la cual la amplitud de cada píxel (i, j) se ubique en la posición (i, j) de la matriz de datos.

- 2) Tomar como patrón la parte de la primera columna de la matriz que representa a la imagen, que corresponde al reflector definido en la entrada por los parámetros I y F . Definir la variable w como el número de muestras del reflector patrón $w = F - I$.

Realizar los pasos 3 y 4 consecutivamente para cada columna de la matriz de datos en dirección izquierda - derecha. La ventana de evaluación de semejanza es limitada en cada columna, se analizan únicamente las muestras comprendidas entre la variable $I - w$ y $F + w$. Las variables I y F cambian en cada iteración. Cada vector de evaluación tiene w muestras y se toma consecutivamente desde $I - w$ hasta $F + w$.

- 3) Encontrar el coeficiente de semejanza usando la función *semejanza entre señales*, los vectores de entrada a esta función son el reflector patrón y la señal de longitud w tomada de la columna actual de evaluación. Guardar el valor encontrado de semejanza en un vector **R** que permita identificar cada iteración.
- 4) Usar el vector **R** para identificar la parte de la columna actual que tiene mayor similitud con el patrón de acuerdo al valor de semejanza. Guardar en las variables I y F el inicio y el final del reflector que obtuvo el mayor coeficiente de semejanza.
- 5) Mostrar como resultado la imagen con el reflector identificado.

3.3. Algoritmo con Autómatas

El algoritmo basado en autómatas de estado finito consiste en un sistema de reconocimiento de patrones como el descrito en la Figura 12. La clave del diseño consiste en aplicar el procedimiento llevado a cabo en el ejemplo 2.2,

considerando el reflector de entrada como patrón. Se proponen los siguientes algoritmos en pseudo código que cumplen con el funcionamiento del diagrama de bloques mencionado.

Las señales numéricas discretas deben ser llevadas al dominio sintáctico, es decir, deben ser transformadas en letras. En el presente trabajo se propone hacer ésta tarea usando la codificación modificada de Freeman mostrada en el ejemplo 2.2. Este tipo de codificación tiene la propiedad de tener rangos exponenciales. El algoritmo *codificación de Freeman*, tomado de [2], realiza esta tarea:

Algoritmo 3.5. Codificación de Freeman

Entrada: Un conjunto de vectores numéricos que representen una señal diferente cada uno.

Salida: Las cadenas de caracteres que representan a cada vector de entrada y la amplitud respecto de la cual se desea codificar.

Método:

Repetir los siguientes pasos para cada vector de entrada.

- 1) Dividir el vector de entrada por la amplitud para normalizarlo, y encontrar la diferencia entre cada componente adyacente haciendo $d_i = y_{i+1} - y_i$, donde y_i es la i -ésima componente del vector.
- 2) Para cada valor de diferencia d_i encontrar la letra correspondiente de acuerdo a la información de la Tabla 2.
- 3) Mostrar la cadena de salida.

Letra asignada	Límite inferior	Límite superior
D	$-\infty$	-0.167567
C	-0.167567	-0.0499
B	-0.0499	-0.022267
A	-0.022267	-0.006633
o	-0.006633	0.006633
a	0.006633	0.022267
b	0.022267	0.0499
c	0.0499	0.167567
d	0.167567	∞

Tabla 2. Codificación de Freeman normalizada.

3.3.1. Inferencia del autómata de clasificación de patrones

La imagen sísmica en su totalidad puede ser convertida a cadenas de caracteres usando el algoritmo anterior. Aquellas cadenas que identifiquen un reflector de interés (patrón) deben ser asociadas a una gramática canónica mediante el algoritmo *Inferencia de gramática canónica de estado finito*. Este algoritmo es una traducción libre del original presentado en [2]. El pseudo código es el siguiente:

Algoritmo 3.6. Inferencia de gramática canónica de estado finito

Entrada: Cadenas de caracteres.

Salida: La gramática canónica de estado finito $G_c = \{V_N, V_T, P, S\}$ asociada a las cadenas de entrada.

Método:

- 1) Chequear cada cadena de entrada e identificar los distintos símbolos terminales usados en estas. Llamar a este conjunto de símbolos terminales V_T .

2) Para cada cadena $x_i = a_{i1}a_{i2} \cdots a_{in}$ generar las reglas de producción correspondientes:

$$\begin{aligned} S &\rightarrow a_{i1}Z_{i1} \\ Z_{i1} &\rightarrow a_{i2}Z_{i2} \\ Z_{i2} &\rightarrow a_{i3}Z_{i2} \\ &\vdots \\ Z_{i(n-1)} &\rightarrow a_{in} \end{aligned}$$

Cada símbolo Z_{ij} representa un nuevo símbolo no-terminal.

3) El conjunto de símbolos no-terminales V_N consiste de S y todos los símbolos no-terminales Z_{ij} producidos en el paso 2. El conjunto P consiste de todas las reglas de producción generadas en el paso 2.

Una gramática canónica, a la cual se le puede asociar un autómata de estado finito, sólo puede producir las cadenas de caracteres que sirvieron para su creación. Ninguna otra cadena puede ser producida. Teniendo en cuenta lo anterior, se pueden tener en cuenta errores de transformación para crear una gramática que pueda producir otras cadenas de caracteres, aunque con un costo dependiente de los errores de transformación cometidos (Sección 2.2.2). Este tipo de gramática recibe el nombre de gramática expandida de estado finito GEEF. El proceso para llegar a esta gramática a partir de una de estado finito se muestra en el algoritmo 3.7. Este pseudo código es una modificación del original presentado en [2]. La diferencia radica en la forma para hallar los costos de sustitución. Se propusieron nuevos costos para inserción, borrado y sustitución.

Algoritmo 3.7. Construcción de la gramática expandida de estado finito

Entrada: Una gramática de estado finito $G = \{V_N, V_T, P, S\}$

Salida: La gramática expandida de estado finito $G' = \{V'_N, V'_T, P', S'\}$ donde P' es un conjunto de reglas de producción con costo.

Método:

- 1) Para cada regla de producción en P con la forma $A \rightarrow a$ de la gramática G , cambiar la regla a la forma $A \rightarrow aF$, donde F es un nuevo símbolo no-terminal.
- 2) Hacer $V'_T = V_T \cup \{\lambda\}$, $V'_N = V_N \cup \{F\}$ y $S' = S$.
- 3) Hacer $P' = P$ con costo 0 (cero) para cada regla de producción original.
- 4) Para cada no-terminal A en V'_N agregar la producción $A \rightarrow \lambda A$ (con costo igual a 0) al conjunto P' .
- 5) Producción del error de sustitución:
Para cada producción $A \rightarrow aB$ en P hacer:
Para cada terminal b en V_T hacer:
Si $A \rightarrow bB$ no está en P' entonces añadir la regla $A \rightarrow bB$ (con costo dependiente de la Tabla 3) a P' .
- 6) Producción del error de inserción:
Para cada no terminal A en V'_N hacer:
Para cada terminal a en V_T hacer:
Agregar la producción $A \rightarrow aA$ al conjunto P' (con costo igual a 3) si esta regla no existía en tal conjunto.
- 7) Producción del error de borrado:
Para cada producción $A \rightarrow aB$ en P ($A \neq B$) hacer:

Agregar la producción $A \rightarrow \lambda B$ al conjunto P' (con costo igual a 3) si esta regla no existía en tal conjunto.

8) Agregar la producción $F \rightarrow \lambda$ (con costo igual a 0) al conjunto P' .

9) Mostrar G' .

	D	C	B	A	o	a	b	c	d
D	0	1	2	3	4	5	6	7	8
C	1	0	1	2	3	4	5	6	7
B	2	1	0	1	2	3	4	5	6
A	3	2	1	0	1	2	3	4	5
o	4	3	2	1	0	1	2	3	4
a	5	4	3	2	1	0	1	2	3
b	6	5	4	3	2	1	0	1	2
c	7	6	5	4	3	2	1	0	1
d	8	7	6	5	4	3	2	1	0

Tabla 3. Costo de sustitución con símbolos terminales de Freeman

En la referencia el algoritmo considera igual costo para todas las sustituciones. En la propuesta actual el costo de sustitución varía de acuerdo con la codificación de Freeman. Cada intervalo de diferencia entre muestras adyacentes que codifica Freeman se puede ver como la pendiente entre cada par consecutivo de muestras. De esta forma es menos costoso (en términos de señales sísmicas) sustituir una pendiente determinada por su contigua que por una más alejada. En otras palabras, es más costoso cambiar una 'o' por una 'D' que por una 'a' o por una 'A'. La propuesta de costos que sigue este principio es mostrada en la Tabla 3.

Si se sigue un proceso con los algoritmos propuestos anteriormente, en el punto en que se obtiene la gramática expandida de estado finito se obtiene a la vez el autómata de estado finito. Este puede tener forma como el mostrado en la Figura

14, con la diferencia de que los costos varían de acuerdo a la Tabla 3. Además el costo de borrado e inserción se ha fijado en 3 y no en 1 como propone la referencia [2]. Si se analiza el autómata se puede obtener una matriz de costos para pasar de un estado a otro con cada símbolo terminal usado. El algoritmo diseñado para analizar las reglas de producción y obtener los costos mencionados es descrito a continuación:

Algoritmo 3.8. Costos en el autómata de estado finito

Entrada: Un autómata de estado finito o la gramática de estado finito $G = \{V_N, V_T, P, S\}$ que describe.

Salida: Las matriz de 3 dimensiones C que contiene los costos para cada símbolo terminal en V_T .

Método:

- 1) Crear la hipermatriz C con cantidad de filas y columnas iguales al número de símbolos no terminales contenidos en V_N . La profundidad de la hipermatriz es igual a la cantidad de símbolos terminales contenidos en V_T . Llenar cada espacio de la matriz creada con ∞ (un número muy grande o infinito).
- 2) Numerar los símbolos no terminales contenidos en V_N , en el orden en que aparezcan en tal conjunto (el estado inicial S debe ser el número 1 y el final F el último en la lista), esto servirá para identificar el cambio de un estado a otro en la hipermatriz. Hacer lo mismo con el conjunto símbolos terminales, teniendo en cuenta que el último símbolo terminal debe ser λ .
- 3) Para cada regla de producción en P :

Llenar el espacio correspondiente en la hipermatriz según el número de cada no terminal y el número del terminal de la regla, con el costo que aparezca en ésta (Figura 18). Este espacio de la hipermatriz corresponde al número de la fila como el no terminal de la izquierda, el número de la columna es el que corresponde al no terminal de la derecha de la regla y el número de matriz es el del terminal de la regla (recordar que una regla de estado finito es de la forma $A \rightarrow aB,c$ donde c es el costo de transición). Las reglas de la forma $A \rightarrow a$ (sin símbolo no terminal de la derecha de la regla) no son tenidas en cuenta para el cálculo de la matriz de costos.

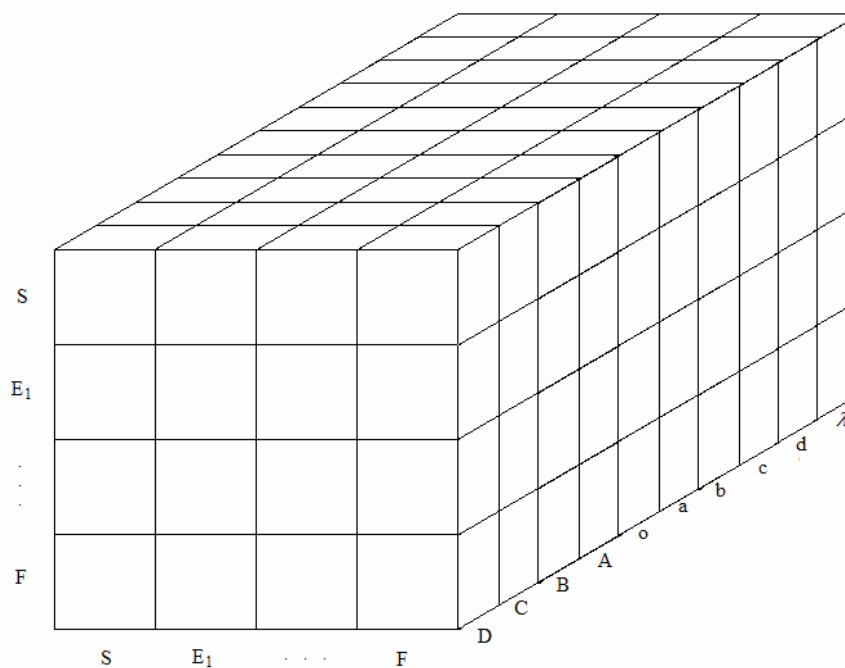


Figura 18. Hipermatriz de costos para códigos de Freeman.

Una vez se tiene la matriz tridimensional de costos que identifica al autómata generado para un patrón dado, se puede plantear un algoritmo que encuentre el mínimo costo de producción de una cadena de entrada usando tal autómata. Este proceso se denomina *parsing de estado finito* (Figura 12). El pseudo código de un algoritmo que realiza este proceso es el siguiente es presentado en [2]. La traducción libre del original es la siguiente:

Algoritmo 3.9. *Parsing de estado finito*

Entrada: Un autómata de estado finito con costos de transición o la gramática que lo identifica. La matriz de costos de transición entre estados C . Una cadena de entrada S .

Salida: El vector M_{1n} que representa el mínimo costo de producción de la cadena de entrada desde el estado 1 hasta el estado n .

Método:

1) $M_{11} = 0$, $M_{1j} = \infty$ (o un número muy grande), para $1 < j \leq n$.

2) Para $1 \leq j \leq n$ hacer:

$M_{1j} = \min\{M_{1k} + C_{kjm}, 1 \leq k \leq n\}$. Donde m es la profundidad de la hipermatriz C , es decir, la matriz de costos del terminal vacío λ .

3) Para $1 \leq h \leq s$ donde s es la longitud de la cadena de entrada hacer:

Crear un vector auxiliar $M^{aux} = M$.

Para $1 \leq j \leq n$ hacer:

$M_{1j} = \min\{M^{aux}_{1k} + C_{kji}, 1 \leq k \leq n\}$. Donde el subíndice i representa la matriz bidimensional componente de C que contiene los costos del terminal h la cadena de entrada S , es decir $S(h)$.

Al llegar a $j = n$, para $1 \leq j \leq n$ realizar:

$M_{1j} = \min\{M_{1k} + C_{kjm}, 1 \leq k \leq n\}$.

4) Mostrar el vector M_{1n} que contiene el mínimo costo de producción de la cadena de entrada S desde el estado 1 hasta el estado n .

En la posición final del vector M se encuentra el costo de producción de la cadena de entrada con el autómata de entrada, ya que toda palabra producida por el autómata debe finalizar en el estado F .

3.3.2. Identificación con autómatas

Usando los algoritmos planteados en el numeral anterior además del de codificación de Freeman se puede diseñar un código que identifique reflectores en imágenes de sección sísmica usando en forma iterativa el diagrama de la Figura 12. El algoritmo diseñado es el siguiente:

Algoritmo 3.10. Identificación de reflectores usando autómatas de estado finito

Entrada: Una imagen sísmica en escala de grises, el inicio ' I ' y el final ' F ' del reflector en la primera traza de la imagen, y un módulo ' M ' para cambio de patrón.

Salida: La imagen sísmica con el reflector identificado.

Método:

- 1) Representar la imagen sísmica como una matriz de datos para la cual la amplitud de cada píxel (i, j) se ubique en la posición (i, j) de la matriz.
- 2) Encontrar un autómata de estado finito y su respectiva hipermatriz de costos C tomando como patrón al vector que comienza en la muestra $(I,1)$ y termina en la $(F,1)$ (nótese que el patrón pertenece a la primera traza de la imagen), la longitud es $w = F - I$. El autómata se halla usando en orden los algoritmos: *codificación de Freeman, inferencia de la gramática canónica de estado finito, inferencia de la gramática expandida de estado finito y costos en el autómata de estado finito.*

Realizar el siguiente paso para todo h entre 2 y el módulo para cambio de patrón M .

- 3) Usar el algoritmo *parsing de estado finito* para encontrar el costo de producción de cada señal de longitud w tomada del intervalo que pertenece a la traza h y que comienza en la muestra $I - w$ y termina en la muestra $F + w$. (Las variables I y F cambian en la iteración $h-1$. Con los costos de producción encontrados tomar como reflector aquella parte de la traza actual que ha obtenido el menor costo de producción. Cambiar los valores de I y F por el inicio y el final del reflector encontrado.
- 4) Cambiar el patrón de entrada por el reflector encontrado en la traza múltiplo del intervalo M y encontrar el autómata correspondiente usando los algoritmos descritos en el paso 1. Realizar el paso 3 consecutivamente a través de las columnas de la matriz de datos (trazas de la imagen sísmica) de izquierda a derecha hasta encontrar un nuevo múltiplo de M que haga cambiar el patrón.
- 5) Mostrar como resultado la imagen con el reflector identificado.

El parámetro M es necesario ya que un cambio en pocas letras de la cadena patrón puede representar grandes cambios en el autómata. Se observa que se requiere una cantidad mayor de algoritmos en este método que en los anteriores (los métodos estadísticos), pero es deber del programador crear *scripts* que realicen eficientemente los procesos descritos.

3.4. Algoritmo de Levenshtein

Tomando como base el criterio de Levenshtein para distancia entre cadenas y usando la codificación de Freeman se puede diseñar un algoritmo que identifique reflectores de forma semiautomática. Se sigue la filosofía de buscar el reflector

sólo en una ventana de tiempo centrada en el reflector identificado de la traza anterior de tamaño igual a 3 veces el espesor del reflector.

3.4.1. Núcleo del algoritmo

Encontrar la mínima distancia de una cadena a otra sirve en la identificación del vector más parecido a un patrón dado. Un algoritmo que realice la tarea de comparar dos cadenas de entrada y encontrar la mínima distancia entre ellas es aquel que recorra la matriz de la Figura 19 por el camino de menor costo. El algoritmo que realiza esta tarea es presentado en [2]. En el trabajo actual se propusieron nuevos costos para transformación de sustitución, borrado e inserción. El pseudo código correspondiente es el descrito en el algoritmo 3.11.

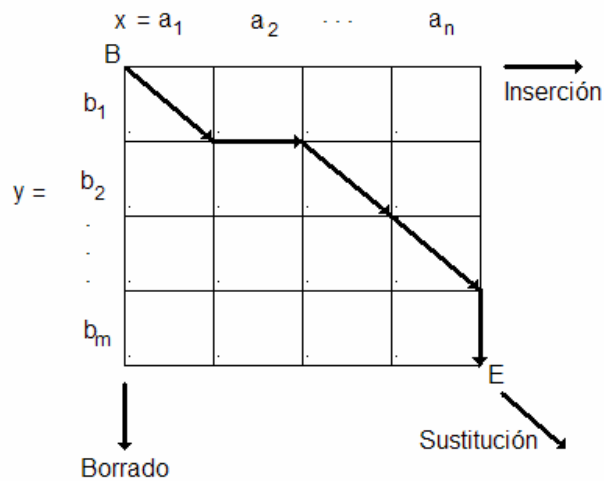


Figura 19. Matriz de recorrido para hallar distancia entre cadenas

Algoritmo 3.11. Distancia medida entre dos cadenas

Entrada: Dos cadenas de caracteres $x = a_1 a_2 \dots a_n$ y $y = b_1 b_2 \dots b_n$.

Salida: El valor de distancia entre las cadenas de entrada.

Método:

1) Hacer $D(0,0) = 0$.

2) Para $i = 1 : n$ hacer:

$D(i,0) = D(i-1,0) + 3$. Esta estructura representa el costo de borrar toda la cadena 1.

Para $j = 1 : m$ hacer:

$D(0,j) = D(0,j-1) + 3$. Esta estructura representa el costo de insertar toda la cadena 2.

3) Para $i = 1 : n$ hacer:

Para $j = 1 : m$ hacer:

$$e_1 = D(i-1, j-1) + S(a_i, b_j)$$

$$e_2 = D(i-1, j) + 3$$

$$e_3 = D(i, j-1) + 3$$

$$D(i, j) = \min(e_1, e_2, e_3)$$

4) Mostrar el costo de producción $D(n, m)$.

La función $S(a_i, b_j)$ representa el costo de sustitución del terminal a_i por b_j . Este costo se halla de la Tabla 3 en el diseño actual. Lo anterior se ha planteado por el hecho de usar siempre codificación de Freeman.

3.4.2. Identificación usando criterio de Levenshtein

Usando el algoritmo descrito en el numeral anterior se puede plantear un método para identificación de reflectores. El algoritmo diseñado es mostrado en el siguiente pseudo código.

Algoritmo 3.12. Identificación de reflectores usando criterio de mínima distancia de Levenshtein

Entrada: Una imagen sísmica en escala de grises, el inicio ' I ' y el final ' F ' del reflector en la primera traza de la imagen, y un módulo ' M ' para cambio de patrón.

Salida: La imagen sísmica con el reflector identificado.

Método:

1) Representar la imagen sísmica como una matriz de datos para la cual la amplitud de cada píxel (i, j) se ubique en la posición (i, j) de la matriz.

Realizar los pasos 2, 3 y 4 para todas las columnas de la matriz de datos $i = 1 : m$ donde m es el número de trazas en la imagen de entrada.

2) Si la columna actual es un múltiplo de M , esto es $i = l \cdot M, l = 1, 2, 3, \dots$. Tomar como patrón al vector que comienza en la muestra $(I, l \cdot M)$ y termina en la $(F, l \cdot M)$, la longitud es $w = F - I$.

3) Encontrar la distancia entre el patrón y cada vector de la columna actual perteneciente al intervalo que comienza en $I - w$ y termina en la muestra $F + w$ usando el algoritmo *distancia medida entre dos cadenas*. (Las variables I y F cambian en cada iteración).

4) El vector que de cómo resultado la menor distancia medida desde la cadena patrón debe ser marcado como el reflector en la columna actual. Las variables I y F toman los valores del inicio y el final del reflector encontrado.

5) Mostrar como resultado la imagen con el reflector identificado.

De esta forma se obtiene el último de los 4 métodos propuestos para identificación semiautomática de reflectores en imágenes de sección sísmica.

Para concluir, en identificación de reflectores por métodos estadísticos se diseñaron algoritmos que clasifican una señal de evaluación según la similitud con una traza patrón. El espacio de búsqueda es limitado a tres veces el ancho del reflector patrón. Gracias a esto, se evita que el algoritmo siga la trayectoria de un reflector diferente al original, ya que existen reflectores muy parecidos en amplitud y frecuencia en una misma imagen sísmica. Se han planteado dos criterios de similitud estadística. Estos son coherencia y semblanza. En el primero se usa un cambio de patrón cada cierto número de trazas con el objetivo de superar las discontinuidades en la imagen, mientras que en el segundo sólo se usa un patrón en toda la imagen. En el Capítulo 4 se observará la consecuencia de éste diseño.

Los métodos sintácticos se han propuesto con el objetivo de explorar un camino innovador en el área de reconocimiento de patrones sísmicos. Algunos algoritmos han sido tomados del único trabajo que ha aplicado esta teoría en el método sísmico [2]. Sin embargo, el citado trabajo no plantea una solución en identificación de reflectores. Por esta razón se han planteado algunas mejoras que presentan una solución a algunos inconvenientes observados, como se resume a continuación: a pesar de que la codificación de Freeman es presentada en [2], en el trabajo actual se formalizó un algoritmo que realiza este proceso usando intervalos normalizados a señales de amplitud pico a pico igual a 2. La tabla 2 presenta estos intervalos. De esta forma para llevar una señal a cadena de caracteres, ésta debe ser dividida por la mitad de su amplitud pico a pico. La inferencia canónica fue tomada de [2]. En los algoritmos para inferir la gramática expandida de estado finito y distancia entre cadenas, se planteó un nuevo conjunto de costos. La Tabla 3 contiene los costos propuestos. La asignación de estos valores se propone teniendo en cuenta que debe tener un costo menor reemplazar una pendiente por su contigua que por una más alejada. Siguiendo este principio también es necesario cambiar los costos de inserción y borrado por

un valor medio como 3. Si el costo de inserción y borrado fuese 1, borrar e insertar un carácter con costo de sustitución igual a 3, sería menos costoso que sustituirlo.

Por otra parte, el Algoritmo 3.8 encuentra los costos en un autómata. En éste, se produce una matriz tridimensional que identifica completamente al autómata. El algoritmo es un diseño original de éste trabajo. Para realizar el *parsing* de estado finito fue necesario tomar el algoritmo planteado en [2].

El método de identificación de reflectores que usa el criterio de Levenshtein también es basado en la codificación de Freeman. Igualmente, se adoptó la nueva propuesta planteada para los costos de sustitución, borrado e inserción. Los algoritmos para reconocimiento sintáctico de reflectores fueron un diseño original presentado en el trabajo actual que utiliza el rango de búsqueda en la señal de evaluación planteado para el algoritmo de coherencia. Con estos métodos se iniciará un trabajo de evaluación del desempeño. Los resultados obtenidos son mostrados en el siguiente capítulo.

4. EVALUACIÓN DEL DESEMPEÑO DE LOS ALGORITMOS

Para realizar la evaluación del desempeño se tomaron en cuenta dos parámetros a medir: la rapidez y el índice de aciertos. El conjunto de datos en su mayoría es sintético. Los parámetros de entrada a los algoritmos son sugeridos en el numeral 4.2. Las pruebas se realizaron sobre tres geometrías de falla: fallas verticales, inclinadas con ángulo negativo respecto de la normal e inclinadas con ángulo positivo respecto de la normal. Además se realizaron algunas pruebas que demuestran el comportamiento en adelgazamientos (*pinch-out*). Los adelgazamientos utilizados son producto de un truncamiento de reflectores.

4.1. Conjunto de datos

Los resultados se hallaron analizando imágenes sintéticas obtenidas de [17], [20] y [26]. También se realizaron pruebas en imágenes reales presentes en la base de datos del convenio 003 de 2005 UIS-ICP, además de una imagen real del campo de gas Boonsville en Estados Unidos de América. Las imágenes usadas para probar los algoritmos son las mostradas en las Figuras 20 a 26. En tales imágenes se han seguido los reflectores primarios (reflectores con mayor nitidez, son los más blancos o más oscuros). En esta etapa no es necesario obtener información de la equivalencia de los ejes con píxeles. Por esta razón, las imágenes mostradas se encuentran en píxeles. También se probó el funcionamiento en una imagen real de la línea 1440 del Campo Escuela de la UIS mostrada en la Figura 5.

Para todas las imágenes desde la 20 hasta la 26 los reflectores primarios son identificables por los círculos rojos. Asimismo se encuentran fallas con inclinación negativa (color verde), fallas con inclinación positiva (color amarillo) y fallas verticales (color azul). La base de datos de las Figuras mencionadas incluye reflectores y fallas que fueron revisados y aprobados por el geofísico Germán Ojeda (Funcionario ECOPETROL – ICP).

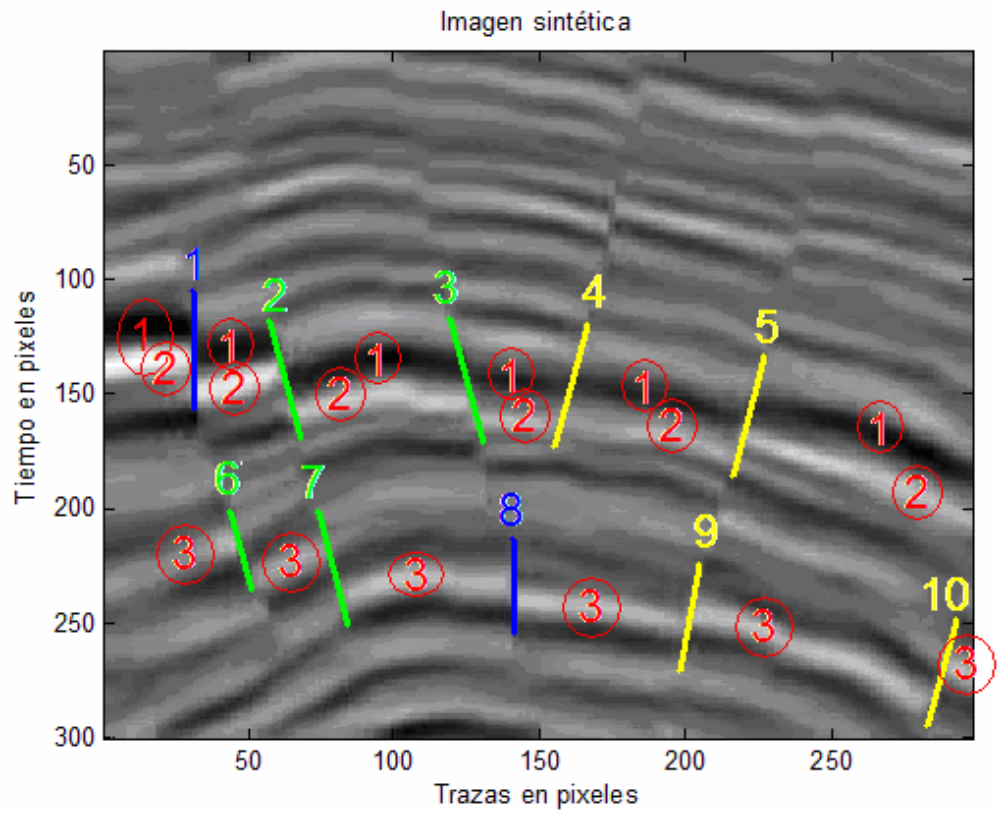


Figura 20. Imagen sísmica sintética con 3 reflectores primarios

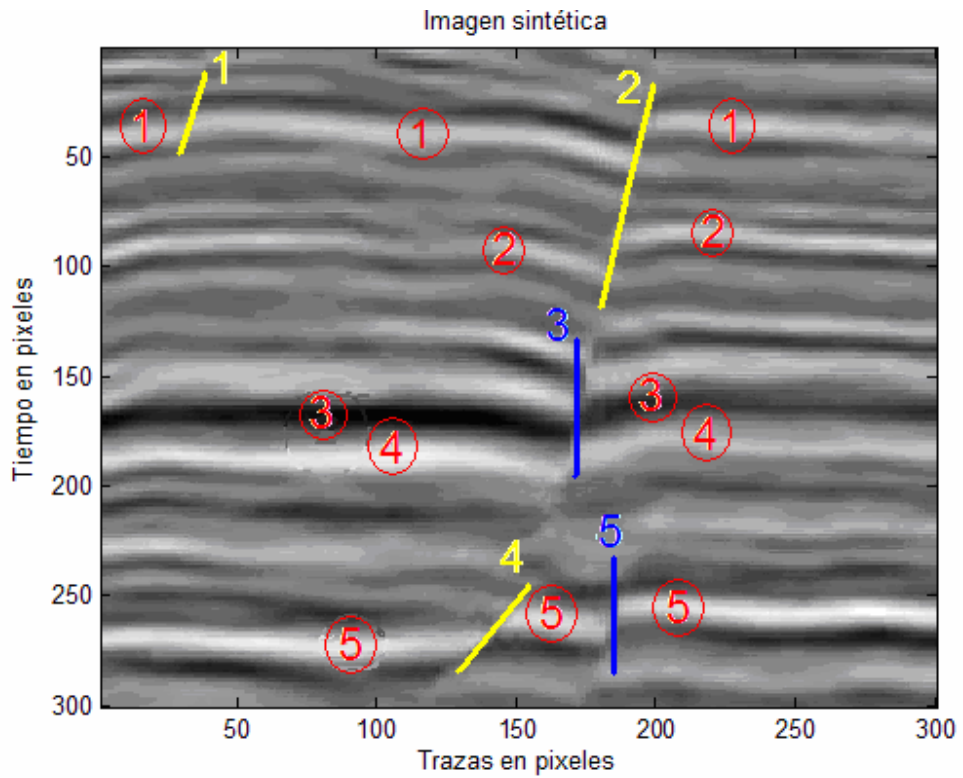


Figura 21. Imagen sísmica sintética con 5 reflectores primarios

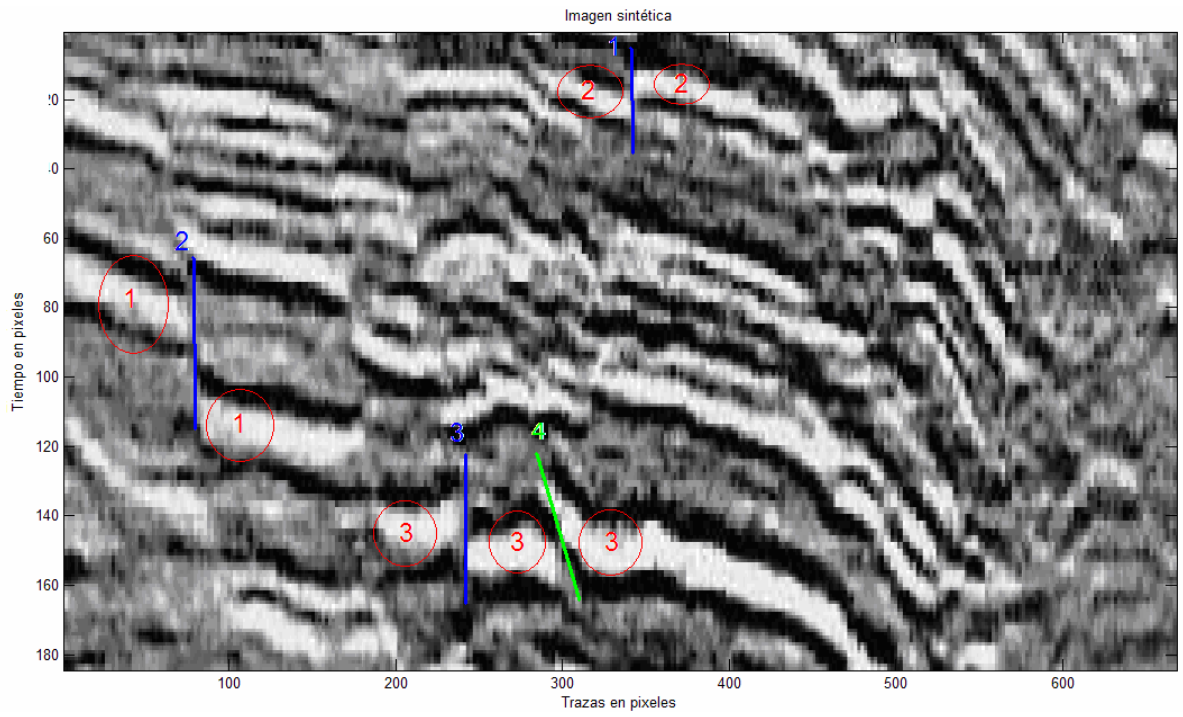


Figura 22. Imagen sísmica sintética con 3 reflectores primarios

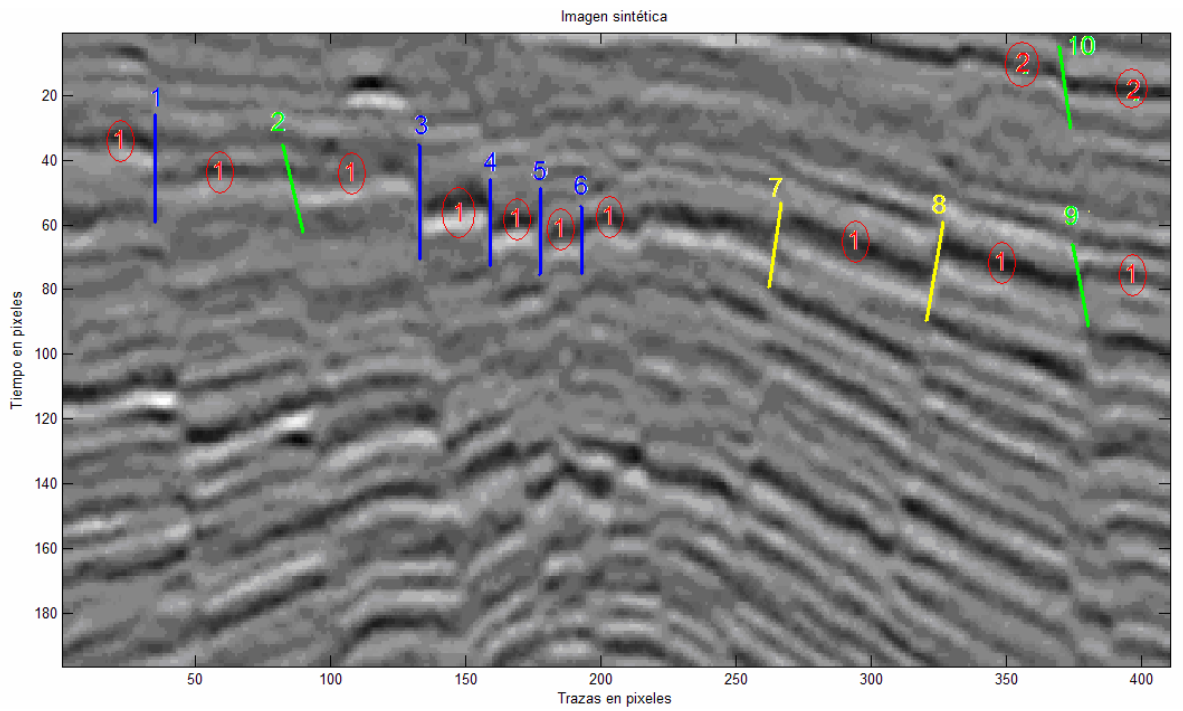


Figura 23. Imagen sísmica sintética con 2 reflectores primarios

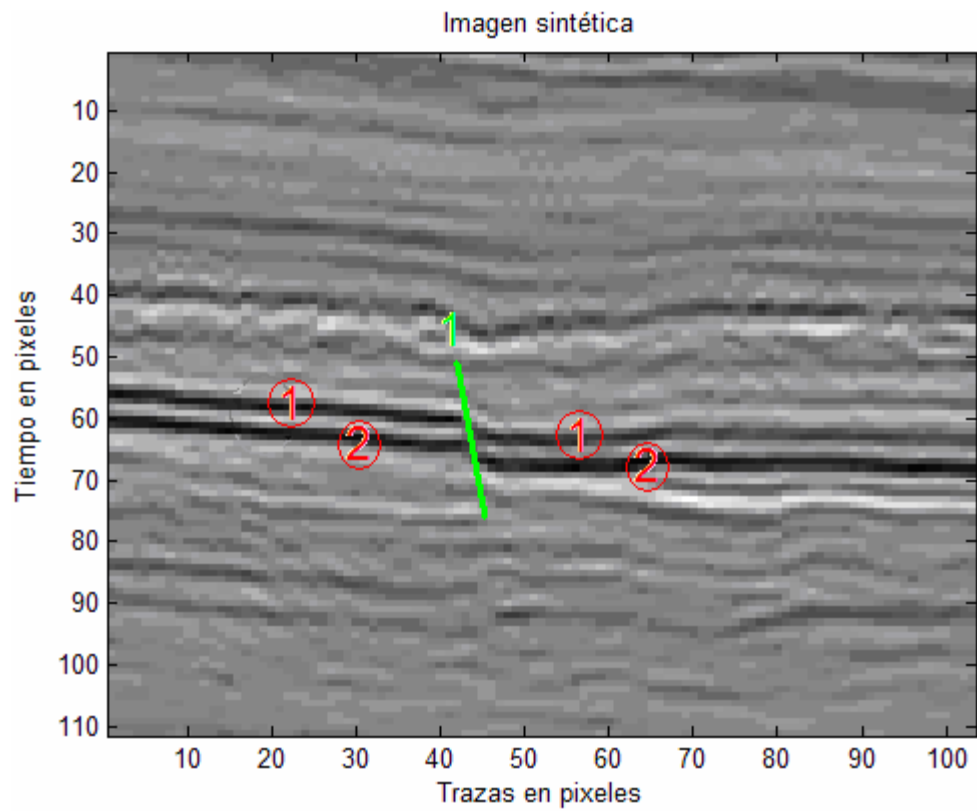


Figura 24. Imagen sísmica sintética con 2 reflectores primarios

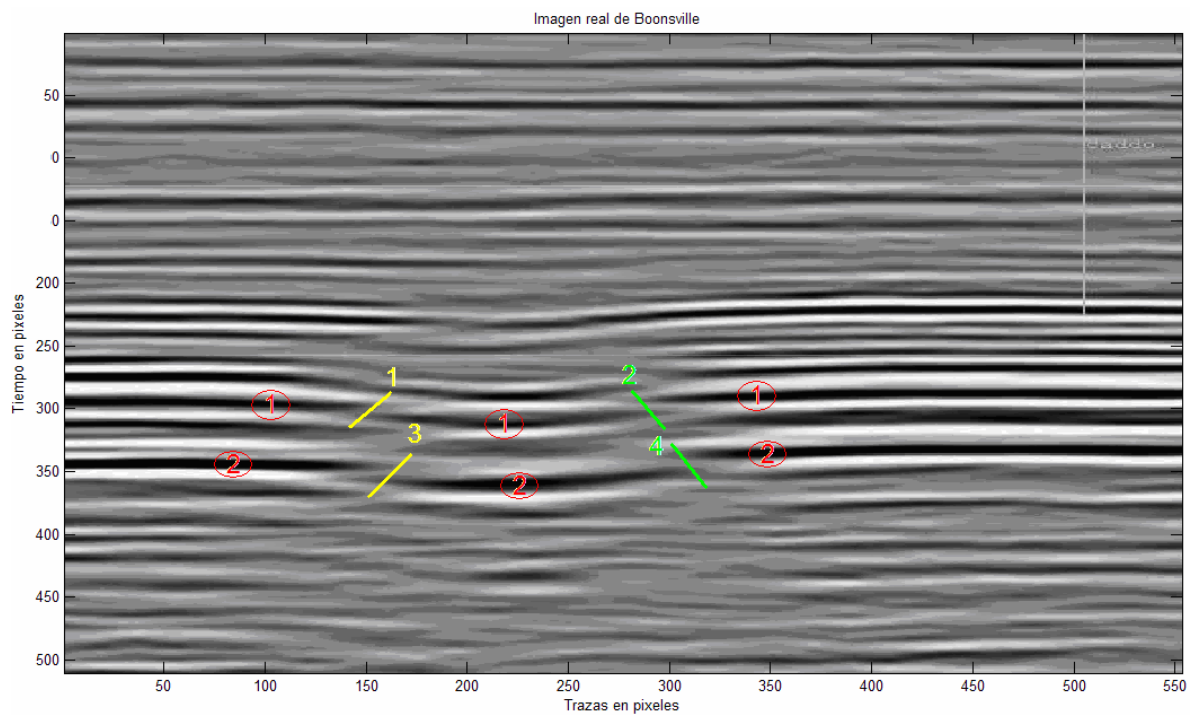


Figura 25. Imagen sísmica de Boonsville con 2 reflectores primarios

En la Figura 26 se muestran dos truncamientos de reflectores. Estos se encuentran encerrados en elipses de color morado. Además se observa una falla vertical. Para estas discontinuidades se evaluó el desempeño.

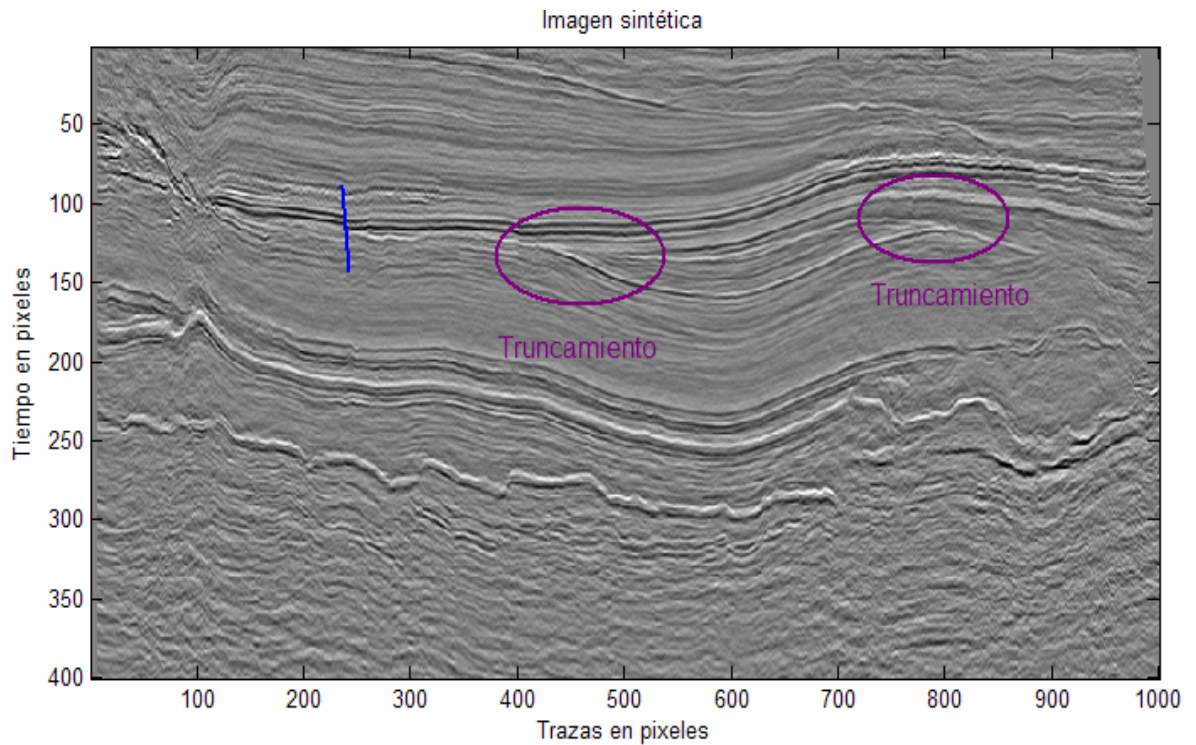


Figura 26. Imagen sísmica sintética con 2 reflectores primarios

4.2. Elección de los parámetros de entrada

Los parámetros de entrada de cada algoritmo son decisivos en los resultados. Por esta razón, se han realizado pruebas para seguir reflectores continuos con diferentes propuestas de ingreso de parámetros. Como conclusión se obtuvo que los parámetros deben ser elegidos para un reflector discreto. De otra manera se pueden obtener resultados erróneos como los mostrados en la Figura 27. Además en los métodos que usan intervalo para cambio de patrón, este debe ser elegido de tal forma que no tome un reflector patrón que no se encuentre afectado por fallas o discontinuidades (Figura 17). La interfaz fue programada para aplicar los algoritmos de derecha a izquierda y también de izquierda a derecha. De esta forma, la falla puede estar a la derecha o a la izquierda de la traza escogida.

La traza escogida debe ser aquella en la cual el reflector patrón se encuentre más nítido, sea geológicamente razonable, y tenga una forma tal que el usuario pueda distinguir un mayor parecido con la parte continua del reflector al cual este pertenece.

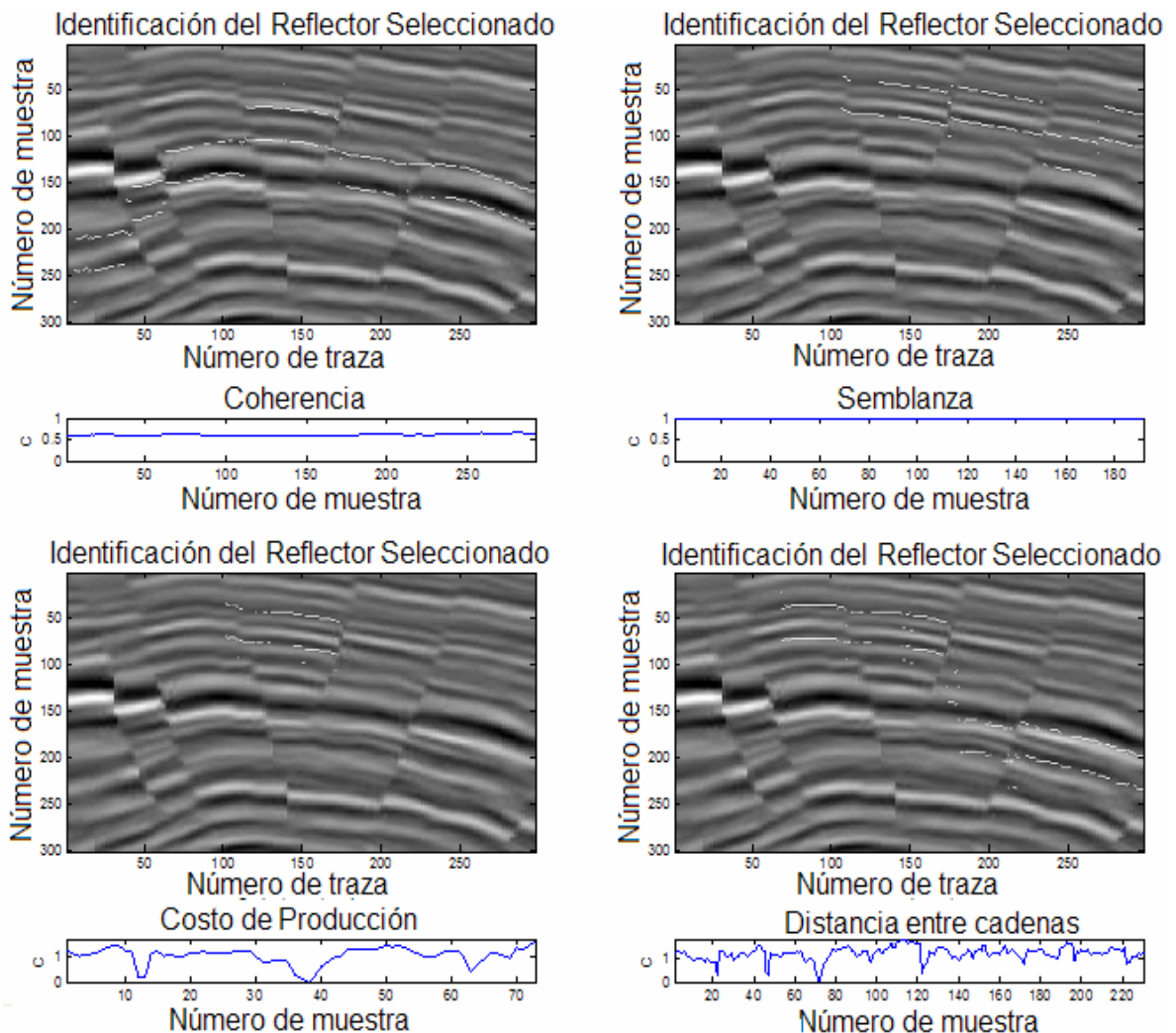


Figura 27. Resultado de elegir más de un reflector como parámetro de entrada

4.3. Resultados en datos sintéticos

Las pruebas se realizaron mediante el uso la interfaz gráfica *ARI* (*Identificación automática de reflectores – Manual en el ANEXO*) programada bajo la plataforma MATLAB®. En tal interfaz se ingresaron los parámetros de entrada requeridos por

cada función según los algoritmos presentados en el Capítulo 3 y en la sección 4.2. La herramienta puede cargar imágenes en los formatos de imagen *.jpg*, *.gif*, *.bmp*, *.tif*, y los formatos internacionales para representación de datos sísmicos *.seg* y *.sgy*. La lectura de los 4 primeros formatos puede realizarse fácilmente mediante las funciones de MATLAB®. Para los formatos internacionales fue necesario usar la función ‘*readseg*’ tomada de la herramienta de libre distribución *SeismicLab* [27]. Las pruebas permitieron observar la imagen de entrada con el reflector identificado. En las tablas 4 a 16 son listados los parámetros de entrada que sirvieron para identificar el reflector en cada traza, y el resultado positivo o negativo. En cada prueba se realizaron 3 intentos, escogiendo parámetros de entrada como se sugiere en la sección 4.2. La abreviación N. A. significa no aplica, y se usó en el algoritmo de semblanza pues éste no usa intervalo para cambio de patrón. En todos los casos se consideró acierto cuando el reflector se identificó de forma correcta en las trazas antes y después de la falla hasta la siguiente discontinuidad.

Algoritmo	Falla	Reflector	Intervalo	Inicio	Final	Traza escogida	Acierto
Coherencia	1	1	25	114	125	17	SI
Semblanza	1	1	25	114	125	17	SI
Autómatas	1	1	25	114	125	17	SI
Lenvenshtein	1	1	25	114	125	17	SI
Coherencia	1	2	25	127	141	14	SI
Semblanza	1	2	N. A.	127	141	14	SI
Autómatas	1	2	25	127	141	14	SI
Lenvenshtein	1	2	25	127	141	14	SI
Coherencia	8	3	25	234	247	153	SI
Semblanza	8	3	N. A.	234	247	153	SI
Autómatas	8	3	25	234	247	153	SI
Lenvenshtein	8	3	25	234	247	153	SI

Tabla 4. Resultados en las fallas verticales de la Figura 20

Algoritmo	Falla	Reflector	Intervalo	Inicio	Final	Traza escogida	Acierto
Coherencia	4	1	25	134	149	142	SI
Semblanza	4	1	N. A.	134	149	142	SI
Autómatas	4	1	25	134	149	142	SI
Lenvenshtein	4	1	25	134	149	142	SI
Coherencia	4	2	25	152	166	140	SI
Semblanza	4	2	N. A.	152	166	140	SI
Autómatas	4	2	25	152	166	140	SI
Lenvenshtein	4	2	25	152	166	140	SI
Coherencia	5	1	25	149	166	231	SI
Semblanza	5	1	N. A.	149	166	231	SI
Autómatas	5	1	25	149	166	231	SI
Lenvenshtein	5	1	25	149	166	231	SI
Coherencia	5	2	15	162	173	199	SI
Semblanza	5	2	N. A.	162	173	199	SI
Autómatas	5	2	15	162	173	199	SI
Lenvenshtein	5	2	15	162	173	199	SI
Coherencia	9	3	40	235	248	174	SI
Semblanza	9	3	N. A.	235	248	174	SI
Autómatas	9	3	40	235	248	174	SI
Lenvenshtein	9	3	40	235	248	174	SI
Coherencia	10	3	25	276	283	281	SI
Semblanza	10	3	N. A.	276	283	281	SI
Autómatas	10	3	25	276	283	281	SI
Lenvenshtein	10	3	25	276	283	281	NO

Tabla 5. Resultados en las fallas con inclinación positiva de la Figura 20

Algoritmo	Falla	Reflector	Intervalo	Inicio	Final	Traza escogida	Acierto
Coherencia	2	1	25	124	137	45	SI

Semblanza	2	1	N. A.	124	137	45	SI
Autómatas	2	1	25	124	137	45	SI
Lenvenshtein	2	1	25	124	137	45	SI
Coherencia	2	2	25	139	151	45	SI
Semblanza	2	2	N. A.	139	151	45	SI
Autómatas	2	2	25	139	151	45	SI
Lenvenshtein	2	2	25	139	151	45	SI
Coherencia	3	1	25	129	143	114	SI
Semblanza	3	1	N. A.	129	143	114	SI
Autómatas	3	1	25	129	143	114	SI
Lenvenshtein	3	1	25	129	143	114	SI
Coherencia	3	2	13	145	156	114	SI
Semblanza	3	2	N. A.	145	156	114	SI
Autómatas	3	2	13	145	156	114	SI
Lenvenshtein	3	2	13	145	156	114	SI
Coherencia	6	3	25	210	238	65	NO
Semblanza	6	3	N. A.	210	238	65	SI
Autómatas	6	3	25	210	238	65	NO
Lenvenshtein	6	3	25	210	238	65	SI
Coherencia	7	3	25	207	234	66	NO
Semblanza	7	3	N. A.	207	234	66	SI
Autómatas	7	3	25	207	234	66	NO
Lenvenshtein	7	3	25	207	234	66	NO

Tabla 6. Resultados en las fallas con inclinación negativa de la Figura 20

Algoritmo	Falla	Reflector	Intervalo	Inicio	Final	Traza escogida	Acierto
Coherencia	3	3	25	157	173	112	SI
Semblanza	3	3	N. A.	157	173	112	SI
Autómatas	3	3	25	157	173	112	SI
Lenvenshtein	3	3	25	157	173	112	SI

Coherencia	3	4	25	152	163	159	SI
Semblanza	3	4	N. A.	152	163	159	SI
Autómatas	3	4	25	152	163	159	SI
Lenvenshtein	3	4	25	152	163	159	SI
Coherencia	5	5	25	255	268	164	SI
Semblanza	5	5	N. A.	255	268	164	SI
Autómatas	5	5	25	255	268	164	SI
Lenvenshtein	5	5	25	255	268	164	SI

Tabla 7. Resultados en las fallas verticales de la Figura 21

Algoritmo	Falla	Reflector	Intervalo	Inicio	Final	Traza escogida	Acierto
Coherencia	1	1	25	26	38	62	SI
Semblanza	1	1	N. A.	26	38	62	SI
Autómatas	1	1	25	26	38	62	SI
Lenvenshtein	1	1	25	26	38	62	SI
Coherencia	2	1	70	36	55	174	SI
Semblanza	2	1	70	36	55	174	SI
Autómatas	2	1	70	36	55	174	NO
Lenvenshtein	2	1	70	36	55	174	NO
Coherencia	2	2	5	78	89	209	SI
Semblanza	2	2	N. A.	78	89	209	SI
Autómatas	2	2	5	78	89	209	SI
Lenvenshtein	2	2	5	78	89	209	SI
Coherencia	4	5	25	255	268	164	SI
Semblanza	4	5	N. A.	255	268	164	SI
Autómatas	4	5	25	255	268	164	SI
Lenvenshtein	4	5	25	255	268	164	SI

Tabla 8. Resultados en las fallas con inclinación positiva de la Figura 21

Algoritmo	Falla	Reflector	Intervalo	Inicio	Final	Traza escogida	Acierto
Coherencia	2	1	25	104	120	87	SI
Semblanza	2	1	N. A.	104	120	87	SI
Autómatas	2	1	25	104	120	87	NO
Levenshtein	2	1	25	104	120	87	NO
Coherencia	1	2	75	11	18	350	SI
Semblanza	1	2	N. A.	11	18	350	SI
Autómatas	1	2	75	11	18	350	SI
Levenshtein	1	2	75	11	18	350	SI
Coherencia	3	3	25	138	148	219	SI
Semblanza	3	3	N. A.	138	148	219	SI
Autómatas	3	3	25	138	148	219	SI
Levenshtein	3	3	25	138	148	219	SI

Tabla 9. Resultados en las fallas verticales de la Figura 22

Algoritmo	Falla	Reflector	Intervalo	Inicio	Final	Traza escogida	Acierto
Coherencia	4	3	25	141	155	337	SI
Semblanza	4	3	25	141	155	337	SI
Autómatas	4	3	25	141	155	337	SI
Levenshtein	4	3	25	141	155	337	SI

Tabla 10. Resultados en la falla con inclinación negativa de la Figura 22

Algoritmo	Falla	Reflector	Intervalo	Inicio	Final	Traza escogida	Acierto
Coherencia	1	1	70	29	38	28	SI
Semblanza	1	1	N. A.	29	38	28	SI
Autómatas	1	1	70	29	38	28	SI
Levenshtein	1	1	70	29	38	28	SI
Coherencia	3	1	50	37	44	122	SI

Semblanza	3	1	N. A.	37	44	122	SI
Autómatas	3	1	50	37	44	122	SI
Levenshtein	3	1	50	37	44	122	SI
Coherencia	4	1	25	54	61	165	SI
Semblanza	4	1	25	54	61	165	SI
Autómatas	4	1	25	54	61	165	SI
Levenshtein	4	1	25	54	61	165	SI
Coherencia	5	1	25	54	62	168	SI
Semblanza	5	1	25	54	62	168	NO
Autómatas	5	1	25	54	62	168	NO
Levenshtein	5	1	25	54	62	168	SI
Coherencia	6	1	25	57	65	184	SI
Semblanza	6	1	25	57	65	184	SI
Autómatas	6	1	25	57	65	184	SI
Levenshtein	6	1	25	57	65	184	SI

Tabla 11. Resultados en las fallas verticales de la Figura 23

Algoritmo	Falla	Reflector	Intervalo	Inicio	Final	Traza escogida	Acierto
Coherencia	7	1	5	55	61	143	SI
Semblanza	7	1	N. A.	55	61	143	SI
Autómatas	7	1	5	55	61	143	SI
Levenshtein	7	1	5	55	61	143	SI
Coherencia	8	1	25	63	73	298	SI
Semblanza	8	1	N. A.	63	73	298	SI
Autómatas	8	1	25	63	73	298	SI
Levenshtein	8	1	25	63	73	298	SI

Tabla 12. Resultados en la falla con inclinación positivo de la Figura 23

Algoritmo	Falla	Reflector	Intervalo	Inicio	Final	Traza escogida	Acierto
Coherencia	2	1	10	41	46	58	No
Semblanza	2	1	N. A.	41	46	58	SI
Autómatas	2	1	10	41	46	58	SI
Levenshtein	2	1	10	41	46	58	SI
Coherencia	9	1	25	70	76	355	SI
Semblanza	9	1	N. A.	70	76	355	SI
Autómatas	9	1	25	70	76	355	SI
Levenshtein	9	1	25	70	76	355	SI
Coherencia	10	2	25	14	20	389	SI
Semblanza	10	2	N. A.	14	20	389	SI
Autómatas	10	2	25	14	20	389	SI
Levenshtein	10	2	25	14	20	389	SI

Tabla 13. Resultados en la falla con inclinación negativa de la Figura 23

Algoritmo	Falla	Reflector	Intervalo	Inicio	Final	Traza escogida	Acierto
Coherencia	1	1	10	58	60	33	SI
Semblanza	1	1	N. A.	58	60	33	SI
Autómatas	1	1	10	58	60	33	SI
Levenshtein	1	1	10	58	60	33	SI
Coherencia	1	2	5	66	69	74	SI
Semblanza	1	2	N. A.	66	69	74	SI
Autómatas	1	2	5	66	69	74	SI
Levenshtein	1	2	5	66	69	74	NO

Tabla 14. Resultados en la falla con inclinación negativa de la Figura 24

Algoritmo	Falla	Reflector	Intervalo	Inicio	Final	Traza escogida	Acierto
Coherencia	1	1	25	291	300	101	SI
Semblanza	1	1	N. A.	291	300	101	SI
Autómatas	1	1	25	291	300	101	SI
Lenvenshtein	1	1	25	291	300	101	SI
Coherencia	3	2	200	339	349	104	SI
Semblanza	3	2	N. A.	339	349	104	SI
Autómatas	3	2	200	339	349	104	NO
Lenvenshtein	3	2	200	339	349	104	SI

Tabla 15. Resultados en las fallas con inclinación positiva de la Figura 25

Algoritmo	Falla	Reflector	Intervalo	Inicio	Final	Traza escogida	Acierto
Coherencia	2	1	250	283	292	385	SI
Semblanza	2	1	N. A.	283	292	385	SI
Autómatas	2	1	250	283	292	385	SI
Lenvenshtein	2	1	250	283	292	385	SI
Coherencia	4	2	25	327	343	354	SI
Semblanza	4	2	N. A.	327	343	354	SI
Autómatas	4	2	25	327	343	354	SI
Lenvenshtein	4	2	25	327	343	354	SI

Tabla 16. Resultados en las fallas con inclinación negativa de la Figura 25

La falla mostrada en la Figura 24 es la versión ampliada de la Figura 26. En esta Figura se observan dos truncamientos de reflectores. En este tipo de contexto geológico se logró que la herramienta siguiese al reflector hasta el adelgazamiento y en este punto tomara otro camino cualquiera. La razón es que el reflector sólo es distinguible hasta la discontinuidad, de allí en adelante el reflector cambia o se combina con otro.

La información de las tablas 4 a 16 es condensada en las tablas 17 y 18. En la primera tabla mencionada, se observa el número de aciertos por tipo de falla en cada algoritmo y el porcentaje al cual conduce. En la segunda se muestra el porcentaje total de aciertos ante fallas. De igual forma, en el desarrollo de los resultados se obtuvo un porcentaje de aciertos en reflectores continuos del 100%.

Algoritmo	Tipo de Falla	Aciertos	Pruebas	Porcentaje
Coherencia	Ángulo positivo	14	14	100%
Semblanza	Ángulo positivo	14	14	100%
Autómatas	Ángulo positivo	12	14	85.15%
Levenshtein	Ángulo positivo	12	14	85.15%
Coherencia	Ángulo negativo	11	14	78.57%
Semblanza	Ángulo negativo	14	14	100%
Autómatas	Ángulo negativo	12	14	85.15%
Levenshtein	Ángulo negativo	12	14	85.15%
Coherencia	Vertical	14	14	100%
Semblanza	Vertical	13	14	92.86%
Autómatas	Vertical	12	14	85.15%
Levenshtein	Vertical	13	14	92.86%

Tabla 17. Resultados de los algoritmos ante distintos tipos de fallas

Algoritmo	Aciertos	Pruebas	Porcentaje
Coherencia	39	42	92.86%
Semblanza	41	42	97.62%
Autómatas	36	42	85.72%
Levenshtein	35	42	83.33%

Tabla 18. Resultados totales de los algoritmos en imágenes sintéticas

4.3. Prueba de rapidez computacional

En la prueba de rapidez fue usada la imagen sísmica sintética de la Figura 20. Ésta imagen cuenta con un total de 300 trazas. La prueba fue realizada siguiendo el reflector número 1 desde la primera traza, variando el espesor del reflector desde 3 hasta 21. El objetivo de la prueba es observar el comportamiento de los algoritmos con respecto al número de muestras del patrón, por esta razón sólo se registró un tiempo por cada prueba. Los resultados son mostrados en la tabla 19 y graficados en la Figura 28. El computador utilizado cuenta con un procesador AMD Turion 64 bits a 1.8 GHz y una memoria RAM de 512 MB de capacidad.

Algoritmo	Tiempo [s]	Muestras analizadas	Algoritmo	Tiempo [s]	Muestras analizadas
Coherencia	2.6996	3	Coherencia	2.8919	12
Semblanza	0.0427	3	Semblanza	0.1592	12
Autómatas	0.6069	3	Autómatas	20.3849	12
Levenshtein	0.2058	3	Levenshtein	3.0166	12
Coherencia	2.7475	4	Coherencia	2.8772	13
Semblanza	0.0581	4	Semblanza	0.1737	13
Autómatas	1.1619	4	Autómatas	24.9062	13
Levenshtein	0.2762	4	Levenshtein	3.7250	13
Coherencia	2.8216	5	Coherencia	2.9345	14
Semblanza	0.0695	5	Semblanza	0.1980	14
Autómatas	1.9851	5	Autómatas	31.0616	14
Levenshtein	0.36819	5	Levenshtein	4.6209	14
Coherencia	2.7970	6	Coherencia	2.9322	15
Semblanza	0.07917	6	Semblanza	0.1993	15
Autómatas	3.2215	6	Autómatas	37.4996	15
Levenshtein	0.5138	6	Levenshtein	5.7290	15
Coherencia	2.7923	7	Coherencia	2.9738	16

Semblanza	0.0919	7	Semblanza	0.2157	16
Autómatas	4.6476	7	Autómatas	44.7491	16
Levenshtein	0.7238	7	Levenshtein	6.9001	16
Coherencia	2.8130	8	Coherencia	2.9190	17
Semblanza	0.1056	8	Semblanza	0.2319	17
Autómatas	6.6212	8	Autómatas	53.3547	17
Levenshtein	1.0004	8	Levenshtein	4.1465	17
Coherencia	2.8057	9	Coherencia	2.9884	18
Semblanza	0.1185	9	Semblanza	0.2438	18
Autómatas	9.1808	9	Autómatas	63.0786	18
Levenshtein	1.3485	9	Levenshtein	9.8092	18
Coherencia	2.83485	10	Coherencia	2.9402	19
Semblanza	0.1346	10	Semblanza	0.2579	19
Autómatas	12.2017	10	Autómatas	67.0590	19
Levenshtein	1.7921	10	Levenshtein	11.4716	19
Coherencia	2.8454	11	Coherencia	2.9505	20
Semblanza	0.1482	11	Semblanza	0.2705	20
Autómatas	15.7393	11	Autómatas	85.2615	20
Levenshtein	2.3214	11	Levenshtein	13.2649	20

Tabla 19. Resultados de rapidez de los algoritmos

Las gráficas de la Figura 28 muestran que los métodos estadísticos tienen un crecimiento aproximadamente lineal con respecto al número de muestras en el reflector patrón. El tiempo de cómputo de los métodos sintácticos muestra crecimiento aproximadamente exponencial. Mientras que en el intervalo de muestras propuesto para la prueba el tiempo de cómputo del método de coherencia es del orden de segundos, en el método de semblanza es del orden de décimas de segundos. En el método de autómatas es del orden de decenas de segundos. En el método de Levenshtein es del orden de segundos, aunque a partir de 12 muestras es mayor que en el método de coherencia.

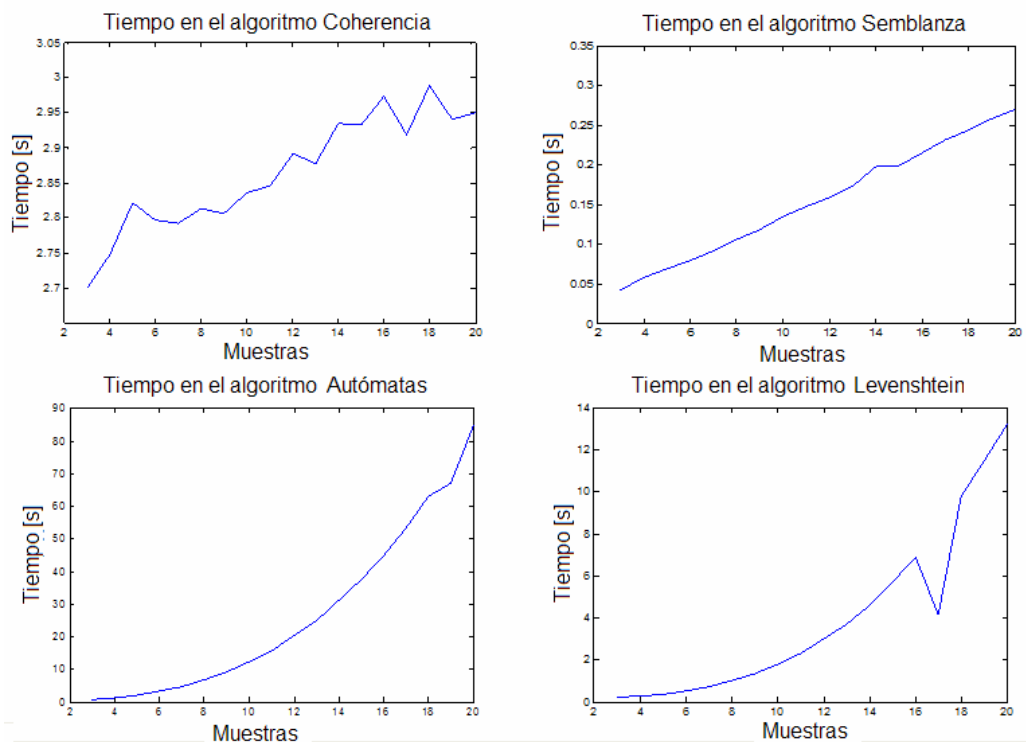


Figura 28. Resultado de rapidez de los algoritmos

4.4. Resultados en datos reales

Las pruebas se realizaron sobre la línea 1440 del Campo Escuela. La versión interpretada corresponde a la Figura 29 (Figura tomada de [30]). Esta imagen es producto del diseño de una estrategia para recobro del campo productor [30]. De la imagen fueron seleccionados los reflectores marcados con color amarillo y color turquesa. Para la prueba, estos reflectores fueron identificados por tramos definidos por las discontinuidades interpretadas en la Figura 29, mediante el uso un algoritmo estadístico y uno sintáctico. En total se utilizaron dos tramos por cada reflector. En la tabla 20 se encuentran los parámetros de entrada para cada tramo:

Algoritmo	Inicio	Final	Intervalo	Traza escogida	Color del reflector	Traza inicial	Traza final
Semblanza	208	212	N. A.	128	blanco	1	298

Semblanza	186	189	N. A.	626	blanco	298	Final
Levenshtein	341	345	25	814	negro	1	897
Leveshtein	264	268	25	927	negro	897	Final

Tabla 20. Parámetros de entrada en identificación de reflectores sobre la línea 1440 del Campo Escuela

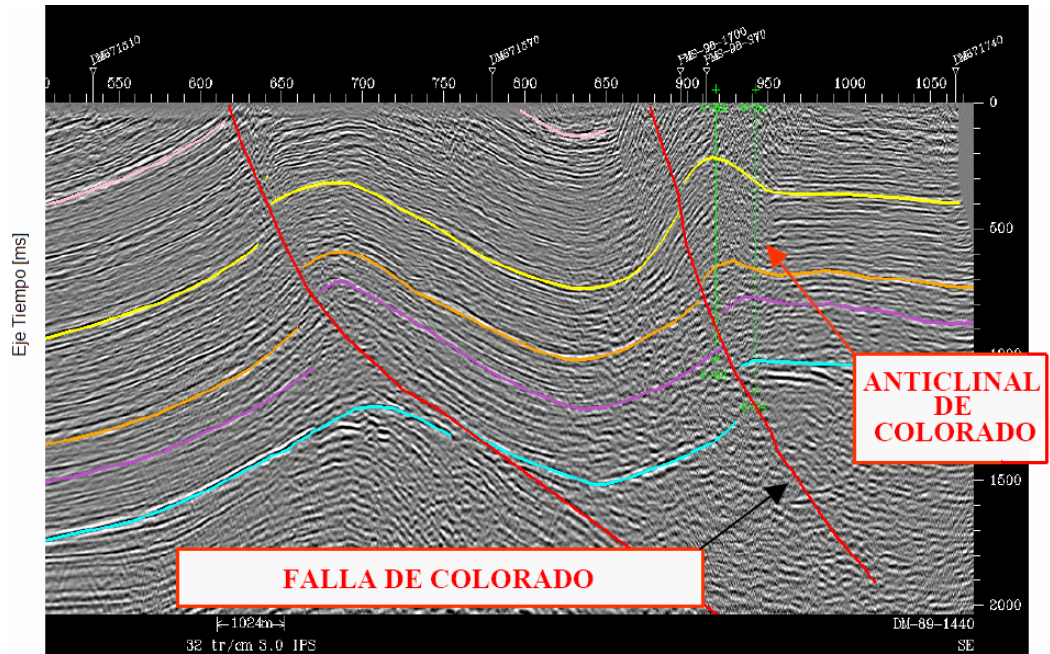


Figura 29. Línea 1440 del Campo Escuela interpretada por un experto

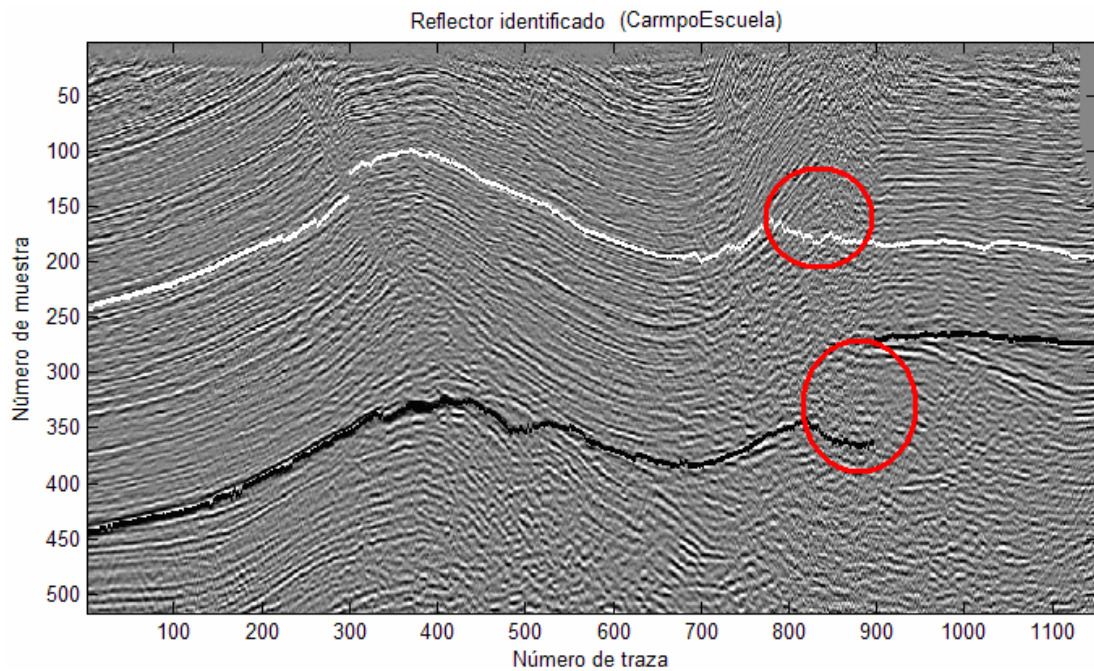


Figura 30. Reflectores identificados por tramos usando Semblanza y Levenshtein

En datos reales no es posible dar una prueba de validez totalmente acertada, ya que la estructura exacta del subsuelo de los sitios donde se realiza sísmica no se conoce con certeza. Lo que se puede hacer es asegurar que el reflector encontrado por los algoritmos es aproximadamente correcto teniendo en cuenta la estructura del lugar dada por un intérprete. En la Figura 30 se observan los reflectores identificados por tramos según los parámetros de la Tabla 20. Los círculos rojos indican las soluciones más erradas de los algoritmos, el resto de los reflectores se encuentran aproximadamente en la misma trayectoria que se muestra en la Figura 29.

En este capítulo se ha evaluado el desempeño de los algoritmos diseñados. Esta evaluación se realizó usando una base datos sintética aprobada por un experto. Las pruebas consistieron en medir la efectividad de los algoritmos cuando se usan para superar discontinuidades de corto desplazamiento. En seguimiento de la trayectoria de un reflector continuo la efectividad es del 100%. En efecto, si se usa un intervalo de evaluación menor al propuesto (tres veces el espesor del reflector) el seguimiento de reflectores continuos es totalmente acertado. Por otra parte la identificación en reflectores discontinuos demostraron que el algoritmo que usa Semblanza como medida de similitud muestra el mayor índice de aciertos con un 97.62% (Tabla 18). El menor valor de aciertos obtenido fue de 83.33% que correspondió al algoritmo que usa criterio de Levenshtein. Es decir que de cada 10 pruebas, por lo menos 8 fueron positivas usando cualquiera de los métodos diseñados. Este resultado demuestra que aunque ésta fue la primera aproximación, los algoritmos sintácticos tienen altas oportunidades en aplicaciones sísmicas.

La prueba de rapidez permitió verificar que el algoritmo que usa Semblanza, además de tener el menor tiempo de cómputo, muestra un crecimiento lineal con respecto del número de muestras por patrón. El método estadístico que usa Coherencia también tuvo un tiempo lineal. Por el contrario, los métodos sintácticos

mostraron un tiempo aproximadamente exponencial. Este hecho constituye una debilidad fuerte en estos métodos.

Finalmente, se realizó una prueba en datos reales que demostró un desempeño satisfactorio de los métodos. Se usó un método sintáctico (Levenshtein) y un método estadístico (Semblanza) por ser los que mostraron los mejores resultados en la imagen de la Línea 1440. Sin embargo, la prueba fue realizada en dos tramos sobre cada reflector, pues la Figura 29 permite observar que en la Línea Sísmica se presentan fallas de alto desplazamiento. Comparativamente, se observa que los reflectores identificados de forma semiautomática coinciden en gran parte con los planteados por [30]. Los errores más notables se encuentran resaltados por los círculos rojos de la Figura 30. En términos prácticos el resultado anterior, junto con el índice de aciertos de la Tabla 18, permite recomendar la implementación de los métodos diseñados en un programa para interpretación sísmica como lo hacen algunas empresas [14], [15] y [16].

5. NUEVAS APLICACIONES DE LOS ALGORITMOS

Durante el desarrollo de este trabajo se encontró que los algoritmos no sólo proponen un camino a seguir en el área de interpretación de datos sísmicos apilados, sino que además pueden ser utilizados por los encargados de realizar el procesamiento de estos datos. Actualmente existen rutinas que requieren grandes volúmenes de tiempo. Son estos procesos los que pueden ser automatizados con el fin de disminuir tiempo. A continuación se presenta una discusión y algunas propuestas sobre las nuevas aplicaciones de los algoritmos.

5.1. Aplicación en identificación de primeros arribos

En un flujo de procesamiento sísmico es necesario obtener los parámetros del modelo de la capa somera (capa más cerca de la superficie). La energía refractada en la base de ésta capa es usualmente la que produce los primeros arribos en una imagen de disparo [28]. Los primeros arribos dependen de la fuente usada en la adquisición.

Una vez es identificada la trayectoria de los primeros arribos en la imagen de disparo, se pueden encontrar tiempos para realizar una corrección estática que permita adecuar la imagen para el resto del flujo sísmico [28]. Esta corrección consiste en llevar los primeros arribos a una línea recta inclinada. En la mayoría de imágenes de disparo, después de realizar la corrección por tiempo de los primeros arribos, los reflectores se visualizan en formas hiperbólicas. Este proceso debe ser realizado en la totalidad de las imágenes de disparo (usualmente del orden de centenas en cada línea sísmica).

Un geofísico debe invertir gran parte del tiempo en el proceso de identificación de primeros arribos para poder continuar con el procesamiento sísmico. Por esto se plantea la aplicación de los algoritmos para reconocimiento semiautomático de reflectores en identificación de primeros arribos. El proceso planteado fue probado

en una imagen real de disparo del área de guariqués. El resultado es mostrado en la Figura 31.

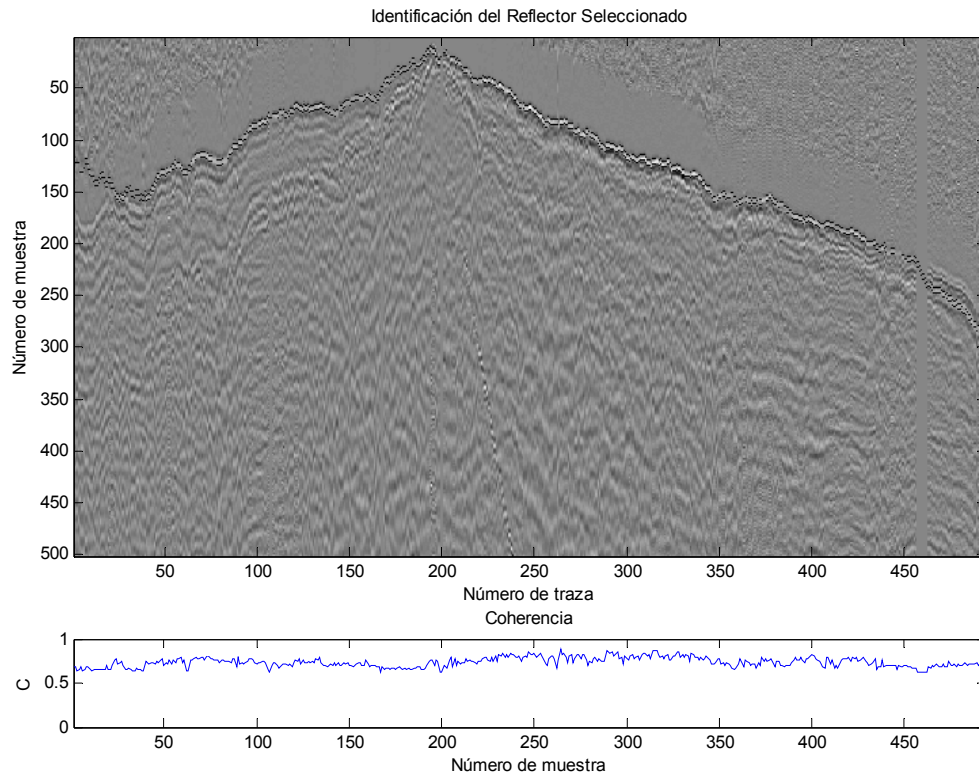


Figura 31. Primeros arribos identificados en una imagen de disparo.

En la Figura 31 se observa la trayectoria de los primeros arribos correctamente identificada en la mayoría de la imagen usando el algoritmo de coherencia. El algoritmo únicamente falló en el inicio de la imagen (izquierda de la imagen), donde las trazas sísmicas pierden gran parte de su amplitud por ser las más alejadas de la fuente. En las últimas trazas (derecha de la imagen) se presentó un error generado por las trazas muertas que se observan. Estas trazas son identificables porque en ellas no se encuentra una señal variante con el tiempo y se debe a sensores que fallaron en su momento. Los parámetros usados en la prueba son los siguientes: Inicio 69, Fin 74, Intervalo 500, Traza 250.

El resultado presentado confirma la utilidad de los algoritmos planteados en correcciones estáticas. Se recomienda continuar con el estudio de esta nueva

aplicación y llevar los algoritmos a un programa de procesamiento sísmico programable como *PROMAX* [29].

5.2. Aplicación en corrección de velocidades por NMO

Las siglas **NMO** significan *Normal-Moveout Correction*, y es una de las etapas que deben ser realizadas en todo flujo de procesamiento sísmico (Figura 4). Consiste en corregir la curvatura hiperbólica que adquieren las trayectorias de los reflectores a los largo de las trazas sísmicas. Esta curvatura se debe al efecto del *offset* en el tiempo de viaje de la onda [28]. El *offset* es la distancia entre la fuente y el receptor. El objetivo de la corrección por **NMO** es llevar los reflectores hiperbólicos a una forma horizontal usando el campo de las velocidades de viaje de las ondas en el subsuelo de la línea sísmica. Este campo de velocidades es replanteado después de realizar la corrección en una imagen como la mostrada en la Figura 32. Se realiza un proceso iterativo que da como resultado una imagen con los reflectores horizontales y un campo de velocidades aproximadamente cercano al real. Después de tener las imágenes corregidas, cada una es apilada (sumada) para formar una traza sísmica. En una imagen apilada como la mostrada en la Figura 5, cada traza es el resultado de la suma de trazas corregidas por **NMO**.

En este proceso también es necesario plantear un algoritmo que identifique semi-automáticamente la trayectoria del reflector y de esta forma llegar a dos objetivos:

- 1) Disminuir la cantidad de iteraciones: considerando la cantidad de imágenes en las cuales se debe corregir por **NMO**. Esto representaría un ahorro en el tiempo que lleva procesar cada línea sísmica.
- 2) Encontrar la trayectoria exacta del reflector: las trayectorias siempre son aproximadas a hipérbolas, con la identificación semiautomática se podría

encontrar la trayectoria del reflector sin necesidad de utilizar una aproximación. Esto puede generar mejores resultados en el campo de velocidades.

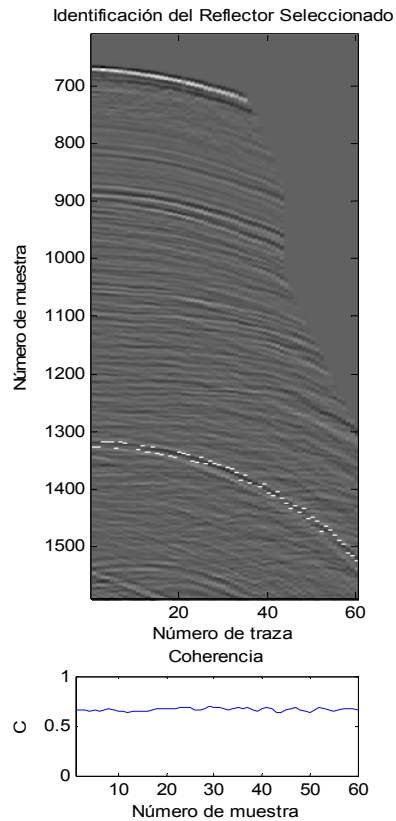


Figura 32. Identificación de un reflector en una imagen para NMO

En la Figura 32 se observa la trayectoria del reflector marcado entre las trazas 1320 y 1330. Después de encontrar esta trayectoria, se comienza la iteración para encontrar el campo de velocidades. De esta forma se propone continuar el estudio y llevar los códigos a un programa para procesamiento sísmico como *PROMAX* [29].

5.3. Aplicación en identificación de curvas de emisividad

Una aplicación no relacionada con el campo de la sísmica es el de transferencia de calor. Como ejemplo de esta aplicación se muestra la Figura 33, la cual sirve

para resolver problemas físicos en transferencia de calor. Esta gráfica corresponde a la emisividad del dióxido de carbono en función de la presión y la temperatura. Si se pretende crear una base de datos con la información que allí se presenta, se puede utilizar alguno de los algoritmos para identificar la curva deseada y luego aplicar un proceso de normalización a los ejes.

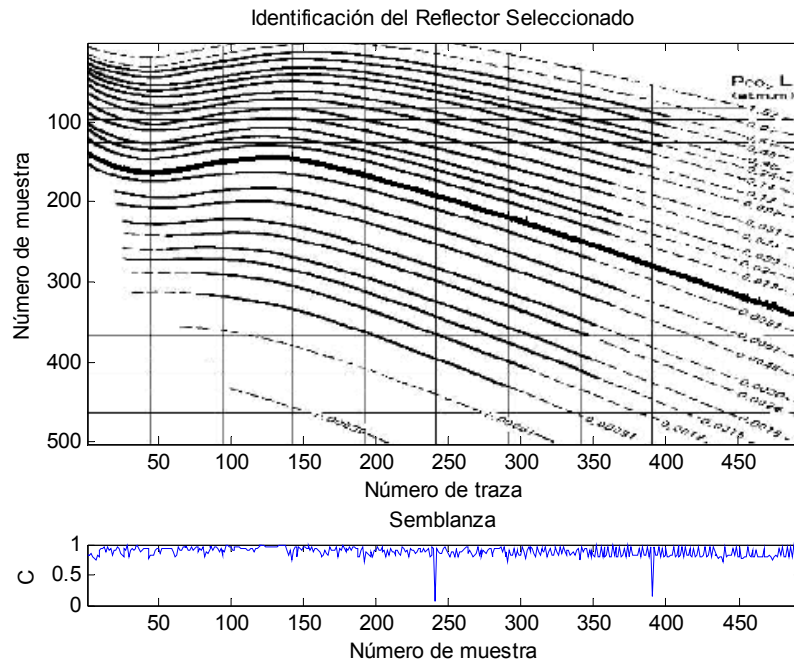


Figura 33. Nueva aplicación en curvas de emisividad de gases

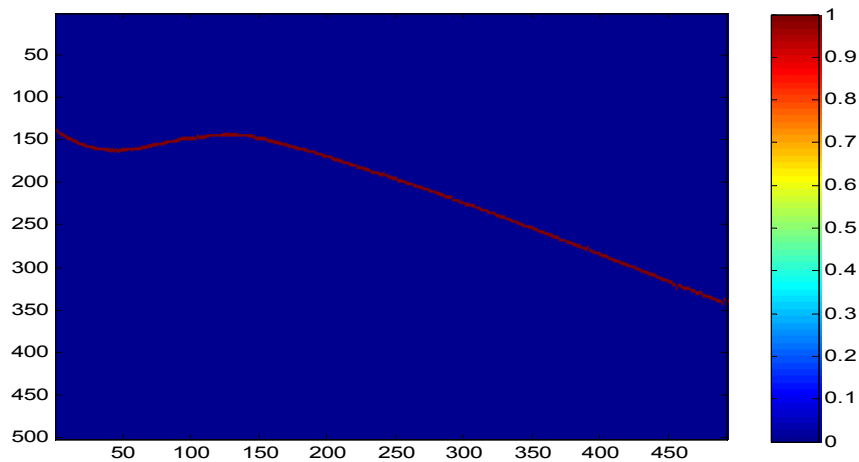


Figura 34. Curva identificada en la grafica de emisividad

Las nuevas aplicaciones propuestas hasta ahora en este capítulo no son las únicas. En sismica existe una gran cantidad de procesos en los cuales se desea encontrar similitudes entre grupos de muestras pertenecientes a las trazas de una imagen. El trabajo en identificación de reflectores pretende ser una ventana innovadora hacia el trabajo en reconocimiento de similitudes usando métodos no comunes como la teoría sintáctica. Sin embargo, para realizar este estudio fue necesario comparar el comportamiento con métodos conocidos, como los estadísticos.

6. CONCLUSIONES Y RECOMENDACIONES

6.1. Conclusiones

- Una imagen sísmica apilada que permita observar la litología terrestre, con una baja componente de ruido, permite realizar una interpretación más aproximada de la situación real del subsuelo. Por esta razón el éxito en sísmica comienza con la eficacia en la realización de la adquisición y el procesamiento de datos y termina con una buena caracterización de reflectores y fallas en la interpretación de la imagen apilada.
- Se han propuesto cuatro flujos de proceso para la identificación semiautomática de reflectores, dos estadísticos y dos sintácticos. Los métodos estadísticos se basan en fórmulas para encontrar un coeficiente de similitud entre una traza de evaluación y una traza o una ventana de trazas como patrón. Los coeficientes utilizados son coherencia y semblanza. Por el contrario, los métodos sintácticos se basan en un complejo flujo de procesamiento que pretende encontrar la señal con menor costo de producción respecto de un patrón. En el método que usa autómatas, el patrón sirve para inferir una gramática de estado finito con errores de transformación. Estas transformaciones permiten producir cualquier cadena de caracteres con un costo específico. La información que brinda un autómata es la misma que contiene una gramática y es describible, de la misma manera, mediante una matriz tridimensional de costos de transición entre estados. El criterio de Levenshtein no necesita de la inferencia de gramática, pues éste encuentra el costo de producción directamente de la cadena de caracteres patrón mediante una medida de mínima distancia. Los algoritmos propuestos para identificación semiautomática, constituyen un diseño original del presente trabajo basado en métodos propuestos en trabajos pasados.
- Los métodos propuestos para el cumplimiento del objetivo de este trabajo, son basados en la teoría presentada para reconocimiento de patrones. Los diseños originales se ven plasmados en los algoritmos del Capítulo 4. El espacio de

búsqueda encontrado como ideal es igual a tres veces el espesor del reflector patrón. Este espacio de búsqueda se basa en el principio geológico de que un reflector no se desplaza una distancia mayor a su espesor por más inclinado que éste sea. En discontinuidades, las que son del tipo de corto desplazamiento son el objetivo que se planteó como umbral de evaluación. Sin embargo, en la herramienta programada ARI se dejó el espacio abierto para la prueba con otros intervalos de búsqueda.

- Los algoritmos originales tomados de [2] fueron modificados con el objetivo de hacer más robusta la herramienta ante errores de transformación. La tabla de costos propuesta (Tabla 3) se basa en el principio de ser menos costosa la sustitución de una pendiente por su contigua (según el código de Freeman) que por una más alejada. Por otra parte, si se cambian los costos de sustitución, también se hace necesario cambiar los costos de inserción y borrado por un valor medio como 3. De esta forma se evita el proceso que obtiene un costo menor borrando e insertando un terminal determinado que sustituyéndolo.
- El algoritmo de semejanza fue el más acertado en el seguimiento de reflectores discontinuos en imágenes sísmicas, con un 97.62% de aciertos sobre el total de pruebas. Sin embargo los demás algoritmos mostraron un comportamiento relativamente bueno, el porcentaje más bajo de acierto fue de 83.33% en el algoritmo que aplica criterio de Levenshtein. Es decir que si se realiza un análisis de 10 imágenes, por lo menos 8 de ellas saldrán positivas usando cualquier método.
- La prueba de rapidez constituye una confirmación de que el método que usa coeficiente de semejanza como prueba de similitud es el método más robusto. Obtuvo una gráfica de rapidez con promedio de décimas de segundos, mientras que el resto de los algoritmos obtuvieron tiempos del orden de segundos y decenas de segundos. De hecho, los algoritmos sintácticos obtuvieron un crecimiento aproximadamente exponencial, lo que implica cierta ineficiencia en la complejidad espacial de estos métodos. Sólo los métodos estadísticos obtuvieron un comportamiento aproximadamente lineal. En

aplicaciones en las cuales se necesite rapidez, los métodos estadísticos constituyen la mejor solución

- La prueba en datos reales promueve la utilización de los algoritmos en herramientas programables para interpretación sísmica. Los resultados constataron que los reflectores identificados de manera semiautomática coinciden en gran medida con los identificados por un intérprete.
- A pesar de que el presente trabajo es una primera aproximación en el uso de métodos para reconocimiento sintáctico de patrones en identificación de reflectores sísmicos, los resultados obtenidos muestran el potencial de esta teoría en la Geofísica y en el procesamiento de imágenes y señales.
- Bajo parámetros adecuados de entrada, los algoritmos pueden ser utilizados en la identificación de reflectores continuos con un porcentaje de veracidad aproximadamente cercano al 100%. En otras aplicaciones en las cuales se desee seguir la trayectoria de una línea continua presente en una imagen, los algoritmos propuestos pueden ser eficientemente utilizados. La aplicación en seguimiento de curvas de emisividad comprueba esta afirmación.

6.2. Recomendaciones

En los algoritmos para reconocimiento sintáctico se han observado algunas falencias que pueden ser mejoradas en trabajo futuros. Los tópicos sugeridos son los siguientes:

- La codificación de Freeman propuesta para convertir señales discretas numéricas a cadenas de caracteres involucran sólo 9 símbolos terminales que encasillan las pendientes entre puntos adyacentes en intervalos crecientes en forma exponencial. En trabajos futuros, se recomienda explorar codificación con un número mayor de símbolos terminales. De igual forma, probar diferentes intervalos como crecientes lineales, en polinomios, etc. La codificación puede ser dependiente del contexto haciendo que cada código

deba ser asignado de acuerdo a la pendiente entre puntos adyacentes y las pendientes anterior y siguiente.

- En el algoritmo para inferencia de la gramática canónica por su modo de operación genera más reglas que las necesarias. Como ejemplo se puede encontrar la gramática canónica de las palabras 'abcd' y 'abce'. La función en MATLAB® *string_r_n2* que opera según el algoritmo 3.6 encuentra la gramática canónica inferida de la siguiente manera:

```
>> [VNn,VTn,Pn,Sn] = canonical_n2([[1; 2; 3; 4] [1; 2; 3; 5]])
```

```
VNn =
```

```
1 2 3 4 5 6 7
```

```
VTn =
```

```
1 2 3 4 5
```

```
Pn =
```

```
1 1 2  
2 2 3  
3 3 4  
4 4 0  
1 1 5  
5 2 6  
6 3 7  
7 5 0
```

```
Sn =
```

```
1
```

El conjunto de reglas en P_n según la programación de la función en MATLAB® representa el siguiente conjunto de reglas:

P' :
 $S \rightarrow aA$
 $A \rightarrow bB$
 $B \rightarrow cC$
 $C \rightarrow d$
 $S \rightarrow aD$
 $D \rightarrow bE$
 $E \rightarrow cF$
 $F \rightarrow e$

El conjunto óptimo de reglas es el siguiente:

P' :
 $S \rightarrow aA$
 $A \rightarrow bB$
 $B \rightarrow cC$
 $C \rightarrow d$
 $C \rightarrow e$

En futuras aplicaciones se pensará en crear gramáticas canónicas respecto de un número mayor de trazas guía. Cuando se realice tal tarea será útil optimizar el algoritmo de inferencia de gramática.

6.3. Trabajos Futuros

Para futuros trabajos en reconocimiento automático de reflectores se propone tener en cuenta los siguientes tópicos de investigación:

- En reflectores discontinuos se propone reconocer la discontinuidad por medio de cambios notables en el valor de coherencia o semblanza. Luego de tener las fallas en una base de datos, automáticamente el algoritmo deberá prestar una mayor atención a las trazas con posibilidad de falla. En estas trazas se

podrá aplicar por ejemplo un algoritmo elaborado especialmente para probar un mayor número de combinaciones para encontrar la mejor.

- Los reflectores continuos pueden ser identificados en la mayor parte de la imagen. Después de esto, para superar la falla e incluso caracterizarla se puede usar un algoritmo genético como se propone en [20]. Sin embargo, un enfoque que continúe la línea de reconocimiento sintáctico podría servir en esta tarea usando gramáticas de árbol o el algoritmo para MPM (*Match primitive measure*) [2].

REFERENCIAS

- [1] F. Gómez, I. D. Olaya, C. Piedrahita, J. C. Ramón. *El método sísmico en la exploración de recursos minerales e hidrocarburos*. Facultad de Ciencias Universidad Nacional, Medellín. 1990. ECOPETROL Instituto Colombiano del Petróleo ICP. 190 Págs
- [2] K. Y. Huang. *Syntactic Pattern Recognition for Seismic Oil Exploration*. Series in Machine Perception and Artificial Intelligence – Vol. 46.
- [3] M. Bahorich, S. Farmer. *The Coherence Cube*. The Leading Edge 14(10). 1053-1058, oct 1995.
- [4] E. J. W. Jones. *Marine geophysics*. Wiley 1999.
- [5] K. S. Fu. *Syntactic Pattern Recognition and Applications*. Prentice-Hall, Englewood Cliffs, NJ, 1982.
- [6] K. Y. Huang, K. S. Fu. *Syntactic Pattern Recognition for the Classification of Ricker Wavlets*. Geophys. 50. 1548-1555. 1985.
- [7] R. A. Wagner, M. J. Fisher. *The string to string correction problem*. JACM 21. 168-173. 1974.
- [8] N. Neidell, M. Taner. *Semblance and other coherency measures for multichannel data*. Geophysics 36(3). 482-497, jun 1971.
- [9] H. Gish, D. Cochran. *Generalized Coherence*. Acustics, Speech, and Signal Processing. IEEE International Conference on. Published: 1988. Volume 5. Pages: 2745-2748.

- [10] K. Marfurt, V. Sudhaker, A. Gersztenkorn, K. Crawford, S. Nissen. *Coherency calculations in the presence of structural dip*. Geophysics, 64(1):104-111, ene 1999.
- [11] W. A. Gardner. *Exploitation of spectral correlation in cyclostationary signals* Spectrum Estimation and Modeling, Fourth annual ASSP workshop on. Published: 1988. Pages: 1-6.
- [12] S. C. Chapra, R. P. Canale. *Métodos numéricos para ingenieros*. Ed. Mexico: McGraw Hill. 1999
- [13] M. Tanner. *Semblance and other similarity measurements*. Rock Solid Images, nov 1996.
- [14] <http://www.slb.com/content/services/software/geo/petrel/index.asp>
Shlumberger.
- [15] <http://www.cwp.mines.edu/cwpcodes/> Center Wave of Phenomena.
Colorado School of Mines
- [16] <http://www.halliburton.com/ps/Default.aspx?navid=220&pageid=876&prodid=PRNaa110262578211478> Halliburton.
- [17] M. C. Quijano. *Detección de discontinuidades sísmicas a partir de datos sísmicos 2D*. Grupo de investigación Petrosísmica. Informe de avance No. 2. ECOPETROL Instituto Colombiano del Petróleo ICP. 2005.
- [18] R. E. Sheriff, I. P. Geldart. *Exploración Sismológica: Procesamiento e interpretación de datos*. Vol. II. Limusa. 275. 1991.

- [19] R. McQuillin, M. Bacon, W. Barclay. *An introduction to seismic interpretation*. Gulf Publishing Company. 1984.
- [20] M. Aurnhammer and K. Tönnies. *A genetic algorithm for automated horizon correlation across faults in seismic images*. IEEE Transactions on evolutionary computation, Vol. 9, No. 2, April 2005.
- [21] F. Ali and T. Pavdilis. *Syntactic recognition of handwritten numerals*. IEEE Transactions on Syst. Man Cybern. 7 (1977) 537-541.
- [22] H. Bunke and A. Sanfeliu. *Syntactic and structural pattern recognition – Theory and applications*. World Scientific, 1990.
- [23] A. V. Aho, T. G. Peterson. *A minimum distance error-correcting parser for context-free languages*. SIAM J. Comput. 1. 305-312. 1972.
- [24] R. E. Sheriff. *Geophysical methods*. Prentice Hall 1989.
- [25] C. A. Pérez, G. Y. Ojeda. *Identificación de reflectores en imágenes sísmicas aplicando coherencia temporal*. XII Simposio de Tratamiento de Señales, Imágenes y Visión Artificial. Barranquilla 2007. ISBN 978-958-8252-59-9.
- [26] P. Bakker. *Image Structure Analysis for seismic interpretation*. PhD. Thesis. Technische Universiteit Delft, 2002.
- [27] M.D. Sacchi. <http://www-geo.phys.ualberta.ca/saig/SeismicLab/index.html> 2006.
- [28] Öz Yilmaz. *Seismic data analysis, processing, inversion, and Interpretation of seismic data*. Volumen I. Society of exploration Geophysicists, 2001.

[29] LANDMARK. ProMAX Version 6.0. Training manual. 1995.

[30] ECOPETROL S.A. *Diagnóstico y estrategias de recobro Campo Colorado*.
Gerencia Magdalena Medio. Instituto Colombiano del Petróleo. 2003

ANEXO. MANUAL DE LA HERRAMIENTA COMPUTACIONAL *ARI*

Manual de *ARI* (Identificación automática de reflectores)

Cargar imagen

Selección del reflector

Selección del algoritmo y el color

Selección de opciones adicionales

Comenzar el análisis

Resultados de salida

Cargar imagen

Para cargar la imagen se debe hacer clic en el botón “Cargar Imagen”. El usuario puede elegir imágenes en los formatos: *.jpg*, *.gif*, *.tif*, *.bmp* y los estándares internacionales para datos sísmicos *.seg*y y *.sgy*.

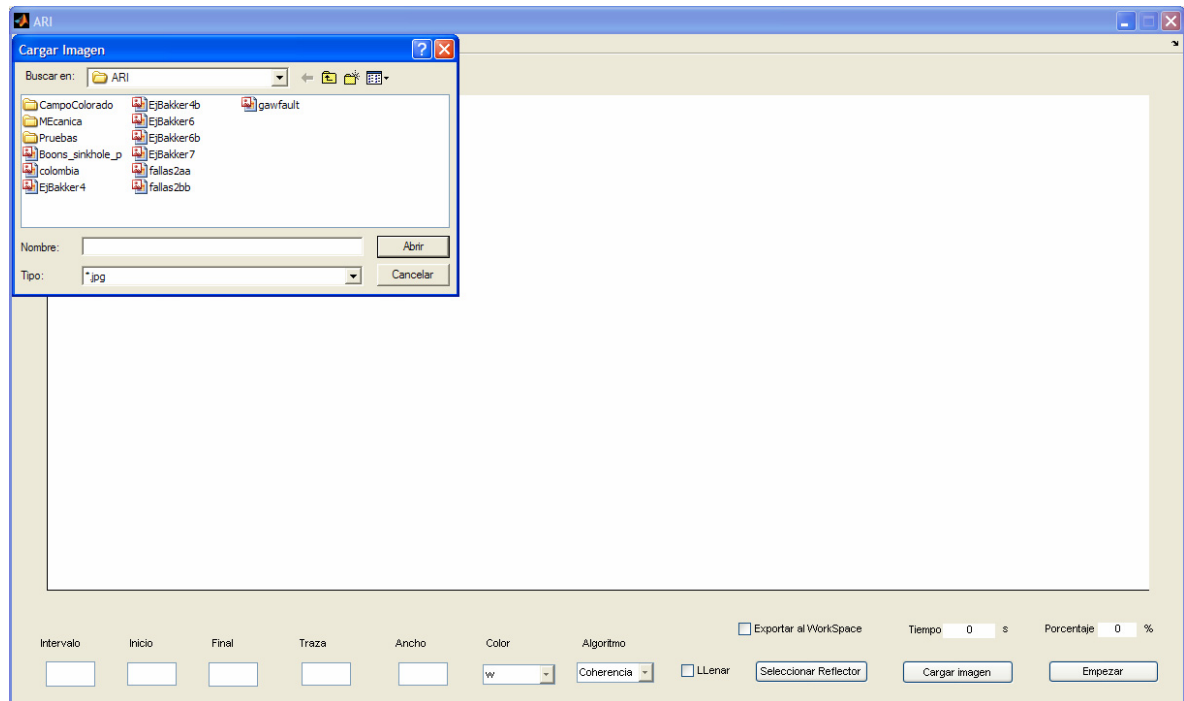


Figura 35. Cargar una imagen en la herramienta *ARI*

Después de cargar la imagen, ésta aparecerá en la parte central de la interfaz.

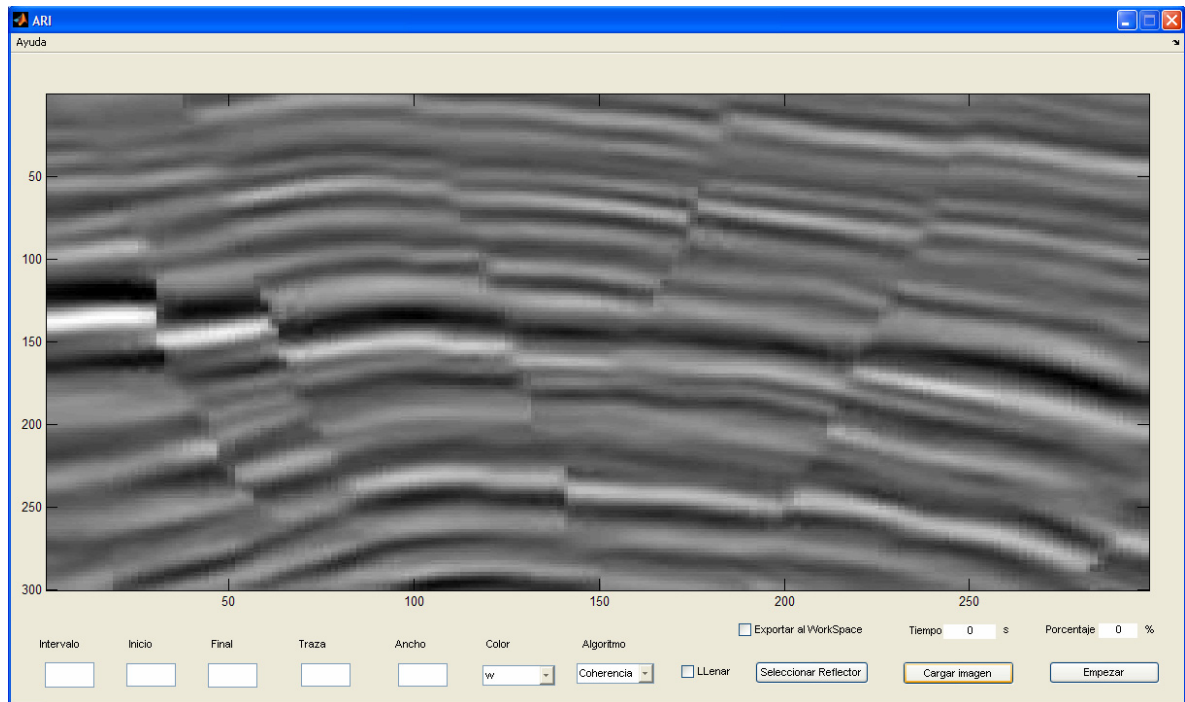


Figura 36. Imagen sísmica cargada en la interfaz ARI

Selección del reflector

Para seleccionar el reflector patrón se puede elegir uno de dos métodos:

1. Dar clic en el botón “Seleccionar Reflector”. Una nueva figura aparecerá (Figura 37). En esta nueva figura se debe dar clic en el botón “Seleccionar” y utilizar el puntero del ratón para seleccionar la ventana donde se encuentra el reflector deseado. Después de realizada la selección se debe dar clic en “Aceptar” para guardar la ventana seleccionada o dar “Cancelar” para rechazar los cambios. En la interfaz ARI quedarán registrados los siguientes valores: “Inicio” como la muestra inicial del reflector seleccionado, “Final” como la muestra final del reflector seleccionado “Traza” como la primera traza a la izquierda de la ventana seleccionada por medio del ratón. El parámetro

“Intervalo” será llenado por defecto con el número 25. El parámetro “Ancho” será llenado por defecto con el número 1.

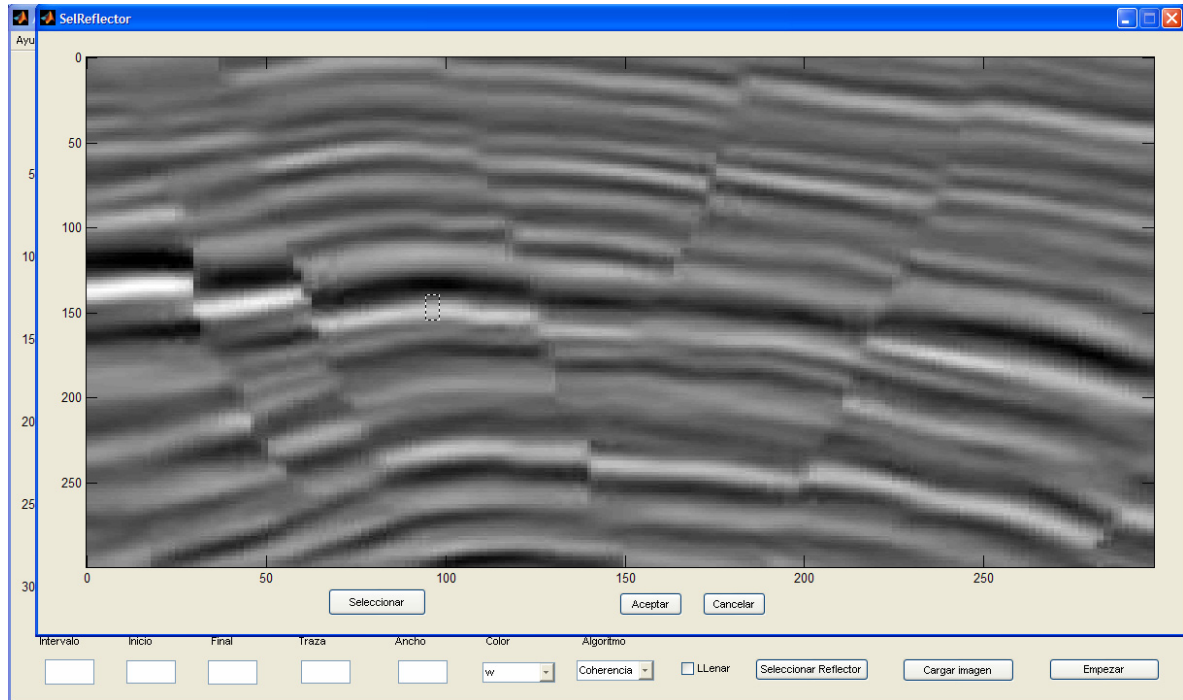


Figura 37. Interfaz SelReflector para seleccionar la ventana del reflector deseado

2. Ingresar los parámetros descritos en el numeral anterior de forma manual.

El parámetro “Ancho” define el espacio de evaluación en la traza siguiente. Este espacio se halla mediante: $Espacio = (2 * Ancho + 1) * (Final - Inicio)$. De ésta forma, si se ingresa el valor 1, el espacio de búsqueda será 3 veces el espesor del reflector.

Después de ingresar los parámetros, en la interfaz ARI aparecerá la traza definida por estos, matizada con líneas negras. El espacio de evaluación en la siguiente traza a la escogida como patrón, aparecerá limitada por puntos negros (Figura 38).

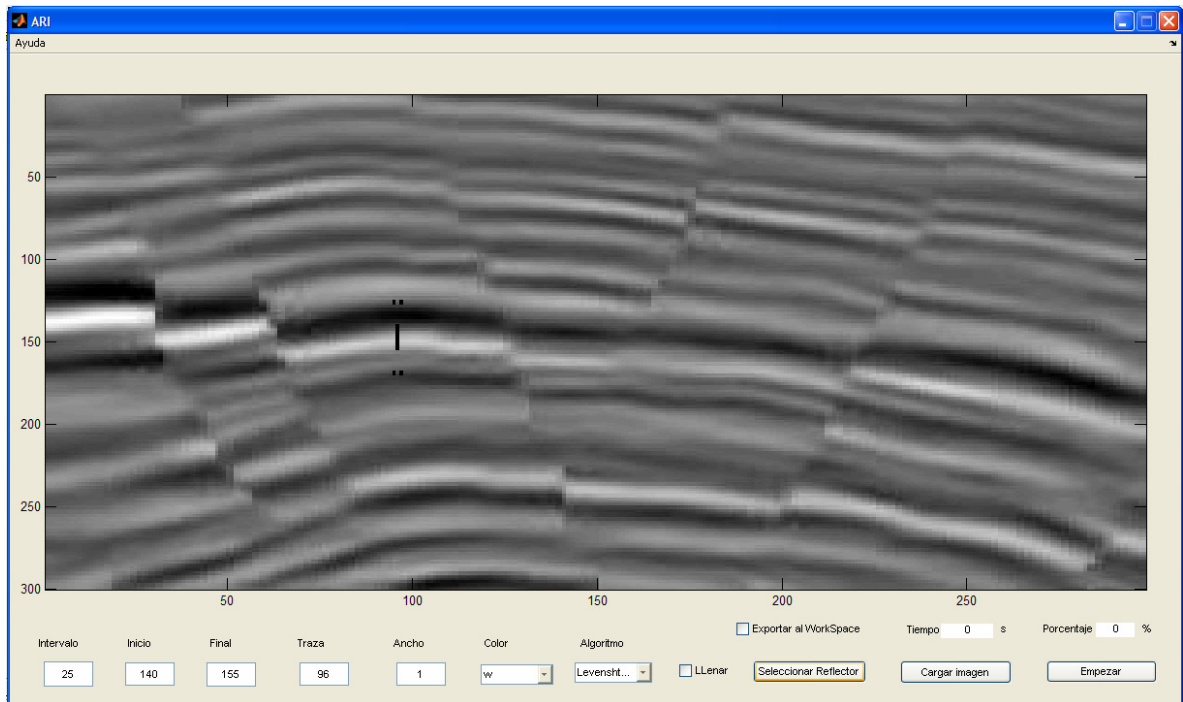


Figura 38. Ventana seleccionada limitada por líneas negras en la interfaz *ARI*

Selección del algoritmo y el color

Es posible elegir uno de cuatro algoritmos basados en medidas para encontrar similitudes, o escoger los cuatro métodos a la vez. El color con el que es marcado el reflector puede ser escogido entre negro o blanco. También puede ser seleccionada la forma de visualización entre llenar la totalidad del reflector, o simplemente marcar los límites del mismo. Lo anterior se escoge mediante el chequeo del elemento “Llenar”.

Selección de opciones adicionales

Los datos de salida pueden ser exportados al *workspace* chequeando la caja “Exportar al *workspace*”.

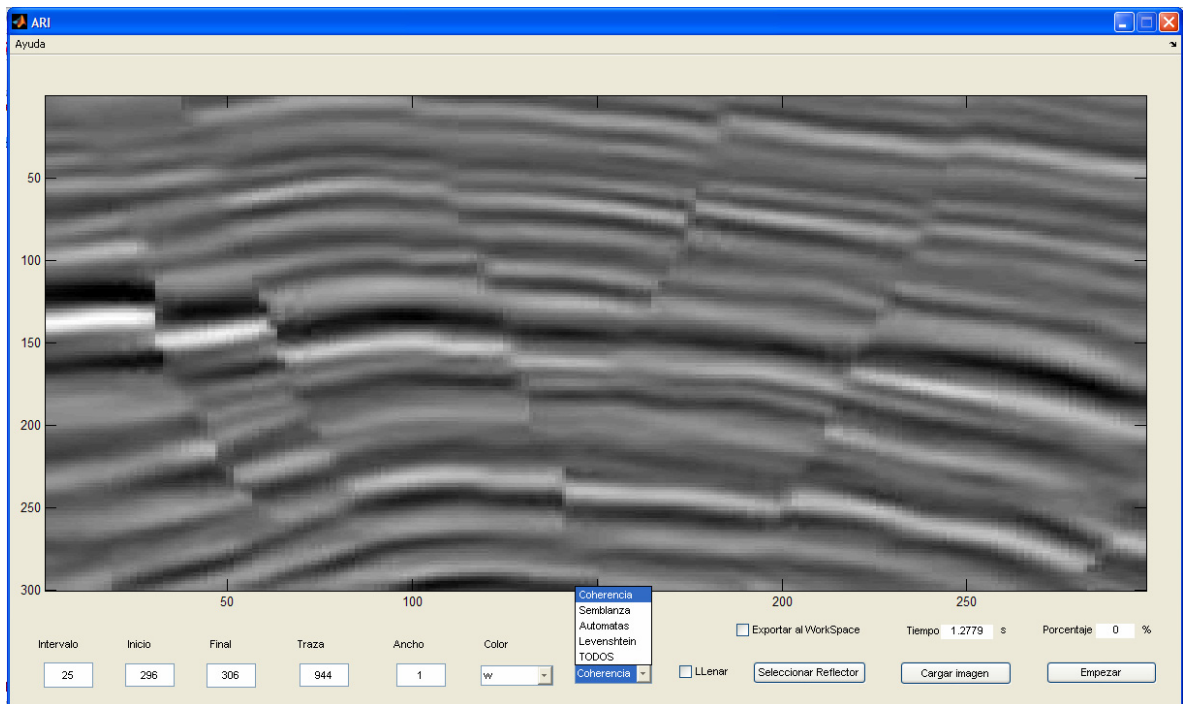


Figura 39. Selección del algoritmo en la interfaz ARI

Comenzar el análisis

Una vez cargada la imagen y los parámetros de entrada, se puede comenzar el análisis dando clic en el botón “Empezar”. El resultado será dibujado en una nueva imagen junto con la gráfica de coeficientes que resultaron del análisis con cada método a lo largo de la imagen.

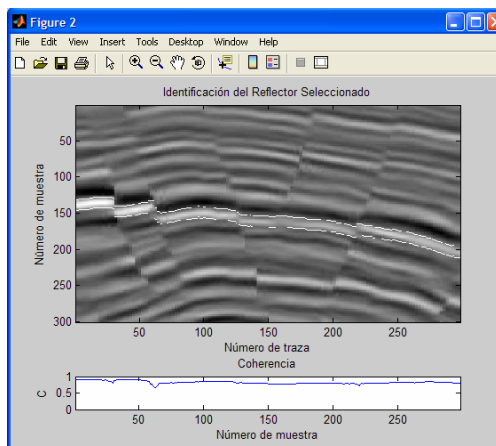


Figura 40. Imagen con el resultado encontrado por ARI