

3

O Sistema Recon

3.1. Introdução

A proposta do presente capítulo é descrever o sistema que serve de ambiente para o desenvolvimento deste trabalho, ou seja, o sistema *Recon*. Pode-se subdividir este capítulo, então, em três partes distintas. A primeira parte se destina a apresentação da estrutura de dados utilizada pelo sistema para representar as seções geológicas. A seguir, descreve-se as principais funcionalidades do sistema, básicas para a realização de um balanceamento de uma determinada seção geológica. Em uma terceira etapa são descritos o que foi desenvolvido dentro do sistema para torná-lo mais funcional para o geólogo ou geofísico.

O Sistema *Recon* vem sendo desenvolvido pelo convênio entre o Tecgraf/PUC-Rio e a PETROBRAS desde 1991. O sistema utiliza uma tecnologia de modelagem geométrica baseada em uma estrutura de dados topológica, uma biblioteca também desenvolvida pelo Tecgraf. Esta tecnologia faz com que o *Recon* tenha não só uma edição gráfica inteligente, mas muitas opções de deformação estrutural (transformações geométricas) que no conjunto facilitam bastante o processo de balanceamento geológico. O sistema já foi aplicado nas bacias de Campos, Recôncavo, Potiguar e Sergipe entre outras.

O Recon permite testar a validação geométrica de seções geológicas, assim como permite a construção de modelos geológicos mais confiáveis, constituindo-se assim numa importante ferramenta exploratória.

3.2. Subdivisão Planar

As estruturas representadas em uma seção geológica podem ser vistas como blocos que se deslocam e deformam geometricamente, mas mantêm entre si uma relação de adjacência durante estas transformações. Uma estrutura de dados que seja capaz de representar estas estruturas geológicas deverá manter as relações de topologia ao longo de todo o processo de balanceamento. A estrutura, além de manter a representação da subdivisão do plano que modela a seção, deve também facilitar o processo de criação da mesma, já que todo este processo de modelagem geométrica deverá se passar em tempo real.

A seção geológica pode ser vista como uma subdivisão planar, pois pode ser estabelecida uma relação entre as entidades geológicas e as entidades topológicas da subdivisão. Na Figura 3.1 é apresentada uma seção. A camada A está fragmentada em três blocos (separados por falhas), blocos A1, A2 e A3. O horizonte A (topo da camada A) é formado por linhas no contorno destes blocos. A cada entidade podem ser associados atributos, tais como idade, litologia, porosidade, etc. A estrutura de dados deve permitir o armazenamento destes atributos.

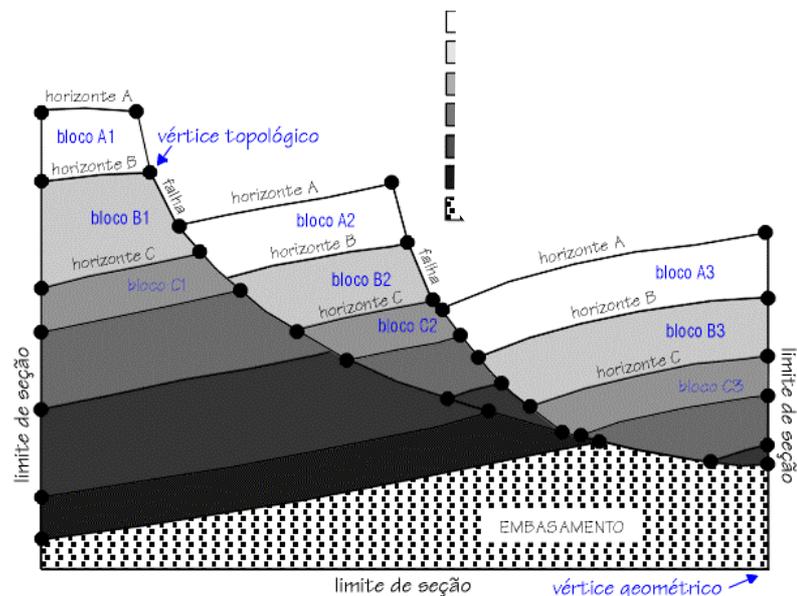


Figura 3.1 –Entidades topológicas de uma seção geológica.

A estrutura de dados utilizada pelo *Recon* é baseada no conceito de subdivisões planares. Uma subdivisão do \mathbf{R}^2 é apresentada na Figura 3.2 abaixo. Nela o plano é dividido em diversas regiões (faces f_1, f_2, f_3 e a face externa), sendo cada uma delas delimitada por um conjunto de segmentos de curva (arestas) que, por sua vez, são delimitados por dois vértices não necessariamente distintos. Os termos vértice, aresta e face são provenientes do fato de que subdivisões planares representam a topologia da fronteira de sólidos homeomorfos a uma esfera [9], [10].

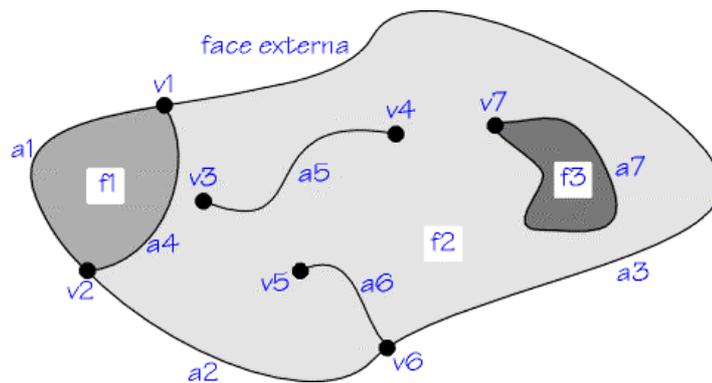


Figura 3.2 -Exemplo de uma subdivisão do \mathbf{R}^2 .

No processo de criação e manutenção de uma subdivisão planar, quando um segmento de curva for incluído, todos os pontos de interseção com os segmentos já pertencentes à subdivisão são calculados. Estes pontos se tornam vértices topológicos e são criadas arestas entre cada um desses vértices. No caso de regiões fechadas por arestas se formarem, novas faces são criadas na subdivisão. Os componentes topológicos que formam uma subdivisão planar são dos seguintes tipos:

- **vértice** - correspondente a um ponto único do \mathbf{R}^2 , não podendo haver dois vértices com a mesma localização geométrica.
- **aresta** - segmento de curva (equivalente homeomorficamente a um segmento de reta) delimitado por dois vértices não necessariamente distintos.
- **ciclo** (loop) - subconjunto conexo e ordenado de vértices e arestas alternados.
- **face** - subconjunto conexo do \mathbf{R}^2 , cuja fronteira é formada por um ou mais ciclos, sendo que, com exceção da face externa, um destes contém os outros e representa sua fronteira externa.

As estruturas de representação podem ser baseadas em vértices, faces ou arestas. Mas, pela sua própria natureza, as baseadas em arestas possuem algumas propriedades que as tornam mais vantajosas do que as baseadas em vértices ou faces. Em uma representação baseada em arestas, cada aresta é limitada por exatamente dois vértices não necessariamente distintos, e cada aresta é adjacente a exatamente duas faces não necessariamente distintas. Além disso, se for considerado que a adjacência aresta-aresta é caracterizada por arestas que compartilham tanto vértices quanto faces, o número de arestas adjacentes a uma dada aresta é limitado a quatro. Estes fatos são explorados em uma estrutura de dados baseada em arestas, resultando em uma estrutura com registros de tamanhos fixos. Cada aresta tem dois usos, que são definidos pelo uso da aresta pelos ciclos adjacentes. Os usos de uma mesma aresta estão orientados em sentidos opostos, levando automaticamente a uma orientação para as faces. Na Figura 3.3, vê-se uma subdivisão planar com a indicação dos ciclos de cada face. Uma das faces possui um ciclo interno. Também é indicado em tracejado a inclusão de uma nova aresta com a conseqüente adição de um ciclo e uma face à subdivisão. Tal como os usos das arestas, os ciclos têm orientação, e neste exemplo, no sentido horário. O uso de uma aresta é o lado de uma aresta visto por uma das duas faces. Os ciclos são compostos pelo encadeamento dos usos das arestas de uma face.

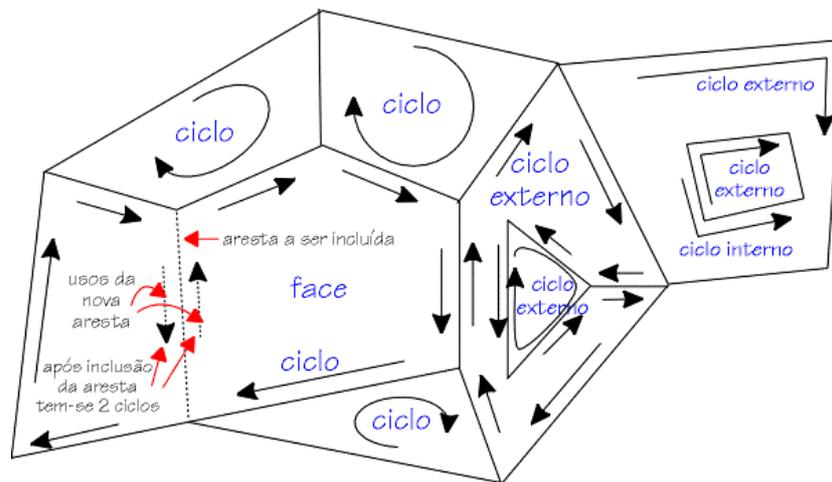


Figura 3.3 - Orientação de usos e ciclos em uma subdivisão planar.

3.3. Modelagem da Subdivisão Planar

No *Recon* a modelagem da subdivisão planar utiliza a biblioteca de rotinas HED desenvolvida pelo Tecgraf/PUC-Rio. O HED é uma estrutura de dados topológica, baseada em arestas, do tipo *half-edge* [40]. Uma estrutura do tipo *half-edge* é uma extensão de uma outra, também baseada em arestas, denominada *winged-edge* e que é sucintamente ilustrada na Figura 3.4.

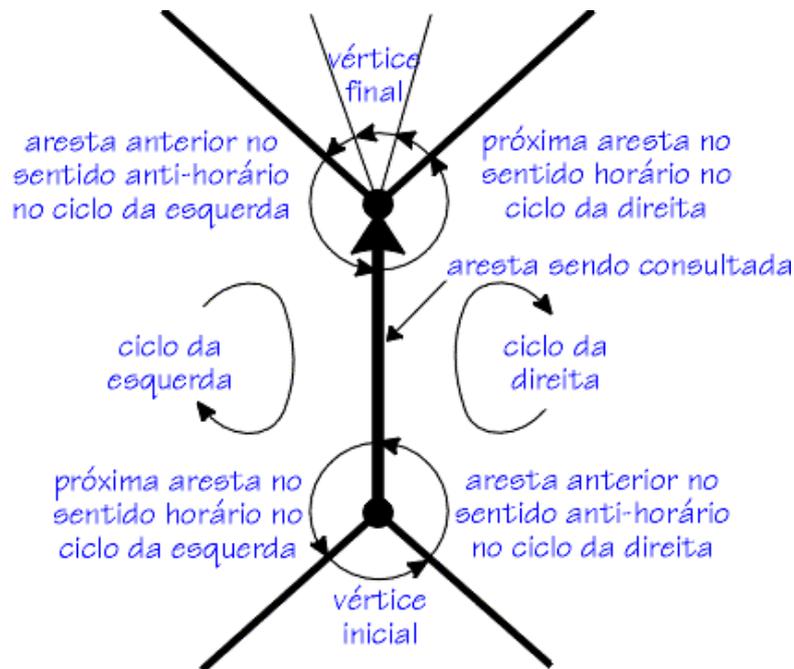


Figura 3.4 - Estrutura de dados *winged-edge*.

Como cada aresta aparece em dois ciclos, duas outras arestas aparecem em seguida a esta aresta em cada um dos ciclos, uma no sentido horário e outra no anti-horário. Mas esta estrutura não utiliza o conceito de uso de uma aresta por dois ciclos que a compartilham. Por sua vez, as estruturas *half-edge* utilizam este conceito.

Observa-se na Figura 3.5 a seguir que a estrutura *half-edge* é hierárquica e formada por listas circulares encadeadas, indo dos níveis mais altos de hierarquia (subdivisão planar) aos mais baixos (vértices). No que diz respeito ao armazenamento explícito das relações

de adjacência, a estrutura *half-edge* não é completa, embora forneça alguns relacionamentos de forma já ordenada. Por exemplo, percorrendo-se o ponteiro *eu_nxt* ou o *eu_prv* (apontam o próximo uso de aresta e o anterior, respectivamente) varrem-se as arestas adjacentes de um ciclo da fronteira de uma face (Figura 3.5a).

Além do encadeamento de ponteiros que estabelece as relações de adjacência, cada entidade possui também um ponteiro para um registro de atributos opcional. A subdivisão planar, a face, o ciclo, a aresta e o vértice podem ter atributos associados. Por exemplo, no caso de representação de seções geológicas, as faces que representam blocos de camadas geológicas podem ter como atributos uma cor, uma litologia, e propriedades físicas como valor de porosidade, decaimento, etc.

A criação da subdivisão planar pode ou não ser um processo interativo. No caso interativo, um processo natural de modelagem é tal que os segmentos de curva são fornecidos um após o outro pelo usuário do sistema, através de um dispositivo como o mouse ou uma mesa digitalizadora.

O pacote HED oferece os recursos necessários para a criação e manutenção de uma subdivisão planar. Basicamente é constituído de uma camada de operadores geométricos que recebe da aplicação as curvas a serem incorporadas à subdivisão. Esta camada trata uma curva, transformando-a em uma ou mais arestas, e passa estas arestas a uma camada de operadores topológicos, chamados de operadores de *Euler*, que fazem a manipulação da estrutura de dados *half-edge* que representa a subdivisão. O HED possui também funções de consulta à estrutura de dados.

Uma outra funcionalidade fundamental que o HED provê é a possibilidade de inserção de um segmento de curva a uma subdivisão. Para tal, deverá haver um pré-processamento geométrico, permitindo que este segmento seja decomposto em segmentos simples que são arestas, contidos inteiramente nas faces da subdivisão já existente.

3.4. O Sistema Recon Original

Nesta seção são descritas as principais funcionalidades e características do sistema *Recon* original, ou seja, antes de se iniciar o desenvolvimento deste trabalho. O sistema apresenta todas as funcionalidades básicas que para se realizar a tarefa de balanceamento de seções geológicas em ambientes distensivos. Dentre estas funcionalidades pode-se destacar:

3.4.1. Criação de Uma Seção Geológica

A criação da seção pode ser feita a partir da importação da mesma através de um outro sistema ou como era mais comumente utilizada, via digitalização de uma seção em papel.

Além disso, o sistema provê um editor gráfico de linhas, às quais são associados atributos próprios (falha, limite de seção horizonte, etc). Aos horizontes ainda tem-se associada uma tabela de idades geológicas, igualmente construída via interface pelo usuário. A edição de linhas está diretamente relacionada às funções do HED de inserção de curvas na subdivisão planar que representa a seção. O uso de uma estrutura de dados topológica e dos recursos de manipulação da subdivisão planar, próprios do HED, discutidos anteriormente. O reconhecimento automático das novas regiões formadas, em tempo real, é o *feedback* ideal para o processo de criação ou importação de uma seção, já que permite a visualização de qualquer inconsistência topológica gerada em um dado instante.

3.4.2. Camadas

A definição das camadas é feita de forma automática a partir do reconhecimento das regiões que vão sendo formadas pela inserção de linhas. Desta forma, o atributo geológico de um bloco é determinado automaticamente pelo sistema com base nos atributos geológicos das linhas na fronteira do bloco. Esta atribuição também pode ser feita de forma explícita pelo geólogo. Os tipos de camada, aqui representados, são a

camada sedimentar, a camada intrusiva, ou a camada indefinida, caso está não tenha sido identificada como nenhuma das duas anteriores.

3.4.3. Módulos

O sistema permite o agrupamento de faces (blocos de rocha) em módulos. A forma simples de definir um módulo é a indicação pelo usuário das faces que devem ser incluídas no novo módulo. Também é possível fazer operações entre módulos, como união e separação. Além de um algoritmo de ajuste implementado para corrigir as inconsistências topológicas aqui traduzidas por *gaps* gerados pelos erros introduzidos durante o processo de balanceamento.

As transformações geométricas, inerentes ao processo de balanceamento, são aplicadas simultaneamente a agregados de faces, arestas e vértices, chamados aqui de módulos. Cada módulo pode conter todas as entidades topológicas que uma subdivisão planar contém, ou seja, cada módulo é uma subdivisão planar à parte. Uma seção geológica será composta então de módulos, sendo que um deles é o módulo base. Na Figura 3.6, tem-se uma seção dividida em três módulos, constituídos por sua vez de blocos. O módulo I poderia ser o módulo base e os módulos II e III são agregados de blocos que se movimentaram com relação ao módulo base.

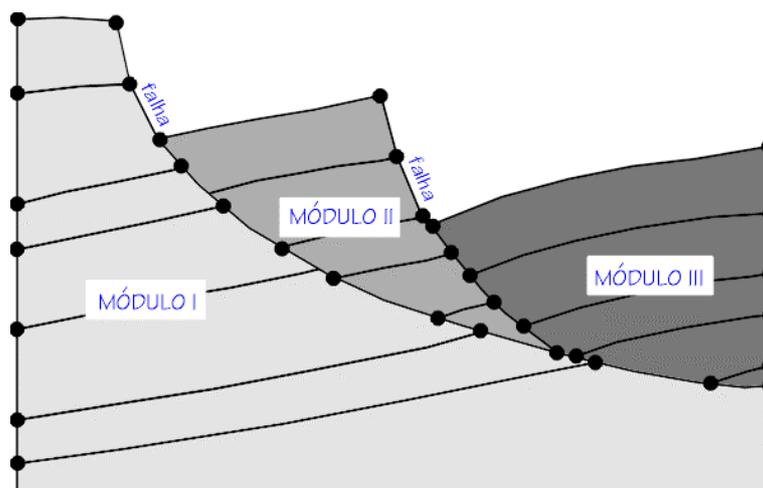


Figura 3.6 -Subdivisão hierárquica da seção dividida em módulos.

3.4.4. Transformações

O Sistema possui o conjunto básico de transformações geométricas. No caso, as transformações já citadas no Capítulo 2. Na Seção 3.6 são documentadas as técnicas envolvidas na implementação do conjunto de transformações de que dispõe o sistema.

3.4.5. Consultas

O *Recon* apresenta um conjunto de operações de consulta, tanto aos atributos das entidades da seção (tipos e idades de linhas, parâmetros de uma camada, etc.), quanto consultas que dizem respeito à geometria da seção (medidas de comprimento de linhas, valor de ângulos, totalização de área, etc)

3.4.6. Ajuste

A operação de ajuste, usada na colagem de módulos, é um procedimento interativo onde o usuário visualmente define a tolerância a ser usada no ajuste. Assim, o geólogo pode fazer tentativas com várias tolerâncias, de forma a obter um ajuste que considere mais aceitável. O objetivo desta operação é corrigir o erro introduzido durante o processo de balanceamento já que não pode haver sobreposição ou vazios, mantendo assim a consistência topológica do modelo.

3.5. Atributos Geológicos na Versão Original do Recon

Esta seção descreve a organização de classes de atributos da versão original do Recon. Neste trabalho, conforme é descrito no Capítulo 5, foi realizada uma reestruturação completa desta organização com o objetivo de permitir a incorporação de um módulo de análise ao processo de balanceamento.

O HED [40] permite que sejam armazenados os atributos referentes às faces e arestas do

modelo. Geologicamente, cada seção pode ser representada contendo atributos para as camadas e para as suas linhas, que podem ser interfaces entre as camadas ou simplesmente linhas de falha, limite de seção, etc. No *Recon*, cada entidade geométrica da seção está associada aos seus atributos geológicos, mas, na modelagem do sistema, camadas na realidade representam um conjunto de faces com os mesmos atributos. As entidades topológicas HED que apresentam classes de atributos atrelados a si, são:

- **Faces:** que herdam os atributos das camadas a que pertencem. Esses atributos são: porosidade inicial, taxa de decaimento (expressão 2.6) e código de litologia da rocha que compõe a camada. Além disso tem associadas a si os seus respectivos tipos: *depósito, topo de sal, indefinida ou externa*.
- **Arestas:** são formadas por curvas, representadas por poligonais, denominadas aqui de linhas. Estas linhas na seção geológica têm o atributo tipo, que pode ser: *horizonte, falha, limite de seção, discordância, topo do sal* ou até mesmo indefinida, caso esta não tenha ainda recebido qualquer destes tipos. Ao horizonte ainda fica associado um outro atributo que é a sua idade geológica.

O HED provê ainda mecanismos flexíveis que possibilitam o manuseio destes atributos. O programador do aplicativo que usa o pacote HED passa para ele ponteiros de funções do aplicativo. Automaticamente estas funções serão disparadas quando ocorrer uma operação topológica sobre a subdivisão. Por exemplo, se uma face é dividida, uma função escrita pelo programador cliente da biblioteca é chamada e, por exemplo, os atributos das duas faces podem ser herdados da face original.

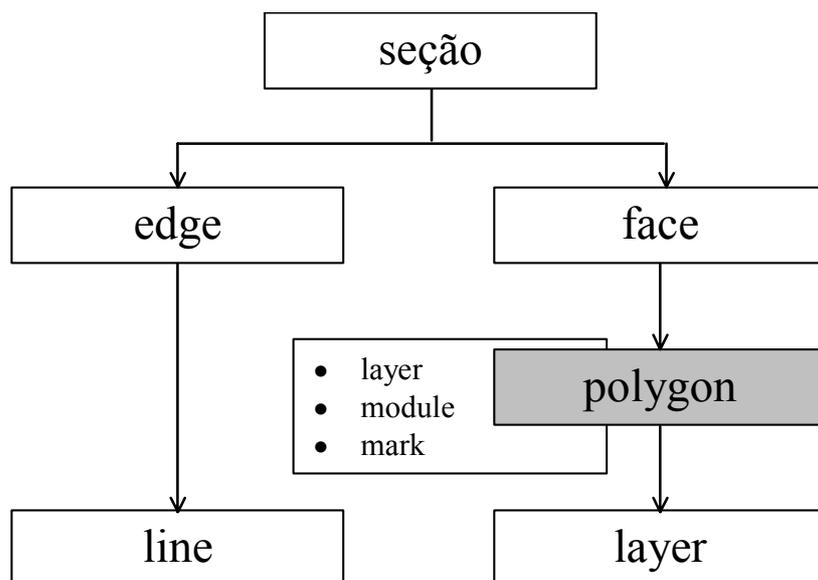


Figura 3.7 – Organização original das classes de atributos no Sistema Recon.

A estrutura original de hierarquia das classes no *Recon* responsável pelo controle dos atributos geológicos associados as entidades topológicas da subdivisão é ilustrada de forma simplificada conforme a Figura 3.7.

A idéia básica envolvida estava nas abstrações criadas entre a geometria (subdivisão planar) e as classes e sub-classes dos atributos geológicos. Associados a uma seção estão os elementos topológicos de aresta e face, que são basicamente tipos do HED. Na concepção original, a super-classe *Line* que representava o conjunto de todos os tipos geológicos de linhas (horizontes, falhas, etc.) e a aresta topológica não apresentavam entre si uma outra camada de abstração. Por outro lado, entre as faces topológicas do HED e a super-classe *Layer*, é possível observar em destaque a classe *Polygon* que representava justamente a interface entre atributo e geometria de forma explícita.

Cabia à classe *Polygon* gerenciar esta interface e ainda ter pendurados a si própria outros atributos não identificados a dados geológicos e ou mesmo não relacionados à subdivisão planar, tais como apontadores para o módulo de trabalho (*module*), ao qual uma instância de *Polygon* estaria associado, e uma lista de marcadores de transformação (*mark*).

3.6 As Transformações Geométricas

Os blocos de rocha de uma estrutura geológica sofrem movimentações e deformações de origem tectônica ao longo do tempo. As transformações geométricas realizadas no balanceamento também correspondem a movimentações e deformações destes blocos, em geral na tentativa de reverter os mecanismos geológicos ou de simulá-los.

A aplicação das transformações geométricas requer, primeiramente, que sejam definidos módulos de trabalho, sobre os quais serão aplicadas as transformações. O usuário, baseado na sua experiência em geologia, escolherá os blocos da seção que serão agrupados, formando o módulo.

Basicamente o critério será o de selecionar blocos que sofreram o mesmo tipo de movimento ou deformação em conjunto, para que possa ser aplicada a estes blocos a transformação inversa visando a reconstituição da seção. Um processo interativo natural de definição de um módulo é o apontamento, pelo usuário, das faces que representam os blocos a serem agrupados.

As transformações relevantes para o processo de balanceamento podem ser divididas entre as que movimentam corpos rígidos e as que causam deformação. As da primeira classe são a translação e a rotação, que são feitas por algoritmos tradicionais. Já as transformações não rígidas têm algoritmos relativamente complexos e são baseadas nos princípios geológicos discutidos no Capítulo 2.

Todas as operações de transformação, com exceção da descompactação, constam de três fases: a coleta das informações necessárias, preparação destas informações de forma que fiquem adequadas a serem aplicadas a cada uma das coordenadas que definem a geometria do módulo e a aplicação da transformação propriamente dita ponto a ponto.

3.6.1. Transformações Rígidas

3.6.1.1. Translação

As transformações de translação e de rotação de corpos rígidos são apropriadas para uso em regiões onde o movimento tectônico gera o deslizamento e basculamento de blocos, associados a falhamentos planares.

Falhas com perfil plano causam a translação (deslizamento) do bloco baixo, sem a formação da chamada geometria de *rollover*, ou seja, não ocorrem mudanças na forma dos blocos, apenas movimentação. A translação de um módulo promove o deslocamento do mesmo em uma determinada direção e distância.

3.6.1.2. Rotação

O deslocamento ao longo de uma falha, com trajetória circular, promove a rotação dos blocos sobre a mesma, associada a um movimento de translação. Durante o deslizamento e a rotação ao longo da falha, as paredes dos blocos alto e baixo permanecem em contato até que seja atingida a parte mais profunda da falha, onde ela tende a ser horizontal (*profundidade de detachment*). Aí pode surgir um espaço entre os dois blocos. No caso dos blocos serem formados por materiais pouco dúcteis (rochas indeformáveis), não haverá fluxo plástico e estes espaços não serão preenchidos devido à deformação e à adaptação do perfil destes blocos. Estes vazios abaixo destes blocos causam uma instabilidade, possibilitando a rotação dos mesmos.

3.6.2. Transformações Não Rígidas

3.6.2.1. Cisalhamento Simples

As falhas geradas pela movimentação tectônica são acompanhadas do abatimento de blocos, devido a falta de estabilidade dos mesmos, que faz com que eles percam a

sustentação com a separação dos blocos pela falha. Esse abatimento ocorre segundo uma direção (ângulo). Em geral esta direção está próxima à vertical e coincide com a orientação da força de gravidade. O bloco se acomoda sobre o leito da falha, ao se deformar plasticamente, ajustando-se ao perfil da falha, ou seja, o bloco cisalha segundo uma direção. As partes do bloco cisalhado se deslocam na direção do abatimento, sendo o deslocamento diferenciado ao longo do corpo do bloco em função do perfil da falha. Na realidade o cisalhamento pode ser aplicado a um módulo, ou seja, a um grupo de blocos. A este módulo é aplicado um campo de deslocamentos, formado por linhas paralelas. Assim, todas as linhas deste campo tem uma direção única, mas cada linha está associada a um deslocamento que depende da distância (Figura 3.8a) ao longo da linha (unidimensional) entre um perfil destino (leito da falha) e um perfil origem (fronteira do módulo que se abate sobre o leito). O mesmo mecanismo é usado na reconstituição da geometria original, ou seja, no processo inverso ao do abatimento do módulo. Cada vértice do módulo é submetido ao campo de deslocamentos.

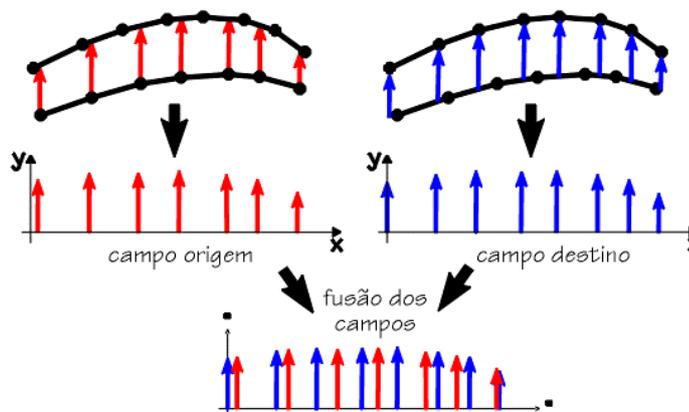
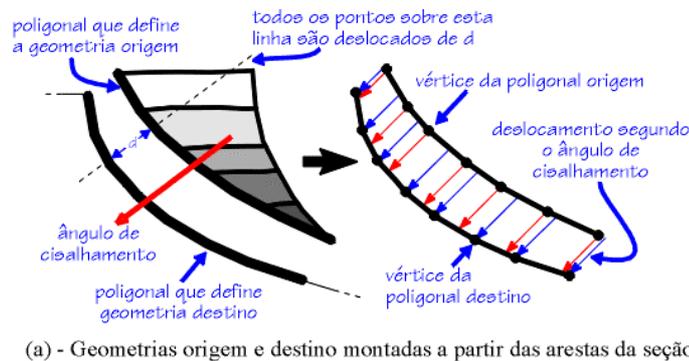


Figura 3.8 – Cisalhamento simples: montagem do campo de deformações.

As rotinas que coletam as informações necessárias para o cisalhamento fornecem ao algoritmo duas poligonais (geometria origem e destino) e o ângulo de cisalhamento.

A fase do algoritmo que prepara o campo de deslocamentos é a parte mais complexa conforme pode-se observar na descrição abaixo:

- **Passo 1** - *É criado um novo sistema de coordenadas auxiliar para o campo de deslocamentos a ser montado. Para obter suas coordenadas no novo sistema, as poligonais origem e destino são transladadas e rotacionadas. Com a rotação, os deslocamentos do campo só terão componentes no eixo y e as poligonais se desenvolverão na direção do eixo x. Com a translação, o primeiro ponto das poligonais coincidirá com o início do eixo x. Esta modificação do sistema de coordenadas facilita as fases seguintes do algoritmo. Nesta fase, é feito um teste para detectar dobras nas linhas origem e destino, ou seja, se existe mais de uma ordenada para uma mesma abscissa (Figura 3.9). Em caso afirmativo, a operação é interrompida. Um outro teste é feito para garantir que os limites, na direção x, da poligonal destino fiquem iguais ou maiores que o da poligonal origem, pois esta se projetará sobre a poligonal deformada. Se necessário, a poligonal destino é prolongada retilinearmente.*

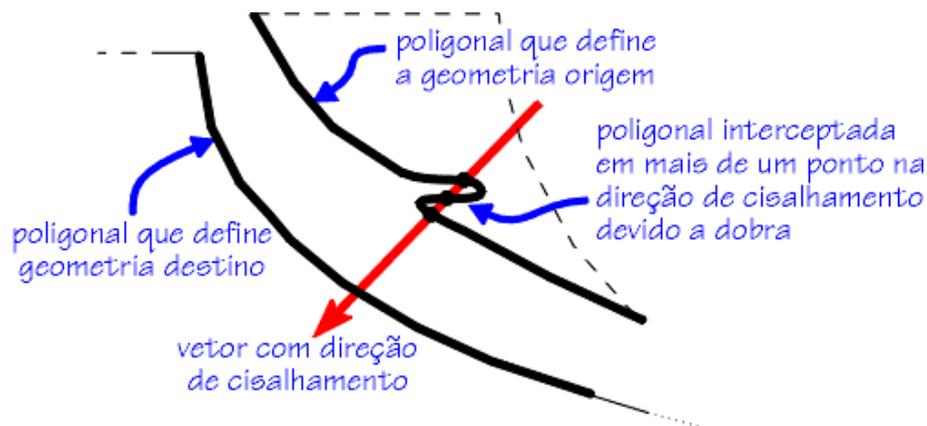


Figura 3.9 – Cisalhamento simples: dobras na geometria origem.

- **Passo 2** - *A partir das poligonais origem e destino, já no novo sistema de coordenadas, são criados os campos de deformação origem e destino, respectivamente (Figura 3.8.b). O primeiro campo de deformações (o origem) parte de cada ponto da poligonal origem, até a interseção da vertical que passa por este ponto com a poligonal destino. A deformação ou deslocamento para a dada abscissa será a diferença entre as ordenadas do ponto de interseção e do ponto da linha origem. O segundo campo chega a cada ponto da linha destino, partindo da interseção da vertical que passa por este ponto com a poligonal origem. Os valores do campo também serão a diferença entre as ordenadas dos pontos e os respectivos pontos de interseção.*
- **Passo 3** - *Após a criação dos dois campos, eles são fundidos em um só campo de deformações, a ser usado posteriormente no cisalhamento. Esta fusão se baseia nas abscissas que devem ter sido previamente colocadas em ordem crescente. No caso de pontos com mesma abscissa é evitada a repetição da mesma no campo resultante.*

A aplicação do algoritmo ao campo de deslocamentos simplesmente repetirá para todos os vértices do módulo em transformação os seguintes passos:

- Transformação para o sistema de coordenadas auxiliar, que é o sistema do campo de deformações.
- Aplicação do campo de deformações, ou seja, é feita uma interpolação linear da abscissa, e é obtida uma ordenada correspondente à deformação naquela faixa do campo.
- Transformação para o sistema de coordenadas original da seção.

3.6.2.2. Move Sobre Falha

A transformação de movimentação de um bloco sobre uma falha lítrica é na realidade a combinação de duas das transformações acima descritas: translação e cisalhamento puro. Inicialmente são requeridos:

- a geometria da falha sobre a qual haverá a movimentação
- ponto p_0 de referência sobre o módulo a ser movido
- ponto p_1 sobre a linha destino (falha)
- ângulo de cisalhamento

Com os pontos p_0 e p_1 define-se o vetor que rege a parcela de translação aplicada ao módulo. O cisalhamento simples, por sua vez, para ser executado necessita como dados de entrada a geometria destino, a linha de falha selecionada, a geometria origem, que é representada pelo conjunto dos segmentos da falha selecionados que interfaceia o módulo, e o ângulo de cisalhamento, que já é dado de entrada. A Figura 3.10 ilustra o move-sobre-falha realizado dentro do ambiente *Recon*.

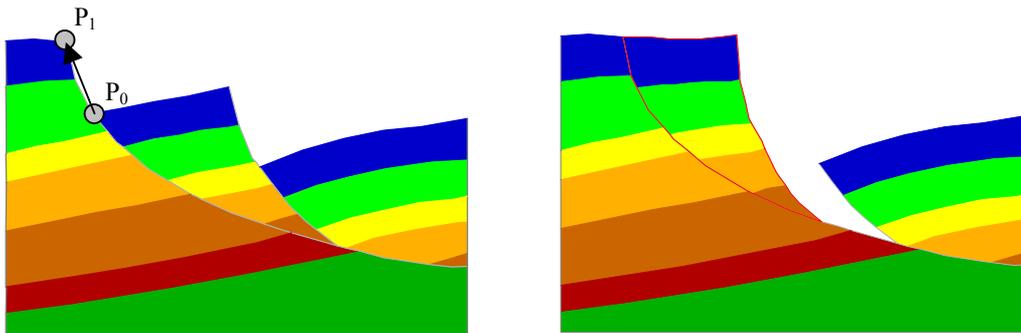


Figura 3.10 – Exemplo de restauração com o Move sobre falha geométrico.

3.7. Novos Desenvolvimentos

No início do desenvolvimento deste trabalho, foi possível interagir com vários geólogos e geofísicos usuários do sistema o que contribuiu bastante para que novas funcionalidades fossem incorporadas ao sistema, buscando torná-lo mais amigável sob o ponto de vista da interface com o usuário dentro da filosofia do balanceamento de seções geológicas.

A estratégia adotada, portanto, ao início do trabalho foi implementar uma nova versão para o sistema, mais atualizada e funcional sob o enfoque da interface gráfica.

Inicialmente as únicas alterações realizadas foram as migrações dos sistemas de interface e gráfico conforme pode-se observar na seção 3.7.1 a seguir.

A presente seção visa apresentar as principais funcionalidades incorporadas ao *Recon*, descrevendo as ferramentas utilizadas e os ganhos obtidos tanto sob o ponto de vista do uso do programa quanto efetivamente as novas possibilidades de desenvolvimento que surgiram em função destas implementações.

3.7.1. Nova Interface Gráfica

No início do seu desenvolvimento, início da década de 90, os recursos computacionais eram mais escassos principalmente no que diz respeito às bibliotecas de interfaces gráficas disponíveis na época. O mesmo pode-se dizer com relação aos sistemas gráficos.

O sistema de interface do sistema foi inteiramente reescrito utilizando-se da biblioteca *IUP* [39], desenvolvido no Tecgraf/PUC-Rio. O IUP é um sistema portátil de criação de interfaces que oferece uma *API* (*Application Program Interface*) de configuração através de três linguagens básicas: *C* [38], *Lua* [34] e *LED*. Sua biblioteca contém várias funções para a criação e a manipulação de diálogos.

A idéia básica do IUP é permitir que um programa possa ser executado sem modificações em diferentes sistemas, sendo, portanto, a alta portabilidade uma de suas principais características. Os sistemas suportados são: *Motif*, *Microsoft Windows* (contém bibliotecas para todas as versões). Por isso, o IUP possui algumas vantagens em relação aos *toolkits* de interface comerciais disponíveis no mercado.

Foi substituído também o sistema gráfico do *Recon* para o, *CD* (*Canvas Draw*), uma biblioteca gráfica independente implementada em várias plataformas usando as suas respectivas bibliotecas gráficas nativas. O *CD* [13] contém funções que instanciam objetos gráficos vetoriais e *raster*. Além disso, esses objetos podem ser visualizados tanto em uma área de desenho, *canvas*, quanto em superfícies mais abstratas, como *Clipboard*,

Metafile e *Postscript*, o que possibilita a exportação de modelos do Recon nesses formatos.

Com a migração do *Recon* para os sistemas IUP/CD, foi possível gerar versões do *Recon* para outras plataformas, entre elas o *Windows*, o que não era possível na versão antiga do sistema. Além disso, a interface tornou-se muito mais amigável sob o ponto de vista do usuário, com um *layout* bem mais moderno.

3.7.2. Geração e Importação de Novas Seções

Com a substituição do sistema de interface do sistema não foi mais possível gerar novas seções via mesa digitalizadora, já que o *IUP* não possui *driver* para este periférico. Com isso, foi necessário lançar mão de outros mecanismos para a geração de novas seções.

Somadas as bibliotecas acima descritas utilizou-se uma outra biblioteca para se fazer a leitura de imagens raster, o IM [35], igualmente desenvolvida pelo *Tecgraf*. O IM é capaz de ler e escrever imagens *bitmaps* em formatos *raster*, tais como BMP, PCX, GIF, TIF, etc. A IM possui total compatibilidade com imagens cliente do CD, fornecendo armazenamento persistente para imagens cliente.

O que se ganhou foi a possibilidade de importação de uma imagem raster, e que permitiu sofisticar a entrada dos dados até aqui feitos ou por leitura de uma seção no formato padrão do *Recon*. A imagem é importada e colada ao fundo da tela do sistema. Neste processo o *Recon* passa a ser também um ambiente de interpretação de dados sísmicos armazenados em forma de imagem digital, já que através do editor de linhas do sistema é possível usar a imagem colada como fundo de tela e aí sim gerar uma nova seção a ser balanceada com base na imagem importada.

Foi implementado também, dentro do sistema, um conversor de formato de dados geológicos em uso na *PETROBRAS*, formato *BIG* [62], para o formato de dados do sistema *Recon*. Mais do que isso, o formato *Recon* foi cadastrado como um objeto da

base de dados do Sistema Integrado de Geologia e Geofísica da *PETROBRAS*, o *SIGEO* [63], de onde o *Recon* pode ser executado de forma totalmente integrada.

Além disso, constatou-se que uma característica básica do processo do balanceamento, que está relacionada ao minucioso processo de tentativa e erro, não foi considerada na implementação original do sistema. Sendo assim, a tarefa de balancear seções geológicas no *Recon* era de fato bastante árdua uma vez que ao usuário era atribuída a incumbência de gerenciar os modelos gerados salvando e carregando inúmeros arquivos no intuito de obter no final um mapa completo com todas as etapas de balanceamento. Isso tudo era feito sem que se disponibilizasse nenhum recurso que introduzisse quaisquer facilidades.

3.8. O Gerenciador de Balanceamento

O balanceamento de seções geológicas é por definição um processo de repetição, que envolve várias tomadas de decisões e, por conseguinte, caminhos distintos em busca da configuração original do solo. É, portanto, uma operação de tentativa e erro. Para cada geometria obtida em uma determinada etapa de balanceamento tem-se uma nova decisão a ser tomada. Sendo assim, é natural em um dado instante o geólogo deparar-se com a dúvida de ter que optar entre uma operação ou outra. Qual a mais conveniente só será possível saber alguns passos adiante.

Devido a esta peculiaridade, inerente ao processo, foi implementado no *Recon* um módulo responsável pelo gerenciamento da restauração geológica da seção. Até aqui, para documentar as etapas envolvidas no balanceamento era necessário salvar cada arquivo com cada geometria obtida durante as transformações as quais era submetida uma determinada seção. Esse procedimento, além de lento e trabalhoso, não permitia ao geólogo uma visão global de seu trabalho. Foi justamente com base nesta limitação que surgiu a motivação para o desenvolvimento de um gerenciador gráfico e interativo para a manipulação, controle e visualização das etapas do balanceamento.

O gerenciador de balanceamento utiliza-se de uma estrutura de dados que armazena os nomes das seções e os acessa via leitura do arquivo de dados no formato que o *Recon* já utilizava anteriormente. Devido à natureza hierárquica do problema, a estrutura de dados utilizada é uma árvore.

Árvores são abstrações computacionais para representação de dados. São estruturas essencialmente dinâmicas, permitindo inserção e destruição de elementos. As operações de inserção, destruição e busca são realizadas eficientemente sob o ponto de vista computacional. Árvores trazem, implicitamente, a idéia de hierarquia, por isso são estruturas muito utilizadas em gerenciamento de arquivos [20].

Uma árvore pode ser definida de várias formas. Talvez, a maneira mais natural seja defini-la recursivamente. Existe um nó r , denominado *raiz*, que contém zero ou mais sub-árvores, cujas raízes são ligadas diretamente a r . Estes nós raízes das sub-árvores são ditos filhos do nó r . Dentro do modelo de balanceamento, o nó raiz significa a seção em sua geometria atual, ou seja, a seção a ser balanceada. A Figura 3.11 exemplifica a estrutura de uma árvore genérica. [14]

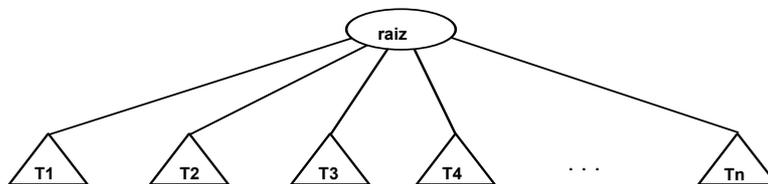


Figura 3.11 – Estrutura de uma árvore genérica

Em geral usa-se classificar uma árvore quanto ao número de filhos. Em uma árvore genérica, por exemplo, o número de filhos de um determinado nó é qualquer. Uma *quadtree* apresenta quatro filhos por nó, uma árvore binária apresenta dois filhos. Dentro da questão do balanceamento o mais conveniente é não amarrar o número de filhos de um determinado nó já que não se pode saber *a priori* quantas alternativas de balanceamento uma seção vai apresentar em uma dada etapa. O desenho esquemático da Figura 3.12 exemplifica como foi feita a implementação da estrutura de dados para o gerenciador de

balanceamento. Pode-se observar que para cada nó da árvore apresenta-se uma lista de nós filhos.

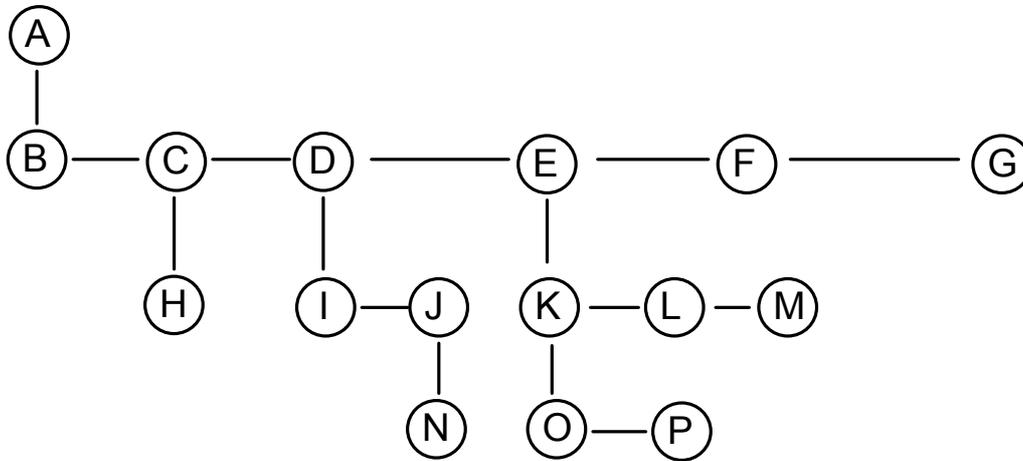


Figura 3.12 – Relação hierárquica entre nós da árvore.

Cada nó apresenta portanto um ponteiro para o primeiro filho e um outro para o próximo irmão. Além disso, outras informações adicionais são guardadas como ponteiros para o pai, para o último filho e para o irmão anterior. Essas informações adicionais servem para aumentar a eficiência no uso de operações como inserção e busca. Observando o desenho esquemático acima vê-se que, por exemplo, para o nó **E**, teríamos **A** como seu pai, **K** como primeiro filho, **M** como último filho, **F** como próximo irmão e **D** como o irmão anterior. Por sua vez, o nó **P** teria **K** como pai e **O** como irmão anterior.

A implementação foi feita utilizando-se o paradigma da programação orientada a objetos e seu código foi escrito na linguagem C++ [56]. Foram criadas duas classes: uma que implementa a abstração de um nó da árvore, e uma outra que por sua vez implementa a abstração do objeto árvore propriamente dito. A segunda instancia objetos da primeira e gerencia as relações hierárquicas entre os nós. É possível observar abaixo trechos do código da classe que representa um nó da árvore, Figura 3.13, principalmente no que se refere as relações entre os nós especificamente.

```

class VisNode
{
    protected:
        VisNode* _father;
        VisNode* _firstChild;
        VisNode* _lastChild;
        VisNode* _nextBrother;
        VisNode* _prevBrother;
        ...
    public:
        VisNode();
        virtual ~VisNode();
        ...
        VisNode* Father (void) { return _father; }
        VisNode* FirstChild (void) { return _firstChild; }
        VisNode* LastChild (void) { return _lastChild; }
        VisNode* NextBrother(void) { return _nextBrother;}
        VisNode* PrevBrother(void) { return _prevBrother;}
        ...
        virtual void Select(void) {};
        virtual void Execute(void) {};
        ...
        virtual void MSelect (List<VisNode> *) {};
        virtual void MExecute(List<VisNode> *) {};
        // ...
};

```

Classe VisNode

Figura 3.13 – Implementação da classe VisNode.

Para representar graficamente a estrutura de dados hierárquica acima descrita, foi implementada a classe *VisNode*, que representa um nó, e a classe *VisTree*, que representa o novo objeto de interface para navegação no projeto de balanceamento, Figura 3.14, que é responsável pelo *layout* e interface com o usuário. A classe *VisTree* portanto é quem

possibilita a visualização do histórico do balanceamento, desenhando todas as tomadas de decisão e os caminhos que definiram as eventuais configurações finais obtidas.

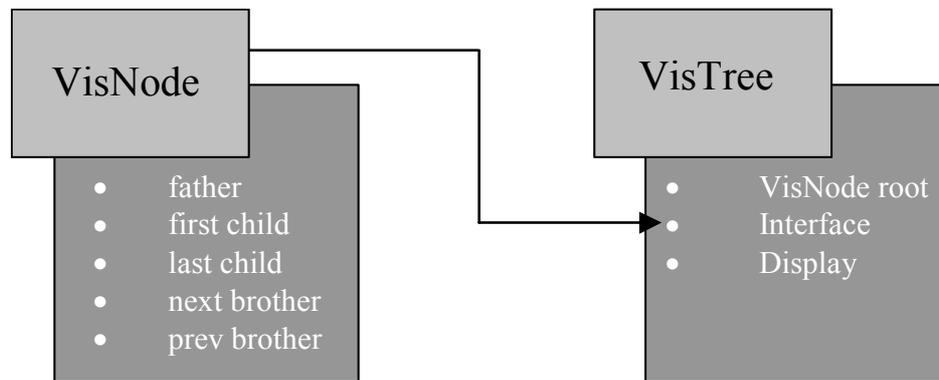


Figura 3.14 – Relacionamento entre as classes *VisNode* e *VisTree*.

Para incorporar a árvore de decisões de balanceamento ao *Recon* foi necessário introduzir ao sistema o conceito de projeto. Dentro do sistema, um projeto é caracterizado pela existência da árvore de balanceamento. Foi incorporado um novo formato de arquivo, *ASCII*, que faz uma representação recursiva da árvore. Este arquivo ao ser carregado para o sistema, abre o seu projeto e permite a sua visualização através da árvore implementada. Cada nó deste arquivo, tem um nome associado, que referencia o arquivo correspondente ao nó onde são armazenados os dados de cada etapa do balanceamento. O formato do arquivo que contém as seções é o mesmo que já era utilizado pelo antigo sistema, ainda sem a árvore. Um módulo de projetos foi então criado para gerenciar a abertura dos projetos já existentes e a criação dos novos, e a atualização da árvore de balanceamento. O resultado final pode ser observado na Figura 3.15.

Para o projeto abaixo, Figura 3.15, pode-se observar que *sp* é o nome do arquivo gerenciador que contém as informações referentes à topologia da árvore. O nó *sp1* representa a seção a ser balanceada. O nó que apresenta a cor mais clara, *sp9*, representa a seção selecionada e a linha poligonal destacada em amarelo determina o caminho de balanceamento desde a seção original *sp1* até a seção selecionada *sp9*. Por apresentar uma estrutura de dados dinâmica, é possível inserir e eliminar nós na árvore. Algumas

sofisticações também foram implementadas, tais como inserção de textos referentes as seções e para um determinado caminho de balanceamento é possível executar uma animação com suas etapas.

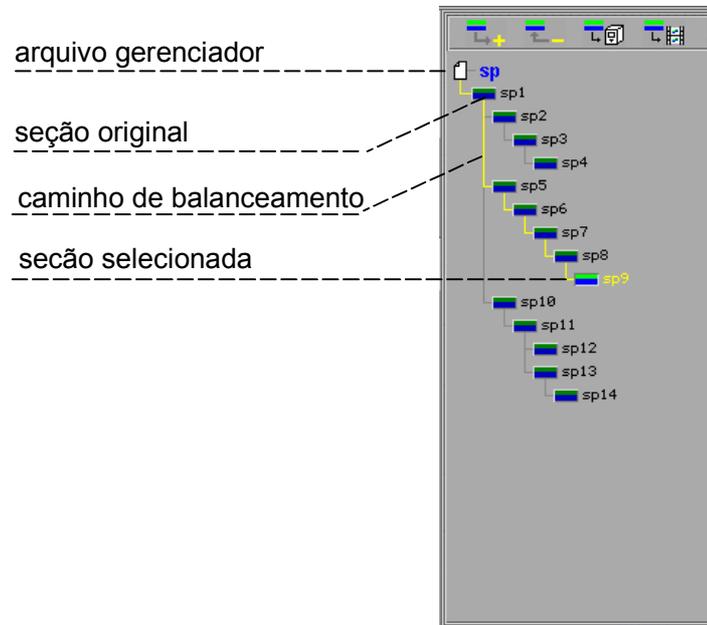


Figura 3.15 – Interface do gerenciador de balanceamento.

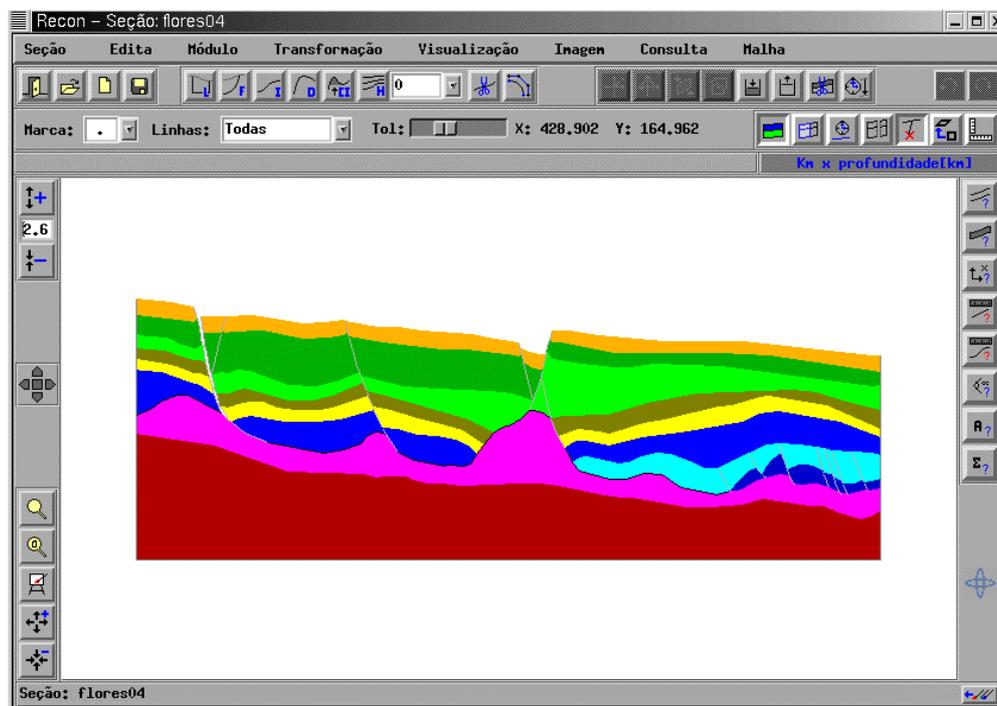


Figura 3.16 – Interface do Sistema Recon sem o gerenciador de balanceamento.

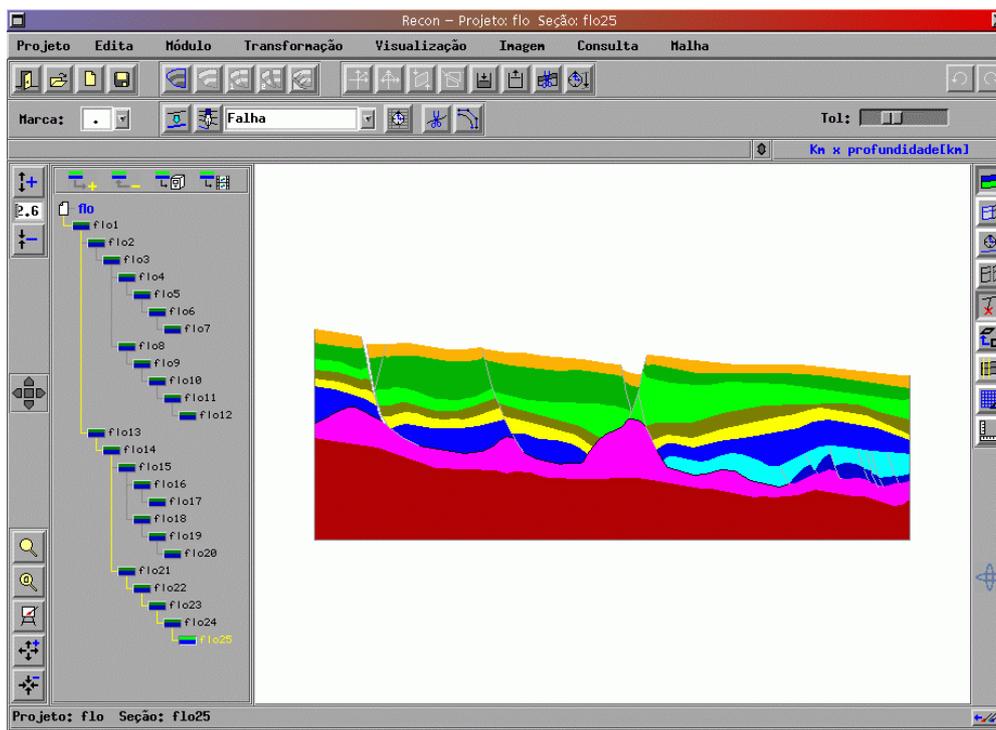


Figura 3.17 - Interface do Sistema Recon com o gerenciador de balanceamento.

3.9. Implementação da Modelagem do Sal

O balanceamento de seções geológicas que apresentam camada de sal requer uma disciplina um pouco diferente do que a usualmente adotada. Isso se deve ao comportamento do sal discutido na Seção 2.2.3.

Na realidade a aproximação que normalmente se faz é reconstituir a seção acima da camada de sal e a medida que formam-se os vazios em cada etapa, preenche-se os mesmos com o sal considerando-se que o comportamento de escoamento do sal é passivo no processo de distensão da crosta.

A seqüência das Figuras 3.18, 3.19 e 3.20 ilustra este procedimento. Na Figura 3.18 tem-se uma determinada seção em uma dada etapa de balanceamento. Na seqüência, a Figura 3.19 apresenta a mesma seção submetida a algumas transformações realizadas nos blocos posicionados acima da camada do sal a esquerda do seu maior domo, onde é possível

observar que essas transformações formaram um polígono vazio (face HED sem atributo algum associado). A seguir, na Figura 3.20, é possível visualizar a camada de sal preenchendo esses vazios

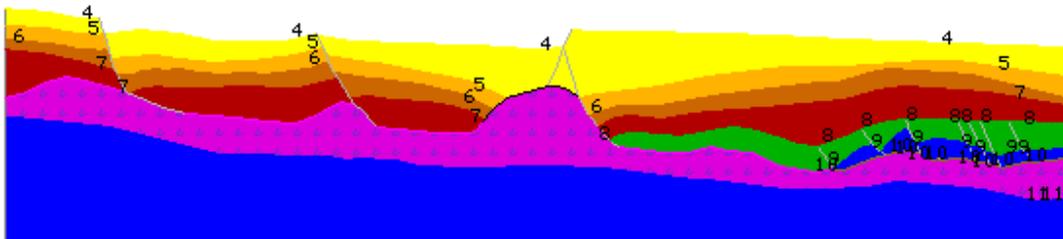


Figura 3.18 – Seção com camada de sal: etapa 1.

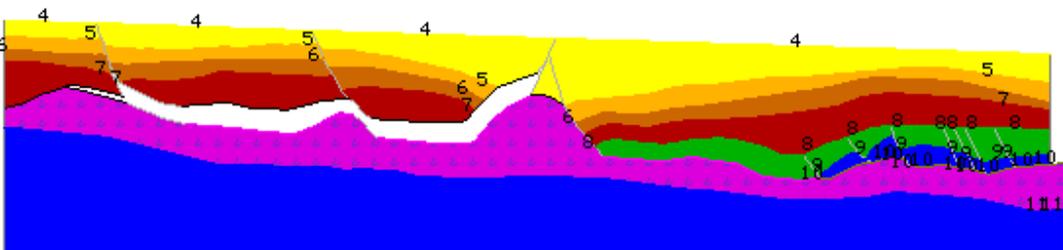


Figura 3.19 – Seção com camada de sal: etapa 2.

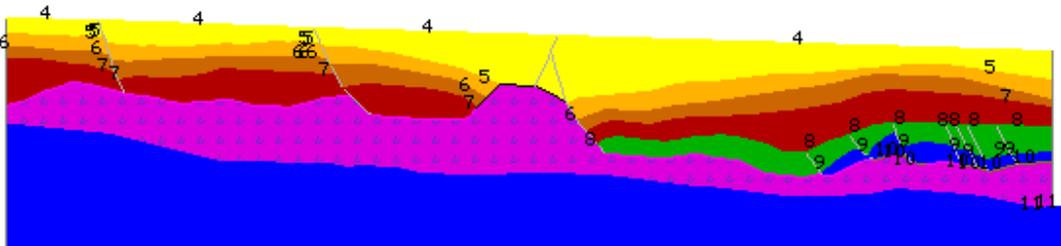


Figura 3.20 – Seção com camada de sal: etapa 3.

O que em geral o geólogo deseja controlar, portanto, durante as etapas de balanceamento é justamente o fluxo do sal. Isso se deve ao fato do escoamento do sal ser tridimensional e por conseguinte, mesmo a seção sendo definida perpendicularmente ao transporte tectônico (garantindo que as camadas não apresentem perdas durante o balanceamento), ainda assim o sal presente na seção poderá apresentar um acréscimo ou perda de material.

Com o advento do gerenciador de balanceamento, esse controle pôde ser incorporado ao sistema. A idéia consiste em, cada etapa do balanceamento, representado por cada nó da

árvore de visualização acima descrita, armazenar a área de sal presente na seção nesta etapa. Esse dado fica pendurado como atributo do nó da árvore. Além disso, consulta-se, para um determinado nó selecionado (seção corrente) a área correspondente ao seu nó pai, o que possibilita informar a taxa de variação da área da camada de sal entre etapas adjacentes do processo de balanceamento.

Assim, ao final do processo, disponibiliza-se todo o histórico da área do sal e o percentual de variação entre as etapas de balanceamento ao longo do processo. Paralelo a isso, duas outras ferramentas foram implementadas para facilitar esse procedimento. A Figura 3.21 mostra o elemento de interface presente no diálogo principal do programa que informa o histórico de variação do sal ao longo do balanceamento da seção. Cada número da seção representa um nó da árvore ou uma etapa de balanceamento.

Sal		
Secao	Area	Taxa
16	2.5e+07	0,00
15	2.5e+07	1,83
14	2.4e+07	8,79
13	2.2e+07	1,02
12	2.2e+07	0,01
direção do fluxo		← →

Figura 3.21 – Visualização do histórico de variação da área de sal.

Para preencher os vazios presentes na seção da Figura 3.19 é necessário, antes de tudo, eliminar a camada antiga, ou seja, existente antes de se retrodeformar os blocos de rocha. A primeira ferramenta refere-se a eliminação automática da camada de sal. Com o HED essa tarefa torna-se facilitada, bastando identificar o polígono de sal, no caso uma face HED, percorrer a sua lista de arestas e eliminar aquelas que tenham adjacência com a face externa.

A segunda trata-se de um algoritmo de cálculo automático da área do polígono de sal. Uma vez eliminada a camada de sal o que se deseja é justamente acomodá-la nos novos

vazios gerados. Como o fluxo de sal na seção se dá no seu sentido longitudinal o novo polígono é preenchido automaticamente bastando indicar a taxa de variação do sal entre uma seção e outra e a direção do fluxo (para a esquerda ou para a direita da seção).