Capítulo 7

Implementação do Dual-T-Snakes

Na implementação do Dual-T-Snakes alguns detalhes merecem especial atenção. Além disso, alguns aspectos das T-Snakes foram revistos de acordo com as características específicas do novo método.

Neste Capítulo procuramos analisar com certo detalhe estes aspectos, particularmente no que diz respeito aos parâmetros e às forças envolvidas.

O modelo de Balloon (Cohen, 1991), desenvolvido na seção 3.4, será sempre uma diretriz para a apresentação que segue. Isto se deve principalmente pela utilização de uma força normal extra do tipo daquela usada no Balloon.

No final desta discussão apresentamos um fluxograma que fornece um *over-view* do algoritmo.

7.1 Escolha de Parâmetros

O primeiro ponto a ser analisado se refere à escolha dos parâmetros. Uma vez que o modelo básico é dado pelas T-Snakes, nós temos a tensão elástica (a_i) , o comprimento natural (l_i), rigidez (b_i), o fator de escala k para a força normal e o coeficiente λ da força de imagem. Este ponto é central no Dual-T-Snakes porque precisamos escolher parâmetros para duas snakes: a interna e a externa. A Figura 7.1 ilustra bem as dificuldades inerentes a este fato.

Nesta figura representamos um pequeno trecho das snakes. Os vetores correspondentes à tensão elástica (α_i) e à rigidez (β_i) estão desenhados de acordo com a análise da seção 3.4.3. Estamos supondo que o comprimento natural l_i seja nulo e que a região tem pouco contraste, portanto, com força externa despresível.

Assim, as snakes só poderão caminhar em direção à borda caso as resultantes intínsecas sejam orientadas no sentido da mesma.

Contudo, observemos que as forças internas possuem resultante apontada para a borda no caso da snake externa, mas apontando no sentido contrário para a snake interna.

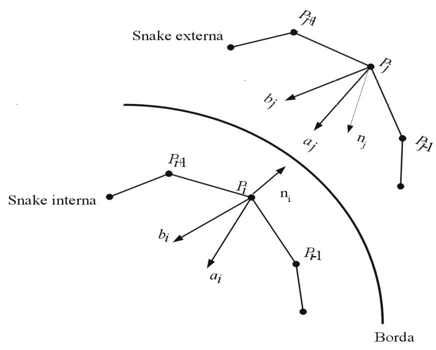

Figura 7.1: Forças internas para as Snakes do Dual-T-Snakes.

A dificuldade maior decorrente deste comportamento está na escolha do fator de escala para a força normal. Este fator precisa ser alto suficiente para que a T-Snake interna se movimente no sentido desejado, mas não pode aumentar demasiadamente a resultante das forças internas da outra snake, pois, se isto ocorrer, poderemos ter dificuldades para estabilizar esta snake próximo da borda desejada.

A forma mais direta para resolver esta dificuldade seria usar parâmetros distintos para as snakes internas e externas. Contudo, isto implica em calibrar duas snakes, o que é indesejável, visto que não temos um procedimento sistemático para a escolha de parâmetros em modelos deformáveis em geral (R. Fisker, 1998).

Uma outra proposta seria tomarmos como negativos os parâmetros a_i, b_i. Com isto, mudaríamos o sentido das forças internas (Figura 7.2). No entanto, esta escolha não tem se mostrado eficiente na prática e o papel de suavização das forças internas neste caso torna-se pouco claro (Figura 7.2).

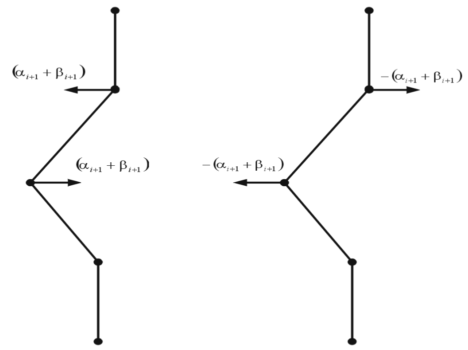

Figura 7.2: Forças internas com parâmetros positivos (esquerda) e negativos (direita).

A força de tensão dada por (5.52) tem como uma de suas finalidades uniformizar a distribuição dos snaxels ao longo da snake ao tentar evitar que os mesmos se aproximem (ou se distanciem) além do comprimento natural l_i usado na definição da força de mola (equação 5.52).

No entanto, a reparametrizacão das T-Snakes é feita partindo-se da projeção da snake sobre a triangulação. Logo uma snake com distribuição uniforme de pontos pode passar a ter uma distribuição pouco uniforme após a reparametrização, como mostra a Figura 7.8 a seguir.

Considerando estas observações, nós optamos primeiramente por diminuir o número de parâmetros, facilitando assim sua escolha, em detrimento (possivelmente) de uma maior suavidade do resultado final do Dual-T-Snakes. Neste sentido, usamos apenas rigidez, força normal e força de imagem, colocando nulos todos os outros parâmetros.

Mesmo assim, devemos ficar atentos para o problema indicado na Figura 7.1, onde vemos as forças de rigidez e normal apontando ambas para a borda no caso da snake externa mas apontando em direções distintas no caso da interna. As definições da força normal serão a chave para tratar esta dificuldade, como veremos a seguir.

O modelo final a ser resolvido numericamente tem a forma:

$$v_i^{(t+\Delta t)} = v_i^t + \frac{\Delta t}{\gamma}\left(\beta_i^t + F_i^t + f_i^t\right), \tag{7.1}$$

o qual corresponde ao modelo de balloons dado pela expressão (3.63), com $\omega_2 = 0$, discretizado por diferenças finitas usando-se um método idêntico àquele da secão 3.2.3. A análise numérica portanto fornece a mesma conclusão obtida na seção 3.2.3; ou seja, quanto menor o intervalo de tempo τ maior a estabilidade numérica.

Neste trabalho, a determinação exata dos parâmetros envolvidos foi feita de forma empírica. Estes são consideramos como constantes e o modelo é submetido a uma etapa de calibração na qual fazemos uma série de simulações computacionais a fim de encontra valores ótimos para estas constantes.

7.2 Passo Máximo

As energias de imagem do Dual-T-Snakes dadas pelas expressões (6.1) e (6.2) fazem sentido somente se não permitirmos passos altos da snake em uma interação. A Figura 7.3 ajuda esclarecer este fato. Observe que, uma vez que o potencial externo depende do contraste da imagem, nos patamares indicados na figura a energia de imagem é nula, e assim se o passo for alto, passando de um patamar para o outro em uma interação, não haverá variação da energia.

Uma forma de evitar esta dificuldade seria usar energias que dependem da própria intensidade imagem (e não do seu contraste), uma vez que a força normal necessita de um limiar para ser definida. Contudo, aumentar o passo de interação torna o movimento da snake mais insável, principalmente para imagens com ruídos.

Assim, tomamos a iniciativa de estabelecer um passo máximo da ordem da distância entre dois pixeis da imagem (*uma unidade*). Desta forma, a evolução das

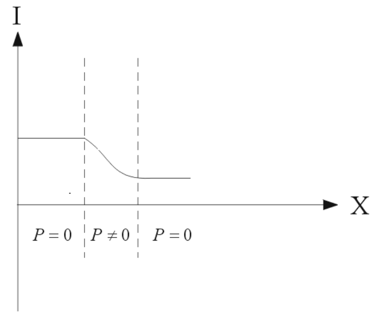

Figura 7.3: Com passo muito alto, a snake pode avançar o degrau perdendo a borda.

snakes se faz de forma mais controlada. Ao mesmo tempo, melhoramos a estabilidade numérica e consideramos apropriadamente as informações da imagem evitando o problema representado na Figura 7.3.

7.3 Força Externa e Força Normal

Um ponto importante a ser discutido é a política adotada para estas duas forças no Dual-T-Snakes.

Assim como no Balloon, usamos a força externa normalizada segundo a expressão (3.57). Esta escolha tem mostrado bons resultados facilitando o controle da snake durante sua evolução.

Exatamente como no caso do Balloon, a força normal nas T-Snakes é usada para fazer as snakes convergirem para as bordas de interesse. Contudo, na vizinhança de uma borda, as T-Snakes e o Balloon funcionam distintamente. A Figura 7.4 representa como a força de imagem e a força normal atuam na vizinhança de uma borda nestes dois modelos.

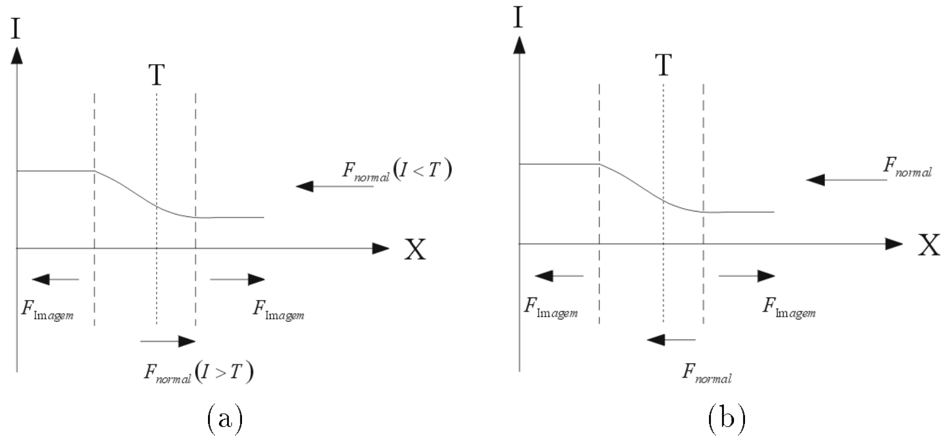

Figura 7.4: (a)Forças de imagem e normal para as T-Snakes. (b)Forças de imagem e normal para o Balloon.

No caso do Balloon não temos inversão da força normal após o limiar T. Por este motivo, os coeficientes κ_1 e κ_2 devem ser da mesma ordem de grandeza a fim de que tenhamos um equilíbrio das forças correspondentes antes da força de imagem inverter seu sentido devido a forma do potencial P definido em (3.3). Este equilíbrio implica em deslocamentos pequenos da snake na vizinhança da borda e portanto é percebido usando-se o critério de parada dado pela expressão (3.48).

Este procedimento não pode ser aplicado para as T-Snakes devido seu método de reparametrização. A Figura 7.5 mostra os snaxels acomodados sobre a borda após uma evolução dada pela equção 7.1 e a T-Snake final após a reparametrização. Esta *perturbação* da snake impede que um critério de parada baseado em deslocamentos pequenos seja utilizado, o que em nossa opinião motivou a idéia do ponto de congelamento usada em (McInerney, 1997; McInerney & Terzopoulos, 1999). Isto fornece um critério de parada eficiente, mas, do ponto de vista do Dual-T-Snakes, traz um problema: o número de interações pode ser muito alto caso as snakes fiquem congeladas várias vezes durante sua evolução.

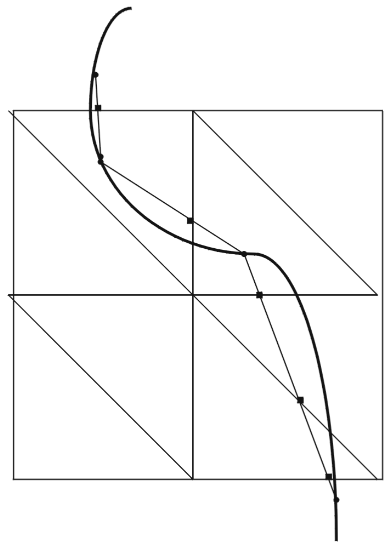

Figura 7.5: Snake antes e depois da reparametrização.

Portanto, se aplicarmos o modelo de Balloons ao Dual-T-Snakes poderemos ter uma perda significativa de performance, uma vez que o Balloon é muito sensível a mínimos locais da imagem. Isto é verificado principalmente para imagens com muito ruído e artefatos no background, que constituem exatamente a principal classe de aplicações do Dual-T-Snakes.

Mesmo a definição (5.54) para a força normal tem suas limitações para imagens deste tipo. Nestes casos, a inversão da força normal usada na definição (5.54) pode conduzir a muitas oscilações da snake decorrentes da não homogeneidade da imagem. A Figura 7.6 ilustra tal situação.

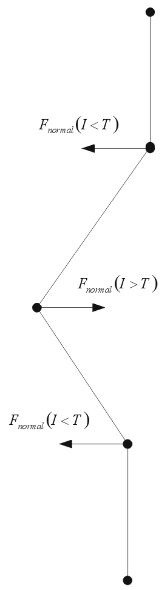

Figura 7.6: Comportamento da força normal sobre a snake podendo gerar oscilações.

Neste caso, poderemos ter oscilação dos snaxels representados o que pode deixar a snake *preza* na região indicada. Se este comportamento se repete constantemente, o número de vezes que a snake atinge o ponto de congelamento aumenta muito.

Considerando estes fatos, a solução que encontramos foi redefinir a força normal como segue:

$$F_i(v_i) = Val(v_i)\, kn_i, \tag{7.2}$$

onde:

$$Val(v_i) = 1 \text{ se } I(v_i) < T, \tag{7.3}$$
$$Val(v_i) = 0 \text{ caso contrário}, \tag{7.4}$$

sendo n_i a normal unitária no vértice v_i, k o fator que define a intensidade da força e T o limiar para a imagem I.

Desta forma, não teremos a tendência a oscilar, como na Figura 7.6. Por outro lado, o efeito de *puxar* a snake para a borda é mantido, uma vez que os snaxels com força normal não nula avançam a snake, podendo retirar os demais de mínimos locais decorrentes de irregularidades na imagem. Certamente ainda podemos ter um congelamento da snake fora da região desejada, contudo o número de vezes que isto acontece é sensivelmente reduzido.

Os testes mostraram que a definição 7.3 é mais eficiente para imagens com ruído, o que veremos no Capítulo 9.

Um ponto importante é que agora não precisamos manter os coeficientes da força normal e de imagem com a mesma ordem de grandeza, uma vez que próximo da borda a força normal será nula. Este é outro aspecto positivo da definição acima para a força normal uma vez que estamos aproveitando esta força apenas para aproximar a snake da região de interesse. Quando próximo da borda, a força normal é anulada pela definição 7.3 e o equilíbrio fica dependendo apenas das forças elástica e externa. Evitamos assim a preocupação com o equilíbrio mais delicado entre a força normal e a força de imagem observado no Balloon (Figura 7.4).

Por outro lado, a suavidade do resultado final pode ser afetada, o que é discutido a seguir.

7.4 Suavização e Forças Internas

Um ponto importante a ser ressaltado com a relação à definição (7.3) é o aparecimento de oscilações da snake quando próximo das bordas. A Figura 7.7 é representativa deste comportamento.

Nesta figura, temos um seqüência de snaxels alinhados e espaçados uniformemente. Neste caso, as forças internas T-Snakes dadas pelas expressões das (5.52 e (5.53) são nulas (o mesmo acontecendo com as expressões discretas (3.60) e (3.61) do Balloon).

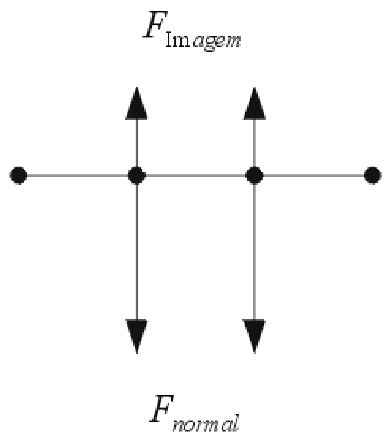

Figura 7.7: As forças internas nulas devido alinhamento e distribuição uniforme de snaxels.

Portanto, neste caso o sistema de forças sobre a snake fica resumido às forças normal e de imagem. Se a snake estiver próxima de uma borda, a definição (7.3) implica que a força normal poderá ser nula para alguns snaxels e portanto teremos apenas força de imagem atuando sobre os mesmos.

Neste caso, nos snaxels onde atua apenas a força de imagem, teremos um *salto*.

Embora limitado a no máximo uma unidade, este salto pode diminuir a suavidade da curva na região.

Este é um *defeito* da definição (7.3). Contudo, do ponto de vista do Dual-T-Snakes, este problema não é crítico uma vez que este método é usado apenas para limitarmos a região de interesse, na qual será definido o espaço de busca. A perda de suavidade indicada não tem se mostrado prejudicial para a determinação posterior do espaço de busca. Além disto, as oscilações alteram a temperatura dos snaxels fazendo com que os mesmos atinjam o ponto de congelamento. O critério de parada das T-Snakes é portanto eficiente nesta proposta.

É oportuno destacar que no método de Balloon a suavidade do resultado final é preservada primeiramente porque a força normal não é desligada nas proximidades da borda. Em segundo lugar, o fato da força norma e de imagem apresentarem intensidades da mesma ordem de grandeza impede *saltos* do tipo discutido acima quando estamos próximos da borda. Possíveis defeitos de suavidade que ocorram em uma interação, são em geral corrigidos pelas forças de tensão e rigidez as quais atuam como um filtro passa-baixa sobre a curva.

O efeito deste filtro pode ser melhor compreendido se tomarmos o sistema linear (3.64) e usarmos a inversa da matriz $[I + \tau K]$ para isolar X^n (idem para Y^n):

$$X^n = [I + \tau K]^{-1} \left\{ X^{n-1} + \tau \left[R_X \left(\bullet, \bar{t} - \tau \right) - N_X \left(\bullet, \bar{t} - \tau \right) \right] \right\}, \tag{7.5}$$

A matriz inversa $[I + \tau K]^{-1}$ produz o efeito de suavização que corta freqüências mais altas da curva. Este efeito é mais complicado de ser obtido no Dual-T-Snakes.

Além das dificuldades relativas à força normal, uma outra justificativa para este problema é com relação ao termo de quarta ordem da equação (3.63). A versão discreta deste termo poderia ser acrescentada às T-Snakes, aumentando a *rigidez* das mesmas.

Contudo, maior rigidez pode implicar em mais dificuldade para expandir a snake interna, o que pode trazer dificuldades para calibrar a força normal. Acrescenta-se a isto a não uniformidade da distribuição de snaxels sobre a snake, característica do método de reparametrização usado. Isto implica, que setores lineares da snake podem apresentar snaxels distribuídos de forma não homogênea (Figura 7.8) criando irregularidades na curva durante sua evolução.

Considerando ainda que nosso objetivo com o Dual-T-Snakes é apenas limitar a região de interesse acreditamos mais interessante demonstrar a potencialidade do método sem termos de quarta ordem.

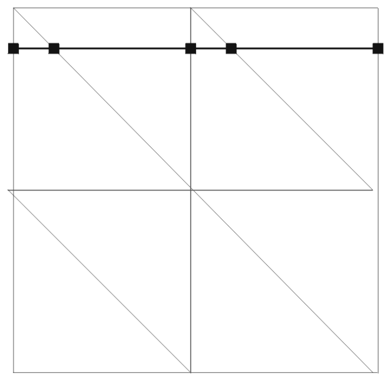

Figura 7.8: Distribuição não-uniforme de snaxels devido reparametrização.

7.5 Resolução da Triangulação

A resolução da triangulação usada está vinculada dentre outros fatores à escala dos objetos de interesse.

Uma metodologia sistemática para definir esta resolução poderia ser útil não apenas para a eficiência do método mas também para a implementação de técnicas multi-grid para T-Snakes e, conseqüentemente, para o Dual-T-Snakes também.

Tal procedimento é altamente dependente da aplicação. Por exemplo, vamos supor como em (Davatzikos & Prince, 1995) que conhecemos a priori que a curvatura dos objetos de interesse na cena tem como limite superior um valor k_0.

Vamos então tomar a seguinte série de Fourier para a fronteira procurada:

$$x(s) = X_0 + \sum_{l=1}^{\infty} \left\{ X_{c_l} \cos(2\pi l s) + X_{s_l} \sin(2\pi l s) \right\} \tag{7.6}$$

$$y(s) = Y_0 + \sum_{l=1}^{\infty} \left\{ Y_{c_l} \cos(2\pi l s) + Y_{s_l} \sin(2\pi l s) \right\} \tag{7.7}$$

Seguindo (Davatzikos & Prince, 1995) vamos supor que somente uma freqüência tem energia. Assim, ignorando possíveis sobreposições, translações e o ponto inicial, as funcões coordenadas (7.6)-(7.7) tornam-se:

$$x(s) = \frac{1}{k} \cos(Lks), \tag{7.8}$$

$$y(s) = \frac{1}{k} \sin(Lks), \tag{7.9}$$

onde L é o comprimento total da borda. Uma vez que $k \leq k_0$, a freqüência máxima na qual esta função pode existir será:

$$l_{\max} = Lk_0/2\pi. \tag{7.10}$$

Para nosso propósito, este desenvolvimento simplificado é suficiente. Pela teoria de amostragem (Jain, 1989), a freqüência $l_{\max}$ significa que o período de amostragem deve ser:

$$T_{sampling} = \frac{2\pi}{l_{sampling}} \leq \frac{2\pi}{l_{\max}} = 4\pi^2/Lk_0, \tag{7.11}$$

o que implica que o número de pontos amostrados $N_{sampling}$ deve satisfazer:

$$N_{sampling} \geq \left\lfloor \frac{L^2 k_0}{4\pi^2} \right\rfloor. \tag{7.12}$$

Como conseqüência, um método multi-grid poderia ser implementado. Por exemplo, poderíamos partir de uma malha com resolução definida pela expressão (7.12) e, após a estabilização da T-snake, refinar a malha e evoluir a T-snake novamente, obtendo agora um ajuste mais fino.

Quando projetamos uma curva sobre a triangulação, além da não-uniformidade da distribuição de pontos obtidos, poderemos ter pontos muito próximos o que pode trazer problemas numéricos.

Para evitar esta dificuldade, optamos por estabelecer primeiramente uma distância mínima para os snaxels que evite os problemas numéricos. Após a reparametrização da snake, analisamos a distância entre cada snaxel e seu vizinho, e caso esta seja menor que o valor mínimo permitido, substituímos estes snaxels pelo nó da malha mais próximo (Figura 7.9).

Além disto, estabelecemos uma distância mínima usada no cálculo da força da expressão (5.53). Se dois snaxels vizinhos possuem distância menor que esta última, tomamos o snaxel seguinte.

Observando a expressão (5.53), vemos que, aumentar a resolução da malha do Dual-T-Snakes implica (de forma geral) em reduzir forças internas da snake, a menos que alteremos os parâmetros correspondentes.

Esta perda de rigidez da snake pode ser interessante para que a mesma avance para regiões com curvatura acentuada.

7.6 Condição de Entropia

Outra questão importante é a maneira pela qual implementamos a condicão de entropia: *uma vez que um nó é queimado ele permanecerá queimado.*

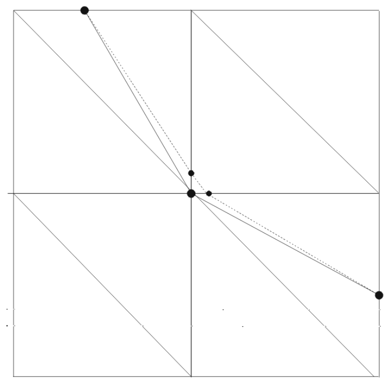

Figura 7.9: Snaxels muito próximos após reparametrização.

O ponto central aqui é a política usada para satisfazer esta condição.

O primeiro passo neste sentido é não alterar o valor da função característica em nós já queimados.

Assim, uma vez movimentada a snake segundo a equação de evolução usada, poderemos ter uma situação como a da Figura 7.10 onde a snake avançou sobre nós já queimados.

Quando projetada sobre a triangulação, os pontos correspondentes à região tracejada ficarão em triângulos onde a função característica não muda seu valor, e portanto não serão considerados durante a aplicação do algoritmo de classificação descrito na seção 5.3.2.

Neste caso, teremos que *corrigir* a snake nestas regiões, o que pode ser feito, por exemplo, substituindo-se os três snaxels indicados pelo nó apontado. Esta é a política por nós adotada.

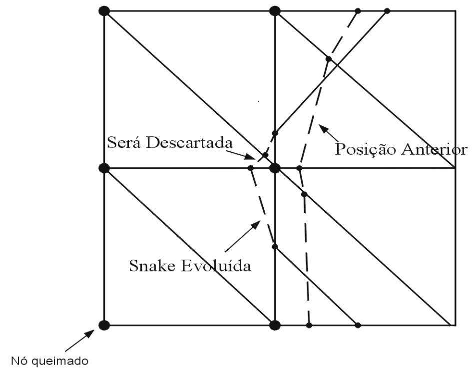

Figura 7.10: Snake avançou para região onde função característica não muda de sinal.

Uma outra possibilidade seria evitar que a snake passe por nós da triangulação já queimados usando testes locais, antes que a snake seja movimentada. Esta política no entanto não diminui a complexidade do método e suas vantagens não são aparentes.

7.7 Pré-Processamento da Imagem

Os métodos de snakes são em geral precedidos de algum tipo de tratamento da imagem. Isto se faz necessário tanto para reduzir ruídos quanto para realçar as bordas.

No caso do Dual-T-Snakes esta questão está diretamente vinculada com o operador de afinidade.

Existe um detalhe de natureza prática a ser apontado inicialmente.

Consideremos a Figura 7.11 . Enquanto o interior do objeto é muito ruidoso, o background é simples. Assim, um método de snakes usual, com a snake inicializada no background poderia ser eficiente e apresentar melhor performance.

Figura 7.11: Imagem contendo objeto com interior complicado mas background simples.

A relação custo/benefício do método dual é mais vantajosa em situações como a da Figura 7.12 onde temos o interior e o exterior do objeto de interesse complicados.

Este fato é importante para definir o tipo de pré-processamento da imagem, caso este se torne necessário. A idéia é usar filtros com pouca localização no espaço a fim de criar uma *faixa* contendo a borda desejada. Esta faixa será extraída via o Dual-T-Snakes e posteriormente a programação dinâmica fará o resto.

Dentre os métodos que podem fornecer tal resultado, temos filtros passa alta (Canny, 1983), estatística para a imagem (Jain, 1989) e método das conexões fuzzy (Udupa & Samarasekera, 1996; Jones & Metaxas, June, 1998), este último sendo particularmente interessante por aproveitar a interação com o usuário na inicialização do Dual-T-Snakes (Apêndice E).

A exemplo do Balloon, fazemos interpolação bilinear da força externa F para termos uma definição contínua da mesma.

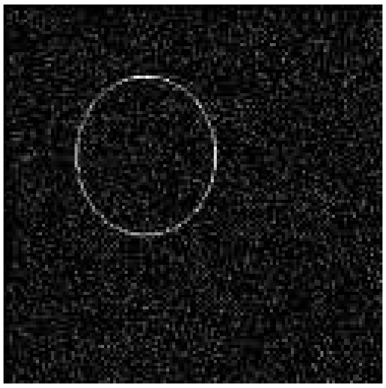

Figura 7.12: Imagem contendo objeto imerso em background com ruído.

7.8 Evitando Mínimos Locais

Uma vez detectado que uma snake congelada não está próxima da borda (critério de proximidade da seção 6.1) o passo seguinte é intervir em seu movimento fazendo com que mesma saia do ponto de congelamento.

Neste trabalho, consideramos três possibilidades:

1. Aumentar a normal,
2. Variar o Limiar,
3. Mudar a resolução da malha.

A primeira política não é interessante para imagens com muito ruído. Isto decorre do fato de estarmos usando a força normal dependente do limiar da imagem.

Como exemplo, consideremos novamente a Figura 7.6. Se o comportamento representado se repetir ao logo da snake então, pela definição (5.54), teremos oscilação da mesma sem retirá-la do congelamento. A definição (7.3) seria mais eficiente, uma vez que esta não inverte o sentido da normal e portanto não cria oscilações. Contudo, a sensibilidade da T-Snake com relação a mínimos locais aumenta o que pode implicar em perda de performance.

Esta política é útil em imagens com muitos artefatos no background mas pouco ruído. Nestes casos, o aumento da força normal tem o papel básico de vencer localmente a resultante das forças de suavização e de campo, tirando a snake de mínimos locais da energia de imagem.

Em imagens muito ruidosas, a variação do limiar da imagem pode ser mais eficiente uma vez que isto pode liberar snaxels presos pelo limiar adotado, perturbando a snake e permitindo que esta avance em direção à borda. Contudo, esta tem a desvantagem de exigir uma estatística da imagem. Além disto, o limiar deve ser relaxado com cuidado para não criar *falhas* muito grandes na borda, o que poder faze com que uma snake parcialmente estabilizada avance além da mesma.

Como já comentado acima, a mudança na resolução da malha altera as forças internas, mudando a resultante de forças sobre cada snaxel. Esta alternativa é particularmente interessante para objetos cujas bordas possuem pontos com alta curvatura.

7.9 Detalhes da Implementação do Dual-T-Snakes

Nesta seção, apresentamos detalhes da implementação que fizemos do Dual-T-Snakes. Nossa implementação é seqüencial, feita para estações IBM e usando os recursos para desenvolvimento de softwares e visualização do Data Explorer (Corporation, 1993).

O Data Explorer é um tipo particular de Ambiente Modular de Visualização (*Modular Visualization Environments: MVE*) o que grosseiramente remete às características comuns dos mesmos (Cameron, 1995):

.Modular

  .Constituído por blocos de código (Módulos).

  .Metodologia orientada a objetos.

  .Novos módulos podem ser criados pelo usuário.

.Ambiente

  .Tipicamente, um ambiente de programação visual.

  .Permite integração e criação interativa de programas.

.Visualização

  .O objetivo final dos programas criados é tipicamente visualização de dados.

Estes sistemas tem como vantagens a interface com o usuário simplificada pelo ambiente de programação visual, a possibilidade de criação de novos módulos e integração destes com outros já disponíveis.

Os MVE têm como limitações a dificuldade de integração entre visualização e simulação e a dificuldade de manipulação de grandes volumes de dados.

Considerando estes fatores, optamos por utilizar um sistema do tipo MVE para o desenvolvimento da implementação do Dual-T-Snakes pois acreditamos que os recursos de programação visual, programação distribuída e modularidade destes sistemas trazem bons resultados.

Assim como no Capítulo 6, chamaremos *Dual Snake* a uma lista de T-Snakes onde a primeira é uma snake externa e as demais são snakes internas a esta.

O programa possui basicamente três estruturas: a fila de dual snakes a serem processadas (*group_dual_in*), a fila de dual snakes ancoradas (*final_result*) e uma *hash table* usada durante a execução do algoritmo para efetuar mudanças topológicas.

O Dual-T-Snakes tem complexidade $O(N_T)$, onde N_T é o número de triângulos da malha.

Cada entrada da hash table tem uma *chave* dada por um valor inteiro calculado pela expressão:

$$chave = 2 * id1 + triangle\_id + 2 * counts\_y * id\_x, \tag{7.13}$$

onde $(id0, id1)$ são as coordenadas (inteiras) do nó esquerdo inferior da célula que contém o ponto e $triangle\_id = 0$ se o ponto pertence ao triângulo inferior e $triangle\_id = 1$ caso contrário (estamos supondo triangulações CF, como a da Figura 7.10).

Na inicialização do loop principal do algoritmo as snakes são projetadas sobre a triangulação, para em seguida aplicarmos um algoritmo de scan line que inicializa a função característica.

Uma vez evoluída a snake segundo a equação 7.1, esta função será atualizada segundo o algoritmo de classificação da seção 5.3.2. Uma vez projetada a snake sendo processada, seus pontos são distribuídos pela hash table para a finalização do método de reparametrizaçao das T-Snakes.

Cada ponto da snake projetada é guardado na entrada correspondente da hash table, juntamente com outras informação, segundo a estrutura a seguir:

```
typedef struct _Pqueue{
 double x;            /* abscissa do ponto */
 double y;            /* ordenada do ponto */
 int id0;             /* abscissa da célula */
 int id1;             /* ordenada da célula */
 int curve;           /* identificador da curva no dual snake */
 int flag;            /*Ponto Ancorado: flag=0; Freezing_Point: flag=2 */
 char triangle_id;    /* 0 para triângulo inferior e 1 para superior */
 int position;        /* Índice do ponto na snake */
 char edge;           /* 0 para horizontal; 1 para diagonal; 2 para vertical */
 struct _Pqueue *next;
} Pqueue;
```

Ao final deste processo, dada uma entrada da hash table, ou ela aponta para NULL ou tem o endereço da fila que guarda os snaxels pertencente ao triângulo correspondente. Esta hash table será percorrida e ao final teremos as snakes propriamente ditas (já reparametrizadas).

Isto foi implementado da seguinte forma. Uma vez encontrada uma entrada da hash table que não seja nula, escolhemos a aresta do primeiro ponto guardado como sendo a *door-in* e a próxima como sendo a *door-out*. Com isto, sabemos exatamente para qual entrada da hash table devemos seguir. Este processo se repete até que a primeira aresta tipo door-in obtida seja encontrada novamente; quando então, teremos uma componente conexa da snake completa.

Uma vez que as dual snakes são independentes entre si, a sua escolha para o processamento é feita seguindo simplesmente a ordem em que aparecem na fila *group_dual_in*.

Selecionada uma duas snake, a curva a ser processada é definida segundo os critérios de energia e afinidade da seção 6.1.

Se uma snake sofreu mudança topológica, as dual snakes por ela definidas são colocadas no final do *group_dual_in* para serem processadas posteriormente.

O algoritmo possui dois loops que compõem o loop principal.

No primeiro, uma snake congelada será processada usado-se algum dos critérios (1)-(3) acima para retirar a snake do mínimo local correspondente. Uma vez feito isto, o segundo loop evolui a snake até que o próximo ponto de congelamento seja atingido ou uma mudança topológica ocorra. Neste último caso, o group_dual_in é atualizado com as dual snakes resultantes e volta-se ao inicio do loop principal. A equação de evolução é dada pela expressão 7.1.

O fluxograma a seguir esclarece como estes elementos foram organizados na implementação do Dual-T-Snakes.

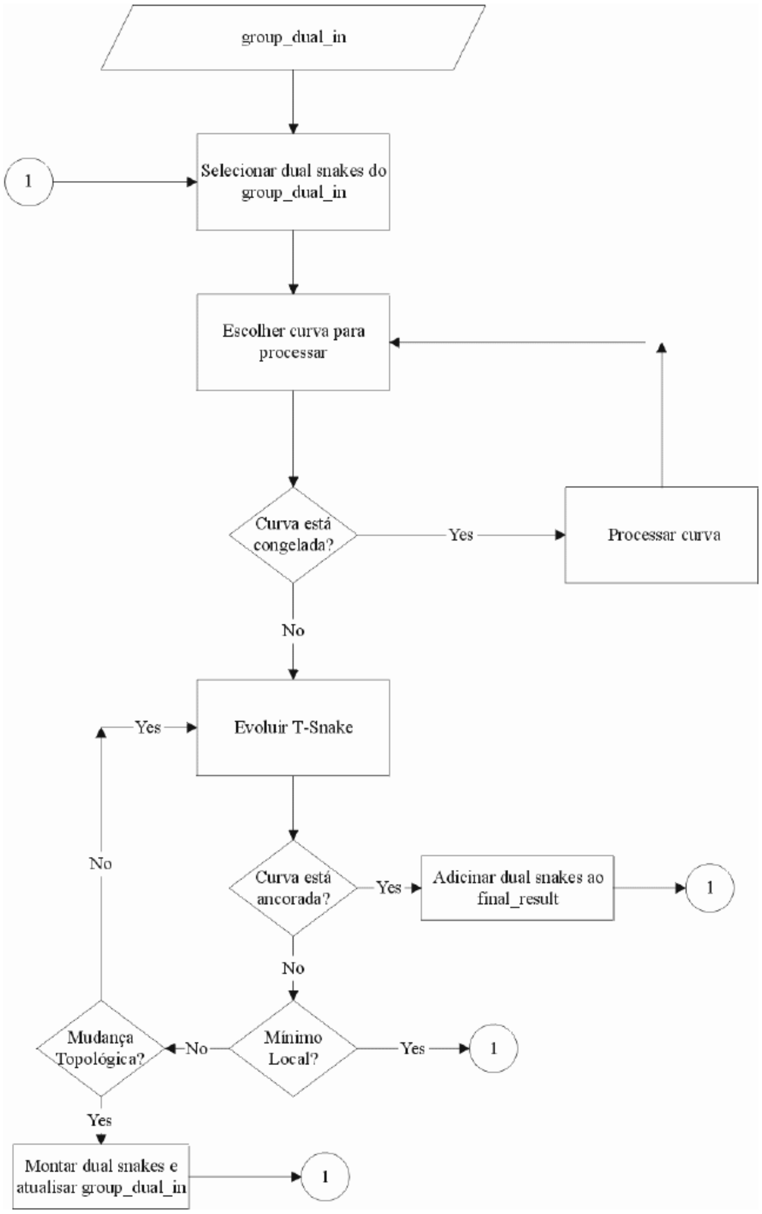

Figura 7.13: Fluxograma descrevendo as principais partes da implementação do Dual-T-Snakes.