Capítulo 6

Dual-T-Snakes

Apesar das habilidades topológicas, o método das T-Snakes também é afetado pelos problemas causados pela não-convexidade da energia de imagem. A utilização de estatísticas para redefinir o sentido da força normal (expressão 5.55) diminui a sensibilidade do método com relação a mínimos locais mas não resolve o problema. A dificuldade está em determinar automaticamente se uma T-Snake congelada atingiu ou não a borda desejada.

Uma forma de diminuir as dificuldades com a não convexidade é através de uma inicialização eficiente (próxima da borda), como destacamos em (Giraldi *et al.* , 2000e; Giraldi *et al.* , 2000a). Quanto mais próxima estiver a solução inicial do mínimo global, mais facilmente o método pode atingi-lo.

No Capítulo 8 nós apresentamos uma contribuição neste sentido, toda ela baseada na metodologia das T-Snakes e no conceito de *propriedade de escala local* ali definido.

Embora útil, o que será demonstrado no Capítulo 9, a necessidade de uma escala local para os objetos de interesse, o que basicamente significa que os padrões de intensidade definem regiões conexas da imagem, limita a classe de aplicações deste método.

Métodos multi-escala tais como (Leymarie & Levine, 1993) podem ser outra possibilidade para tratar a não-convexidade da energia, uma vez que a passagem de uma resolução mais grosseira para outra mais fina perturba a snake facilitando com que a mesma evolua em direção à região desejada. Outras abordagens encontradas na literatura para a inicialização automática de modelos deformáveis em geral fazem uso de Transformadas de Hough (Lai & Chin, 1995), redes neurais e momentos (Williams *et al.* , 1997).

O método dual da seção 5.1.1 é certamente o mais robusto com relação a mínimos locais, dentre os modelos apresentados no Capítulo 5. Contudo, sua versão evolucionária depende do estabelecimento a priori de modelos de forma (ainda que locais). Além disso, a necessidade de correspondência entre os pontos das snakes é outro fator limitante do método. Por outro lado, na sua versão baseada em programação dinâmica (seção 5.2), o método dual apresenta a dificuldade clara com relação à inicialização da snake interna e externa a fim de que o espaço de busca possa ser criado eficientemente. Contudo, este método é o mais robusto pelas propriedades de otimização global da programação dinâmica (seção 4.1).

Assim, nossa motivação inicial para o desenvolver o método descrito a seguir foi *generalizar o método Dual Evolucionário*, de maneira a prescindir de um modelo de forma pré-estabelecido e evitando a necessidade de correspondência entre as snakes interna e externa.

O ponto inicial do novo método veio da observação de que a triangulação das T-Snakes fornece uma forma simples de verificar proximidade de curvas. Ao mesmo tempo, a força normal nas T-Snakes pode ser usada no lugar da força adicional. A definição daquela pode ser usada na decisão de qual snaxel movimentar em uma snake congelada.

Resumindo, a proposta foi colocar o método Dual e as T-Snakes em um modelo unificado. Como resultado, conseguimos uma nova metodologia que é dual e topológica ao mesmo tempo, daí denominá-la *Dual-T-Snakes*. Além disto, podemos agora generalizar o Dual Evolucionário para 3D, como indicaremos a seguir.

Um derivado interessante do Dual-T-Snakes é a redução do espaço de busca, podendo assim ser utilizado para inicializar o Dual Baseado em Programação Dinâmica, vencendo com isto a dificuldade com a inicialização deste último citada acima.

A próxima seção descreve o novo método em detalhes para o caso bidimensional. Indicamos também como o Dual-T-Snakes pode ser estendido para 3D. A seguir, mostramos como usar o resultado do método para inicializar um algoritmo de Viterbi descrito na seção 4.1. Finalmente, apontamos as vantagens e desvantagens do novo método.

6.1 Descrição do Dual-T-Snakes

Nesta seção apresentamos o método Dual-T-Snakes (Giraldi *et al.* , 2000d; Giraldi *et al.* , 2000c), o qual é uma contribuição deste trabalho.

A idéia básica desta nova metodologia é explorar o modelo da T-Snakes para propor um *Modelo Dual de Contornos Ativos Generalizado*: um *contorno externo*, envolvendo os objetos de interesse, contrai e se divide enquanto que *contornos internos* aos objetos se expandem em direção às bordas procuradas.(Figura 6.1).

Uma vez que a posição inicial das snakes será em geral longe das bordas desejadas, sofrendo inclusive mudanças topológicas durante sua evolução, não é interessante usarmos processos de minimização explicita para evoluir as snakes. Assim a escolha do modelo de snakes da seção 5.3.2 é imediata.

Para fazer a snake externa contrair e as snakes internas expandirem em direção às bordas dos objetos nós usamos forças normais do tipo daquela apresentada na

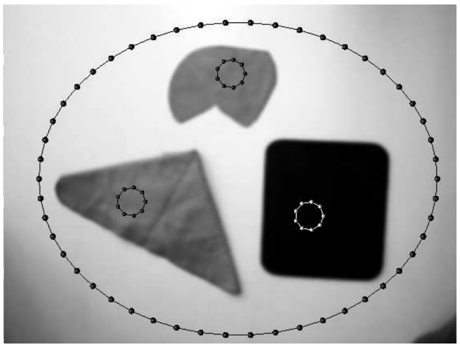

Figura 6.1: Inicialização do Dual-T-Snake.

equação 5.54.

Além disso, é necessário definir elementos que tornem a evolução da snake externa e das internas interdependente a fim de tornar o método unificado. Neste trabalho isto é implementado usando-se a *energia de imagem (normalizada) sobre as snakes* e uma noção de *afinidade*.

Especificamente, nós usamos duas definições para energia de imagem sobre as snakes: uma para o contorno externo $((E_{outer}))$ e outra para o conjunto de contornos internos envolvidos por este $((E_{inner}))$:

$$E_{outer} = \sum_{i=0}^{N-1} \left(- \left\| \nabla I \left(v_i \right) \right\|^2 \right) / N, \tag{6.1}$$

$$E_{inner} = \frac{1}{m} \left(\sum_{k=0}^{m} \left(\sum_{i=0}^{N_k-1} \left(- \left\| \nabla I \left(v_i \right) \right\|^2 \right) / N_k \right) \right), \tag{6.2}$$

onde m é o número de curvas internas, N_k é o número de snaxels da snake interna k, N é o número de snaxels da snake externa, I é a intensidade de imagem, suposta em tons de cinza, a menos de mensão explícita ao contrário.

Se:

$$E_{inner} > E_{outer} \tag{6.3}$$

então uma curva interna deve ser evoluída. Para tal, nós primeiro usamos um *operador de afinidade* para estimar os pixels da imagem mais prováveis de pertencer às bordas dos objetos de interesse. Com base neste operador nós podemos associar a cada snaxel um valor indicando sua proximidade da borda. O campo sobre a snake assim obtido é então binarizado pela comparação com um limiar, assumindo valor zero para snaxels próximos da borda e 1 caso contrário.

Portanto, se a expressão (6.3) é verdadeira, a snake interna com maior número de snaxels com afinidade não nula é escolhida.

Por outro lado, se :

$$E_{outer} > E_{inner} \tag{6.4}$$

então a snake externa será evoluída, caso sua função de afinidade não seja identicamente nula.

Uma vantagem de usarmos um operador de afinidade é simplificarmos a imagem reduzindo assim as dificuldades para a escolha de parâmetros para o modelo.

Dentre as possibilidades para definir o operador de afinidade podemos citar a utilização de um limiar T para a intensidade de imagem, estatísticas, e métodos fuzzy (Apêndice E). Esta última possibilidade é interessante por tirar mais vantagem da interação com o usuário na inicialização do Dual-T-Snakes (posicionamento das snakes internas e externas).

O próximo ponto é avaliarmos a similaridade entre dois contornos. A idéia é tomarmos vantagem da metodologia de subdivisão do espaço da T-Snake: a diferença entre as Funções Características da curva externa e da curva interna (denominada *Characteristic-Diff*) fornece uma noção (local) da proximidade entre as snakes, como mostra a Figura 6.2.

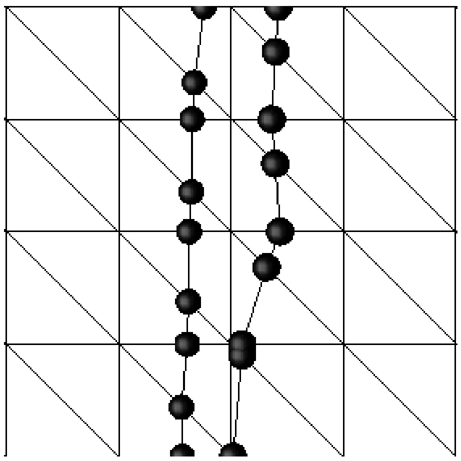

Figura 6.2: Proximidade entre duas snakes medida pela triangulação.

Por exemplo, no caso da triangulação CF desta figura, nós poderíamos interromper o movimento de todos os snaxels de uma snake interna que estão no interior de um triângulo σ se algum dos vértices $v \in \sigma$ tem as seguintes propriedades:

Propriedade (a). Todos os 6 triângulos adjacentes a v têm um vértice onde $Characteristic\_Diff = 0$;

Propriedade (b). Um destes triângulos é cortado pelo contorno externo correspondente

Contudo, a política de interromper o movimento de regiões da snake pode não ser eficiente por introduzir *defeitos* sobre o resultado final (Figura 6.4). Na implementação do método há apenas um *flag* indicando que as condições (a)-(b) acima foram verificadas ($flag = 0$, neste caso). A execução do método termina quando todos os *flags* são nulos.

Uma vez que a condição de entropia da seção 2.1 é adotada, nós podemos associar um ponto interno a cada snake interior. Quando a snake externa sofre um split em duas ou mais snakes estes pontos podem ser usados para identificar qual snake interna está envolvida por uma snake externa gerada na operação de split.

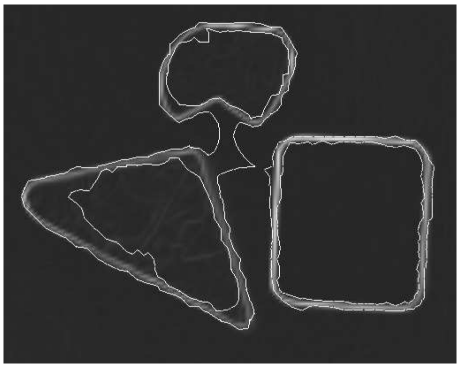

Figura 6.3: Split da Dual Snakes inicial.

O ponto de congelamento é usado para indicar quando uma T-Snake está em uma posição de equilíbrio.

É importante ressaltar que o balanço entre energia/afinidade, juntamente com a noção de proximidade entre as snakes descrita acima, permitem evitar mínimos locais. De fato, se uma T-Snake foi congelada nós podemos aumentar a força normal para os snaxels com afinidade não nula e $flag \neq 0$.

No algoritmo a seguir denominamos uma *Dual Snake* a uma lista de T-Snakes onde a primeira á uma snake externa e as demais são snakes internas a esta. O método pode ser resumido como segue:

Dual-T-Snakes:

1.Colocar todas as dual snakes em uma fila.

2.Até que esta fila fique vazia fazer:

2.1. Retirar uma dual snake da fila.

2.2. Usar as energias das expressões (6.1) and (6.2) e a função de afinidade para encontrar a snake a ser processada.

2.3. Se todos os snaxels da snake escolhida estão congelados aumentar a força normal naqueles com afinidade zero e $flag \neq 0$ até que a energia da snake comece a decrescer.

2.4. Remover a força adicional e deixar a T-Snake evoluir normalmente

2.5 Se a snake sendo processada sofrer mudança topológica, montar as dual snakes resultantes e retornar ao passo 2.1.

2.6 Quando a snake estive congelada, analisar se a mesma está próxima de uma snake de outro tipo (interna/externa). Neste caso, remover a dual snake correspondente da fila.

2.7. Caso contrário, montar a dual snake(s) resultante e voltar ao passo 2.2.

Quando uma snake externa sofre um split, cada parte conexa gerada, juntamente com as snakes em seu interior, formarão uma nova dual snake.

Cada dual snake é completamente independente das outras o que possibilita implementações em arquiteturas paralelas. A Figura 6.4 mostra o resultado correspondente à Figura 6.1.

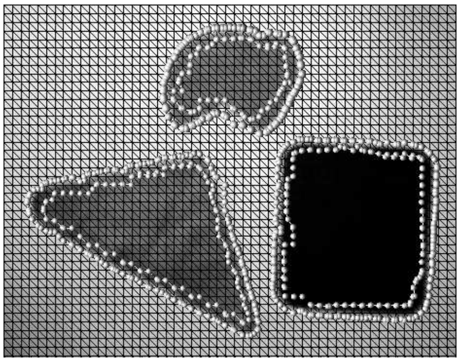

Figura 6.4: Dual-T-Snake result for figure 6.1. A resolução da grade é 5×5 e os snaxels são os pontos em branco.

Para extender o método desta seção para 3D basta usar o modelo das T-Superfícies (Apêndice C) em lugar das T-Snakes. Os demais elementos do modelo correspondente - Dual-T-Superfície - permaneceriam os mesmos do caso 2D descrito acima.

Embora não tenhamos feito testes para o caso 3D, acreditamos que o mesmo possa se constituir em uma nova contribuição para a área, como será ressaltado no Capítulo 10.

6.2 Metodologia para Segmentação

Uma vez completa a execução do Dual-T-Snakes, podemos tirar vantagem da redução do espaço de busca e aplicar um modelo de snakes baseado em programação dinâmica para obter a borda limitada por cada dual snake.

A idéia é aproveitar as habilidades topológicas e a robustez com relação aos mínimos locais do Dual-T-Snakes para inicializar o algoritmo de Viterbi (seção 5.2), uma vez que cada borda está limitada por uma dual snake (Figura 6.4). Assim, conseguimos vencer a limitação com relação à inicialização do Dual Baseado em Programação Dinâmica (Gunn, 1996b), aproveitando com isso as propriedades de otimização global deste último.

O resultado é uma metodologia robusta e eficiente para segmentação e extração de bordas.

Apesar destas vantagens, alguns cuidados devem ser tomados.

Primeiramente, na definição do espaço de busca do Viterbi, não podemos ter intersecção entre segmentos. Além disto, devido a irregularidades na solução do Dual-T-Snakes, podemos ter regiões com densidade de pontos mais baixa que outras.

Uma primeira possibilidade para evitar estes problemas seria utilizar métodos globais para estabelecer correspondência entre curvas (Geiger *et al.* , 1995). Contudo, isto implicaria em um custo extra ao modelo.

Uma observação oportuna vem do fato de que nós precisamos realizar duas buscas para obter o resultado final (seção 5.2). Assim, podemos inicialmente fazer um matching entre as curvas usando testes locais e de pouco custo computacional para evitar os defeitos mencionados. O espaço de busca assim obtido serviria para a primeira busca.

Em seguida, usamos o resultado obtido para redefinir o espaço de busca. Para isto basta tomar dois *offsets* para a solução encontrada na primeira busca, estabelecer um matching entre eles e criar um espaço de busca como anteriormente. Finalmente, aplicamos novamente a programação dinâmica (expressão (4.8)) para obter o resultado desejado.

Outra possibilidade seria usar a solução da primeira busca, discretizá-la regularmente e aplicar a relação de recorrência dada pela expressão (4.8) usando uma janela de busca centrada em cada ponto da curva discreta obtida, tal como em (Amini *et al.* , 1990).

Resumindo, o método das Dual-T-Snakes juntamente com o Viterbi formam uma metodologia para extração de bordas composta por quatro etapas:

(a) Inicialização das snakes internas e externas pelo usuário;

(b) Computação da função de afinidade

(c) Aplicação do Dual-T-Snakes para limitar região de interesse

(d) Extração das bordas desejadas pelo algoritmo de Viterbi

O método herda as habilidades topológicas das T-Snakes o que permite sua aplicação em casos onde a imagem consiste de vários objeto imersos em um background.

6.3 Vantagens do Dual-T-Snakes

Do ponto de vista da metodologia dual apresentada originalmente em (Gunn & Nixon, 1997; Gunn, 1996a), a primeira vantagem do Dual-T-Snakes é que este não necessita de "matching" entre as snakes internas e externas.

Uma vez que não necessitamos de correspondência entre as snakes, não precisamos limitar a forma da snake durante sua evolução, e portanto podemos prescindir de um modelo de forma estabelecido a priori.

No Dual original a força adicional não pode ser definida na direção normal à snake porque isto propicia a ocorrência de auto-intersecções na curva durante sua evolução, como mostra a Figura 6.5.

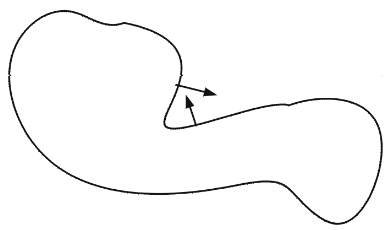

Figura 6.5: Evolução de curva na direção normal pode causar auto-intersecções.

No Dual-T-Snakes pudemos definir a força adicional na direção normal porque o processo de reparametrização das T-Snakes resolve auto-intersecções naturalmente.

Ao mesmo tempo, a imersão do dual na contexto das T-Snakes permitiu aplicar esta metodologia para imagens com vários objetos de interesse, aumentando seu campo de aplicações.

Portanto, o Dual-T-Snakes generaliza o Dual original, tanto para formas mais gerais quanto para topologias mais gerais.

Além disso, existe agora a real possibilidade de extender a metodologia dual para 3D via as T-Superfícies uma vez que não precisamos de correspondência entre os snaxels ou mesmo de um modelo de forma pré-estabelecido.

Do ponto de vista das T-Snakes, o Dual-T-Snakes trouxe efetivamente mais robustez quanto aos mínimos locais da energia de imagem, pelo balanço da energia e pela noção de proximidade entre as snakes internas e externas. Agora, o critério de parada baseado na similaridade entre as curvas é um indicador mais seguro de que o mínimo global foi atingido se comparado com o método de variação do limiar T usado na definição da força normal dada a partir da expressão (5.55)

Finalmente, a redução do espaço de busca permite evitar os problemas causados pelas necessidades de armazenamento em memória e processamento da programação dinâmica.

6.4 Desvantagens do Dual-T-Snakes

Uma questão importante a ser considerada é o passo máximo permitido para uma T-Snake durante uma interação da expressão (5.56). Como veremos no Capítulo 7, este passo deve ser limitado a no máximo a distância entre dois pixels vizinhos por interação, a fim de que a definição das energias de imagem dadas em (6.1)-(6.2) sejam eficientes.

Se comparado com a T-Snakes, isto implica em perda de performance uma vez que o passo de uma T-Snake pode ser maior que este valor, caso não tenhamos imagens com muito ruído.

Além disto, as limitações das T-Snakes apontadas na seção 5.3.2 vão se aplicar naturalmente ao Dual-T-Snakes.

Um ponto mais cuidadoso se refere à escolha de parâmetros do novo método, uma vez que temos movimentos distintos para as snakes interna e externa.

Basicamente, o problema está no fato de que as forças de suavização definidas por (5.52)-(5.54) possuem uma componente normal que (de maneira geral) tende a contrair a snake, o que pode ser favorável para a snake externa (que precisa contrair) mas pode ser desfavorável para a interna (que precisa expandir).

Nos Capítulos 7 e 9 discutiremos estas questões em detalhes e as soluções encontradas.

Por fim, a inicialização do Dual-T-Snakes depende da interação com o usuário, principalmente para o posicionamento das T-Snakes internas. O método de inicialização do Capítulo 8 a seguir é uma possível solução para este problema que pode ser aplicada sob certas condições. Este método pode diminuir o custo computacional do Dual-T-Snakes, uma vez que a curva inicial obtida está em geral próxima do objeto de interesse.