Capítulo 4

Métodos Matemáticos

Neste capítulo discutimos conceitos matemáticos usados em modelos alternativos que procuram superar as limitações do modelo original apresentadas na seção 3.3. A apresentação dos modelos propriamente ditos é deixada para o próximo capítulo.

4.1 Programação Dinâmica

Em Amini at al. (Amini *et al.* , 1990) encontramos uma alternativa ao método variacional (Euler-Lagrange) baseada em programação dinâmica. Nesta seção vamos discutir os principais elementos desta proposta realçando suas vantagens e desvantagens em relação ao método variacional para a otimização da energia das snakes.

Os métodos variacionais vêm sendo usados para a solução de vários problemas em visão computacional, tais como cálculo do fluxo óptico e shape from shading (Black & Yuille, 1993). As limitações deste método no contexto destes problemas assim como no contexto das snakes, se originam no alto grau de diferenciabilidade necessária, gerando instabilidade numérica na presença de ruídos, e dificuldades para garantir se a solução encontrada é o ótimo global (Amini *et al.* , 1990).

A programação dinâmica, por outro lado, tem a vantagem de garantir a globalidade da solução encontrada sendo também numericamente mais estável que o método variacional. Contudo, a programação dinâmica tem limitações em função do montante de memória ($O\left(NM^2\right)$) necessário e da complexidade computacional ($O\left(NM^3\right)$), onde N é o número de snaxels e M é o número de pontos vizinhos a cada snaxel (gerando um *espaço de busca discreto* com NM pontos).

Neste capítulo iniciaremos com a apresentação dos princípios básicos da programação dinâmica no caso discreto (Dreyfus, 1965). O caso contínuo foi deixado para o Apêndice A, não sendo sua leitura pré-requisito para discussão que segue.

4.1.1 Programação Dinâmica no caso Discreto

Neste capítulo, formalizamos conceitos referentes a certos problemas de decisão do ponto de vista da Programação Dinâmica.

Os problemas a serem tratados têm a propriedade de que, a cada instante de uma seqüência finita $(t_1, t_2, ..., t_N)$, uma escolha deve ser feita dentre um conjunto finito de possibilidades . Formalmente, diremos que a cada instante t_i temos um conjunto finito de M possibilidades de escolha (*estados*) constituindo o *estágio* i. Uma vez que N e M são finitos, temos um espaço de estados discreto (*espaço de busca discreto*) com NM elementos.

Os problemas que podem ser colocados desta forma são denominados *problemas de decisão finitos em multi − estágio* e serão tratados nesta seção no contexto da programação dinâmica (Dreyfus, 1965).

Nosso ponto de partida será a aplicação do *Princípio de Optimalidade* que estabelece o seguinte: *O caminho ótimo entre dois pontos é também ótimo entre quaisquer dois outros pontos sobre o mesmo* .

Através deste princípio (Dreyfus, 1965) podemos obter uma expressão recursiva a qual é básica para a aplicação de programação dinâmica em snakes.

Se mantivermos a representação das bordas dos objetos de interesse por curvas contínuas como no modelo original de snakes (seção 3.1) então precisaríamos usar conceitos de programação dinâmica contínua. Infelizmente, esta formulação recai em equações diferenciais complicadas de difícil utilização na prática (ver Apêndice A ou (Dreyfus, 1965)).

Assim, a alternativa adotada por Amini at al. (Amini *et al.* , 1990) assim como nesta apresentação será supor que as bordas procuradas podem ser representadas por um modelo discreto, e assim formular as snakes no contexto discreto em lugar do contínuo usado no capítulo anterior. Vamos supor inicialmente que as curvas são abertas com extremos fixos.

Desta forma, representaremos as curvas de interesse através de um conjunto de pontos $\{v_i = (x_i, y_i), i = 0, ..., N+1\}$ ligados seqüencialmente, lembrando que v_0 e v_{N+1} são fixos (extremos da curva). Como no capítulo anterior, denotaremos o espaço formado por estas curvas por Ad (*espaço admissível*).

Com isto, o funcional de energia associado tomará (ver abaixo) a forma de uma função das $2N$ variáveis correspondentes aos pontos livres $\{v_i = (x_i, y_i), i = 1, ..., N\}$ e o problema de otimização correspondente torna-se um problema de otimização não-linear do tipo:

$$\underset{Ad}{min}\ E(v_1, ..., v_N), \tag{4.1}$$

onde Ad é espaço de curvas admissíveis definido acima.

Nos próximo capítulo veremos o modelo Dual e então explicitaremos o funcional de energia E em (4.11). Por hora, basta termos em mente que este funcional será constituído por uma energia interna e outra externa, podendo ser pensado como uma forma discreta daquele usado no modelo original (seção 3.1). Assim, vamos supor que E tem a forma:

$$E\left(v_1, ..., v_N\right) = E_{int}\left(v_1, ..., v_N\right) + E_{ext}\left(v_1, ..., v_N\right), \tag{4.2}$$

onde E_{int} e E_{ext} são dados, respectivamente, por:

$$E_{int}\left(v_1, ..., v_N\right) = \alpha \sum_{i=0}^{N-1} E_{int}^{i}\left(v_i, v_{i+1}, v_{i+2}\right), \tag{4.3}$$

$$E_{ext}\left(v_1, ..., v_N\right) = \beta \sum_{i=1}^{N} P\left(v_i\right), \tag{4.4}$$

onde $P\left(v\right) = -\left\|\nabla I\left(v\right)\right\|^2$ e α e β são parâmetros a serem estabelecidos a priori.

A expressão (4.3) significa que a energia interna associada a um snaxel v_i depende de seus vizinhos v_{i-1} e v_{i+1} além do próprio v_i.

A idéia central é formular o problema em (4.11) como um problema de decisão finito em multi-estágios. Para isto temos que definir um conjunto finito de estágios e a cada estágio associar um conjunto finito de possibilidades (escolhas). Cada escolha implica em um custo e a solução do problema é a seqüência de escolhas que ao final tem custo mínimo.

Lembrando que a função E em (4.11) depende dos N pontos livres da snake, é imediato definir o número de estágios igual a N. O número de possíveis escolhas para cada estágio vai depender de como discretizamos o domínio.

A Figura 4.1 a seguir representa um exemplo onde temos duas curvas limitando a região onde se encontra a borda procurada. A região entre as curvas é o domínio a ser considerado.

A Figura 4.2 mostra a maneira mais natural de subdividirmos este domínio aproveitando a semelhança geométrica entre a borda procurada e as curvas limites.

Nesta figura, estabelecemos uma correspondência entre os pontos das curvas limites e posteriormente discretizamos os segmentos que unem pontos correlatos em M partes. Se aumentarmos N ou M, melhoramos a precisão na localização da borda, mas aumentamos o custo computacional.

Uma vez que estamos considerando snakes abertas, precisamos escolher os pontos inicial e terminal da snake. O espaço de busca usado (Figura 4.2) impõe que estes

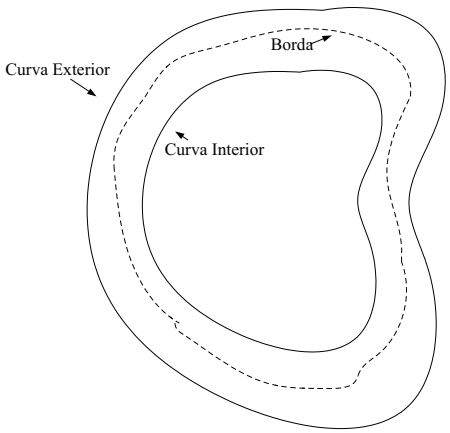

Figura 4.1: Curvas limitando região contendo borda.

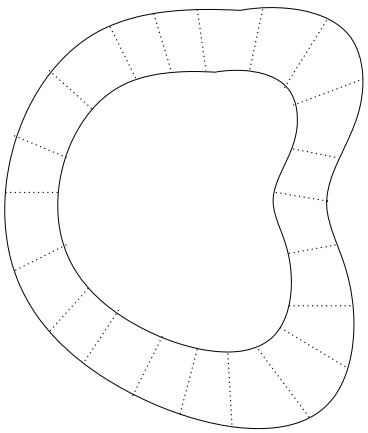

Figura 4.2: Discretização da região de interesse.

pontos pertençam a segmentos consecutivos. Como acima, denotaremos estes pontos por v_0 e v_{N+1}.

Retomando o funcional de energia em (4.2) e usando (4.3)-(4.4), vemos que a energia total da snake pode ser escrita na forma

$$E_{snake}(v) = E_0\left(v_0, v_1, v_2\right) + E_1\left(v_1, v_2, v_3\right) + ... + E_{N-1}\left(v_{N-1}, v_N, v_{N+1}\right), \tag{4.5}$$

onde:

$$E_i\left(v_i, v_{i+1}, v_{i+2}\right) = \alpha E_{int}^i\left(v_i, v_{i+1}, v_{i+2}\right) + \beta P\left(v_i\right), \tag{4.6}$$

e $v = (v_0, v_1, ..., v_N, v_{N+1})$.

Vamos definir uma função auxiliar denominada *função de valor ótimo*, dada por:

$$S\left(v_{i+2}, v_{i+1}\right) = \textit{energia da curva ótima conectando o ponto inicial } v_0 \textit{ ao ponto } v_{i+1}. \tag{4.7}$$

Pelo Principio de Optimalidade (Dreyfus, 1965; Amini *et al.* , 1990), podemos escrever $S\left(v_{i+2}, v_{i+1}\right)$ na forma recursiva:

$$S\left(v_{i+2}, v_{i+1}\right) = \min_{v_i} \left\{S\left(v_{i+1}, v_i\right) + E_i\left(v_i, v_{i+1}, v_{i+2}\right)\right\}. \tag{4.8}$$

Acrescentemos a esta definição a restrição:

$$S\left(v_1, v_0\right) = 0. \tag{4.9}$$

Posto desta forma, observemos que a solução do problema (4.11) é exatamente:

$$\min_{v_N} \left\{S\left(v_{N+1}, v_N\right)\right\}. \tag{4.10}$$

A expressão 4.8 é básica para a aplicação da programação dinâmica discreta em snakes. A seção 5.2 do próximo capítulo apresenta um modelo de snakes baseado nos conceitos acima descritos.

A apresentação de conceitos em programação dinâmica para o caso contínuo está feita no Apêndice A.

A seguir analisaremos questões topológicas.

4.2 Mudanças Topológicas

Neste trabalho denominamos *mudanças topológicas* ao fato de uma curva simples se dividir em duas ou mais componentes conexas (*split*) ou duas curvas simples se juntarem em uma única (*merge*).

Uma forma natural de formular mudanças topológicas seria supor que as curvas em questão são diferenciáveis e representá-las implicitamente; isto é, definí-las como um conjunto de nível de uma função $\Psi : \Re^2 \rightarrow \Re$ definida convenientemente.

Assim, mudanças topológicas ocorreriam naturalmente e suavemente, desde que função Ψ não perdesse sua suavidade. Este é o conceito fundamental das formulações contínuas e implícitas para mudanças topológicas (Malladi *et al.* , 1995).

O resultado destas propostas recai em conceitos de geometria e topologia diferenciais. O ponto de partida destas formulações é o Teorema da Função Implícita que estabelece que (Lima, 1985):

Teorema da Função Implícita: Seja $H : \Re^{N+K} \longrightarrow \Re^N$ uma aplicação suave tal que $0 \in range\,(H)$. Então:

$$M = \left\{ x \in \Re^{N+K} \mid H\,(x) = 0,\ x \text{ é um ponto regular de } H \right\},$$

é uma variedade diferenciável suave de dimensão K.

O método mais conhecido cuja base teórica parte deste teorema, deve-se a Malladi at al. (Malladi *et al.* , 1995) sendo comumente conhecido como Método dos Conjuntos de Níveis (*Level Sets*). Este método será descrito resumidamente no próximo capítulo.

Apesar das habilidades topológicas, o uso explícito de uma dimensão extra torna este método menos intuitivo que os modelos paramétricos, trazendo dificuldades para a incorporação de informações adicionais (de forma, por exemplo), ou vínculos definidos a priori ou criados pelo usuário durante a evolução do método.

Uma alternativa aos modelos implícitos é o método da T-Snakes. Este último é um modelo paramétrico capaz de lidar com mudanças topológicas tanto para 2D quanto 3D. Este método é fundamental para nosso trabalho e será apresentado em detalhes no próximo capítulo. A seguir, discutimos sua base teórica.

4.3 Fundamentos Matemáticos das T-Snakes

O método das T-Snakes (McInerney & Terzopoulos, 1995; McInerney, 1997) aborda as questões topológicas dentro do contexto da topologia combinatória a fim de obter uma formulação paramétrica com habilidades topológicas para as snakes.

Seus elementos básicos são uma decomposição simplicial (triangulação) do domínio,

um modelo discreto de snakes e uma *função característica*, definida nos nós da triangulação, que distingue o interior do exterior das snakes.

O par triangulação/função característica é a chave para a incorporação eficiente de mudanças topológicas em um modelo paramétrico de snakes; o que ficará mais claro após a apresentação que segue.

4.3.1 Decomposição do Domínio

Dado um subconjunto fechado $D \subset \Re^n$ existem dois métodos básicos para decomposição de domínio: não-simpliciais e simpliciais.

Os métodos não-simpliciais são baseados em decomposições celulares do espaço e não podem ser usados para representar superfícies ou contornos sem a utilização de métodos para retirar antigüidades. Caso típico de um método que utiliza este tipo de decomposição do espaço é o *Marching Cubes* (Lorensen & Cline, 1987).

Métodos usando decomposição simplicial do domínio não apresentam ambigüidades para a geração de aproximações poligonais para curvas e superfícies (Allgower & Georg, 1990).

Vejamos alguns conceitos fundamentais nesta área (Allgower & Georg, 1990).

Um conjunto de pontos $\{v_1, v_2, ..., v_{k+1}\} \subset \Re^n$ é dito "afim-independente" se as diferenças $v_2 - v_1, v_3 - v_1, ..., v_{k+1} - v_1$ formam um conjunto de vetores linearmente independente no $\Re^n$.

O fecho convexo $\left\{ v = \sum_{i=1}^{k+1} \alpha_i v_i |\ \alpha_i \geq 0, i = 1, ..., k+1 \text{ e } \sum_{i=1}^{k+1} \alpha_i = 1 \right\}$, com vértices dados por $k+1$ pontos afim-independentes $\{v_1, v_2, ..., v_{k+1}\}$ do $\Re^n$, é chamado um *k-simplex* e será denotado por $[v_1, v_2, ..., v_{k+1}]$.

Seja $\sigma = [v_1, v_2, ..., v_{k+1}]$ um *k-simplex* e seja $1 \leq l \leq k+1$. Se $\{w_1, w_2, ..., w_{l+1}\}$ é um subconjunto de vértices de σ, nós chamamos o $l - simplex\ \tau = \{w_1, w_2, ..., w_{l+1}\}$ uma $face\ l - dimensional$ de σ, ou simplesmente uma $l - face$. De particular interesse são:

(1) $0 - faces$ as quais correspondem a vértice de σ;

(2) $1 - faces\ as$ quais são também chamadas *arestas* de σ;

(3) $(k-1) - faces$ as quais são chamadas $faces$ de σ.

Seja Γ uma família não-vazia de $(n+1) - simplices$ em $\Re^{n+1}$. Nós chamamos Γ uma triangulação do $\Re^{n+1}$ se:

a.

$$\bigcup_{\sigma \in \Gamma} \sigma = \Re^{n+1}, \tag{4.11}$$

b. a intersecção $\sigma_1 \cap \sigma_2$ de dois simplices quaisquer $\sigma_1, \sigma_2 \in \Gamma$ é vazia ou uma face comum aos simplices;

c. a família Γ é localmente finita, isto é, qualquer subconjunto compacto de $\Re^{n+1}$ corta um número finito de simplices $\sigma \in \Gamma$.

Seja Γ uma triangulação do $\Re^{n+1}$, e seja $0 \leq k \leq n+1$. Nós denotamos por

$$\Gamma^k = \{\tau \mid \tau \text{ é uma } k - face \text{ de algum simplex } \sigma \in \Gamma\},$$

a família de todas as k-faces em Γ.

Para $\Re^2$ um 2-simplex é um triângulo. Para o $\Re^3$ um 3-simplice é um tetraedro. Seguindo a nomenclatura padrão, nós chamamos *nó* a um vértice de um simplex e à coleção de nós e arestas de Γ chamamos *malha* (identificada com Γ^1). A triangulação Γ é identificado com Γ^{n+1} ($\Gamma = \Gamma^{n+1}$).

Dois simplices $\sigma_1, \sigma_2 \in \Gamma$ são ditos adjacentes se eles têm uma face em comum.

Dado um simplex, o procedimento usado para obter os adjacentes a este é denominado *pivotiamento* e depende do tipo de triangulação usado. O lema a seguir prepara sua introdução.

Seja Γ uma triangulação do $\Re^{n+1}$. Considere um simplex $\sigma \in \Gamma$ e seja τ uma face de σ. Então, existe um único simplex $\tilde{\sigma} \in \Gamma$ tal que:

(1) $\tilde{\sigma} \neq \sigma$;

(2) τ uma face de $\tilde{\sigma}$;

Prova: Ver (Allgower & Georg, 1990).

Denotemos por $delv_i$ a operação de eliminação de um vértice v_i.

Seja $\sigma = [v_1, v_2, ..., v_{n+2}]$ um $(n+1)$-simplex de uma triangulação de $\Re^{n+1}$, e seja $\tau = [v_1, ..., delv_i ..., v_{n+2}]$ a face de σ oposta ao vértice v_i. Pelo lema anterior, deve existir um único nó $\tilde{v}_i$ diferente de v_i e tal que $\tilde{\sigma} = \left[v_1, ..., \tilde{v}_i ..., v_{n+2}\right] \in \Gamma$. A passagem de σ para $\tilde{\sigma}$ é chamada *pivotiamento*. Nós dizemos que o vértice v_i de σ é *pivotiado* no $\tilde{v}_i$, e que o simplex σ é *pivotiado* no simplex $\tilde{\sigma}$ através da face τ.

4.3.2 Aproximação Linear por Partes

Seja $H : \Re^{n+1} \longrightarrow \Re^n$ uma aplicação.

Dada uma triangulação Γ de $\Re^{n+1}$, nós desejamos aproximar as componentes de $H^{-1}(0)$ usando apenas os valores de H nos nós da triangulação. Não necessitamos de nenhuma suposição sobre suavidade ou continuidade sobre H, a menos que desejemos estudar erros de truncamento para a aproximação linear por partes a ser definida. Vejamos algumas definições:

Seja $H : \Re^{n+1} \longrightarrow \Re^n$ uma aplicação e seja Γ uma triangulação de $\Re^{n+1}$, e seja $\tau = [v_1, v_2, ..., v_{k+1}] \in \Gamma^k$ uma k-face de Γ onde $0 \leq k \leq n+1$.

a. Por $H_\tau : \tau \longrightarrow \Re^n$ denotamos a única aplicação afim que coincide com H nos vértices v_i de τ, isto é:

$$H_\tau(v) = \sum_{i=1}^{k+1} \alpha_i H(v_i), \text{ para } v = \sum_{i=1}^{k+1} \alpha_i v_i,$$

b. Seja $tng(\tau)$ (*espaço tangente de* τ) o espaço linear que contem todas as diferenças $w_1 - w_2$ com $w_1, w_2 \in \tau$. Então a Jacobiana de H é a aplicação linear $H'_\tau : tng(\tau) \longrightarrow \Re^n$ obtido por $H'_\tau(w_1 - w_2) = H_\tau(w_1) - H_\tau(w_2)$ para $w_1, w_2 \in \tau$. Uma vez que H_τ é afim, esta definição não é ambígua, isto é: $H_\tau(w_1) - H_\tau(w_2) = H_\tau\left(\tilde{w_1}\right) - H_\tau\left(\tilde{w_2}\right)$ para $w_1 - w_2 = \tilde{w_1} - \tilde{w_2}$.

c. Finalmente, a aproximação linear por partes de H é obtida pela união:

$$H_\Gamma = \bigcup_{\sigma \in \Gamma} H_\sigma,$$

isto é, $H_\Gamma(v) = H_\sigma(v)$ para todo $v \in \sigma$ e $\sigma \in \Gamma$. Não existe ambigüidade nesta definição uma vez que pode ser facilmente demonstrado que $H_{\sigma_1}(v) = H_{\sigma_2}(v)$ para $v \in \sigma_1 \cap \sigma_2$.

Um ponto $x \in \Re^{n+1}$ é chamado um ponto regular de H_Γ se e somente se:

(a) x não pertence a qualquer k-face $\tau \in \Gamma^k$ tal que $k < n$;

(b) H'_τ tem posto máximo para todo $\sigma \in \Gamma^n \cup \Gamma^{n+1}$ tal que $x \in \sigma$.

Um valor $y \in \Re^n$ é chamado um *valor regular* de H_Γ se todos os pontos de $H_\Gamma^{-1}(y)$ são regulares. Se um ponto não é regular será dito *singular*. Analogamente, se um valor não é regular ele será dito *singular*.

Perturbação: Seja o vetor:

$$\overrightarrow{\varepsilon} = \begin{pmatrix} \varepsilon^1 \\ . \\ . \\ . \\ \varepsilon^n \end{pmatrix}$$

onde $\varepsilon > 0$ e $\varepsilon^i > 0, i = 1, ..., n$, são potências de ε. Denominaremos este vetor um $\varepsilon - vetor$.

Proposição. Para qualquer subconjunto compacto $C \subset \Re^{n+1}$ existem no máximo um número finito de $\overrightarrow{\varepsilon}$ tal que $C \cap H_\Gamma^{-1}(\overrightarrow{\varepsilon})$ contém um ponto singular de H_Γ. Conseqüentemente, $\overrightarrow{\varepsilon}$ é um valor regular de H_Γ para quase todo $\overrightarrow{\varepsilon}$.

Prova: Ver (Allgower & Georg, 1990).

Nós chamamos um n-simplex $\tau \in \Gamma^n$ *completamente rotulado (completely labeled)* se e somente se ele contem soluções v_ε da equação $H_\Gamma(v) = \overrightarrow{\varepsilon}$ para qualquer $\varepsilon > 0$ suficientemente pequeno.

Um $(n+1) - simplex$ $\sigma \in \Gamma$ é chamado *traverse ou de borda* (com respeito a H) se ele contém uma n-face completamente rotulada.

(Princípio Door-In-Door-Out): Um $(n+1) - simplex$ tem nenhum ou exatamente duas n-faces completamente rotuladas.

Assim, podemos finalmente enunciar o seguinte algoritmo, com o qual é possível traçar qualquer componente conexa de $H_{\Gamma}^{-1}(0)$,

Algoritmo de Continuação (*Door-In-Door-Out*)

Inicialização

.$\sigma_0 \in \Gamma$ triângulo de borda (simplex inicial)

.τ_0 uma n-face completamente rotulada de σ_0;

Repetir para $n = 1, 2, ...$

.encontrar $\sigma_n \in \Gamma$, $\sigma_n \neq \sigma_{n-1}$ tal que $\tau_{n-1} = \sigma_n \cap \sigma_{n-1}$;

.encontrar a n-face completamente rotulada τ_n de σ_n tal que $\tau_n \neq \tau_{n-1}$;

.parar quando retornar ao triângulo inicial $\sigma_0 \in \Gamma$.

A inicialização do algoritmo depende da escolha do simplex inicial σ_0. No caso das snakes a escolha deste simplex é feita através dos pontos da própria snake. No próximo capítulo veremos isto em detalhes.

O algoritmo acima gera uma seqüencia

$$\sigma_0 \supset \tau_0 \subset \sigma_1 \supset \tau_1 \subset \sigma_2 \supset \tau_2 \subset \sigma_3 ... \tag{4.12}$$

de triângulos de borda, $(n+1) - simplices\ \sigma_i$, cada um destes contendo duas faces completamente rotuladas τ_{i-1}, τ_i. Uma vez que nenhum $(n+1) - simplices$ pode ter mais que duas n-faces completamente rotuladas, a seguinte classificação é facilmente verificada:

(a) Infinita: Ambas as seqüencias infinitas $\sigma_0, \sigma_1, ...$ e $\tau_0, \tau_1, ...$ geradas são tais que $\sigma_i \neq \sigma_{i+1}$ e $\tau_i \neq \tau_{i+1}, i = 0, 1, ...$

(b) Cíclica: Existe um inteiro finito $\widetilde{n} > 0$ tal que $\sigma_0 = \sigma_{\widetilde{n}}$ e $\tau_0 = \tau_{\widetilde{n}}$. A relação cíclica $\sigma_i = \sigma_{\widetilde{n}+i}$ e $\tau_i = \tau_{\widetilde{n}+i}$ vale para $i = 0, 1, ...$

Uma prova formal desta proposição é obtida via teoria de grafos (Allgower & Georg, 1990).

Uma vez que neste trabalho estamos interessados particularmente em snakes $2D$ a apresentação acima fornece os elementos necessários aos próximos capítulos.

Contudo, reservamos o Apêndice C para a generalização dos conceitos acima para variedades de dimensão qualquer, uma vez que estes serão necessários para a extensão do nosso trabalho para 3D.

4.4 Teoria de Invariância

Esta teoria é particularmente útil em situações em que, conhecidas as grandezas fundamentais $(x_1, x_2, ..., x_n)$ envolvidas em um problema, desejamos determinar uma quantidade $F(x_1, x_2, ..., x_n)$ que se mantenha inalterada (*invariante*) quando as grandezas fundamentais são transformadas.

No caso das snakes, por exemplo, é razoável que sua energia interna não mude quando aplicamos operações de translação ou rotação sobre a curva correspondente. Tal propriedade implica que a energia interna seja independente da posição da snake na imagem o que é uma propriedade geralmente assumida para termos de regularização (Snyder, 1991).

Em visão computacional, particularmente em problema envolvendo calibração da câmera, a utilidade do conceito de invariância aparece mais naturalmente (Gool *et al.* , 1995).

Em processamento de imagem, alguns resultados envolvendo invariância por transformações de escala estão na base das representações multi-escala de imagens (Florack *et al.* , 1992).

Nestes exemplos, a aplicação do conceito de invariância depende da definição precisa das variáveis $(x_1, x_2, ..., x_n)$ de interesse e a definição do "*grupo* " de transformações que vai agir sobre estas variáveis, ou mais precisamente, sobre o espaço que estas definem. Feito isto, temos condições de colocar a seguinte questão:

Existem grandezas (quantidades) $F = F(x_1, x_2, ..., x_n)$ *que não mudam seu valor quando aplicamos as transformações de interesse sobre as grandezas fundamentais* $(x_1, x_2, ..., x_n)$*?*

Esta pergunta nos leva diretamente ao problema de identificação de invariantes, o qual pode ser posto rigorosamente no contexto da teoria dos grupos de Lie (Olver, 1986).

De maneira mais precisa, a busca de invariantes está diretamente relacionada à derivada de Lie correspondente, a qual expressa a variação de uma função real ou de um campo (vetorial, por exemplo) sob a ação de um *grupo* de transformações (Olver, 1986).

Assim, o ponto de partida da teoria de invariância de Lie é um grupo de transformações que atuam em uma variedade diferenciável (ou em um espaço vetorial). A derivada de Lie correspondente fornece uma condição necessária e suficiente para que uma função com valores reais definida na variedade seja invariante pela ação do grupo de transformações. Fazendo uso de um sistema de coordenadas local, teremos um sistemas de equações diferenciais parciais, cujas soluções fornecerão os invariantes procurados (Olver, 1986).

Tal formalismo vem sendo estudado no contexto de problemas em visão computacional (Gool *et al.* , 1995) procurando-se estabelecer sua utilidade prática. Uma primeira dificuldade neste sentido, é que os problemas em visão computacional não se apresentam diretamente na forma de espaços e seus grupos de transformações, o que exige um estudo cuidadoso do problema para apresentá-lo de maneira apropriada para a aplicação da teoria.

No contexto das snakes, não encontramos referências da aplicação da teoria de

Lie na busca de novos modelos. Em geral as pesquisas têm sido conduzidas de maneira mais intuitiva e menos formal (Gunn & Nixon, 1997; Ip & Shen, 1998).

Contudo, uma vez que os conceitos presentes na idéia de invariância têm lugar comum na teoria de Lie, dedicamos o Apêndice B para sumarizar as idéias principais desta teoria no contexto de interesse. No final deste Apêndice apresentamos possíveis aplicações da teoria de Lie no contexto de snakes.

O leitor pode saltar a leitura do Apêndice B sem perda para a compreensão do texto que segue.