

TRABALHO DE GRADUAÇÃO

**RASTREAMENTO DO CONTORNO
DO VENTRÍCULO ESQUERDO
EM IMAGENS DE RESSONÂNCIA MAGNÉTICA
USANDO FILTRAGEM ESTOCÁSTICA**

Rogério de Almeida Richa

Brasília, dezembro de 2006

UNIVERSIDADE DE BRASÍLIA

FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASÍLIA
Faculdade de Tecnologia

TRABALHO DE GRADUAÇÃO

**RASTREAMENTO DO CONTORNO
DO VENTRÍCULO ESQUERDO
EM IMAGENS DE RESSONÂNCIA MAGNÉTICA
USANDO FILTRAGEM ESTOCÁSTICA**

Rogério de Almeida Richa

Relatório submetido como requisito parcial para obtenção
do grau de Engenheiro Eletricista

Banca Examinadora

Prof. Geovany Araújo Borges, UnB/ENE (Ori-entador) _____

Prof. Adson Ferreira da Rocha, UnB/ENE _____

Prof. Francisco Assis de Oliveira Nascimento, UnB/ENE _____

Agradecimentos

Agradeço inicialmente aos meus pais pelo apoio constante durante todo meu curso de graduação longe de casa e aos meus tios que me acolheram como um filho, me provendo todo suporte necessário. Gostaria de agradecer a dedicação, paciência do incansável professor Geovany Araújo Borges, pela base fundamental à minha formação como engenheiro. Agradeço também à todos os colegas do GPDS, que sempre colaboraram com meus trabalhos e em especial ao professor Francisco Assis que me permitiu a entrada no grupo. Finalmente, agradeço o carinho de todos amigos por todos momentos de alegria durante estes últimos anos.

Rogério de Almeida Richa

RESUMO

A segmentação e rastreamento do contorno do ventrículo esquerdo em imagens de ressonância magnética é um trabalho desafiador. O contorno do ventrículo esquerdo (VE) oferece dados importantes sobre o comportamento do coração, sendo útil para descoberta de disfunções, para o cálculo de parâmetros como fluxo sanguíneo que circula no coração etc. Para tornar o rastreamento do contorno do VE robusto à ruído, baixo contraste e outras complicações, filtros estocásticos são propostos para modelar as incertezas na localização do contorno real nas imagens. O algoritmo proposto neste trabalho realiza, em tempo real, o rastreamento em uma seqüência de imagens de IRM compreendendo diversos ciclos cardíacos. Seguindo uma abordagem baseada em contornos ativos, pouco conhecimento sobre a deformação das paredes do ventrículo é requerido para a realização do rastreamento. Finalmente, o método proposto foi avaliado usando uma seqüência de imagens cedida por pesquisadores da University of Southern of California (USC).

ABSTRACT

The segmentation and tracking of the left ventricle's contour in magnetic resonance images is a challenging task. Significant knowledge of the heart's dynamics can be acquired through the tracking of left ventricle (LV). Therefore, important parameters such as blood flow can be calculated and also heart diseases and disfunctions can be detected. In face of complications such as noise, low contrast and other image imperfections, stochastic filters are proposed to robustify the tracking, by dealing with the uncertainty related to the real contour's detection in every frame. This work proposes an algorithm for real time tracking of the left ventricle's contour over an image sequence depicting various heart cycles. Basing this algorithm on active contours, little knowledge of the heart wall's deformation is needed to perform the tracking. Finally, the proposed method is evaluated over an image sequence provided by researches from the University of Southern California (USC).

SUMÁRIO

1	INTRODUÇÃO	1
1.1	RASTREAMENTO DE CONTORNOS	1
1.1.1	APLICAÇÕES	1
1.2	IMAGEAMENTO POR RESSONÂNCIA MAGNÉTICA (IRM)	3
1.3	MOTIVAÇÃO	4
1.4	OBJETIVOS DO PROJETO	5
1.5	APRESENTAÇÃO DO TRABALHO	5
2	FUNDAMENTAÇÃO TEÓRICA	6
2.1	REVISÃO BIBLIOGRÁFICA	6
2.2	TRATAMENTO DAS IMAGENS DE IRM	7
2.3	B-SPLINES	8
2.4	MODELAMENTO DINÂMICO	10
2.4.1	EVOLUÇÃO TEMPORAL	11
2.4.2	MODELO DE OBSERVAÇÃO	13
2.5	FILTROS ESTOCÁSTICOS	14
2.5.1	FILTRO DE KALMAN	14
2.5.2	O FILTRO CONDENSATION	16
3	DESENVOLVIMENTO	19
3.1	INTRODUÇÃO	19
3.2	TÉCNICAS PARA EXTRAÇÃO DE BORDAS	19
3.3	DESLOCAMENTO NORMAL PARA AJUSTE DO CONTORNO	19
3.4	RESTRIÇÕES APLICADAS ÀS MEDIDAS	21
4	RESULTADOS EXPERIMENTAIS	24
4.1	INTRODUÇÃO	24
4.2	IMPLEMENTAÇÃO SEM FILTRAGEM ESTOCÁSTICA	25
4.3	IMPLEMENTAÇÃO BASEADA NO FILTRO DE KALMAN	25
4.4	IMPLEMENTAÇÃO BASEADA NO FILTRO CONDENSATION	26
4.5	ESFORÇO COMPUTACIONAL	26
4.6	AVALIAÇÃO DA METODOLOGIA PROPOSTA	29
5	CONCLUSÃO	36
5.1	TRABALHOS FUTUROS	36
	REFERÊNCIAS BIBLIOGRÁFICAS	37
	ANEXOS	39
I	A VARIANTE 'STAR'	40
II	DESCRIÇÃO DO CONTEÚDO DO CD	42
II.1	PASTAS DO CD	42

LISTA DE FIGURAS

1.1	A silhueta de um intruso pode guiar a câmara de monitoramento para melhor monitorar o ambiente (Fonte: [1]).	2
1.2	Informações sobre o tráfego em uma rodovia aumenta o tempo de resposta dos serviços de emergência ou informação (Fonte: [1]).	3
1.3	A dinâmica do coração pode ser reconstruída em tempo real por meio do rastreamento do contorno das cavidades cardíacas (Fonte: [1]).	3
1.4	O imageamento por ressonância magnética permite a obtenção de imagens com baixa razão sinal ruído, fornecendo uma boa representação das estruturas anatômicas.	4
1.5	Figura ilustrativa da visão frontal do coração fechado e aberto. O ventrículo esquerdo em corte é apresentado na figura à direita (Fonte: Wikipédia).	4
2.1	Filtragem de ruído para IRM.	7
2.2	Histograma de um quadro com pouco ruído (a) e histograma de um quadro com alto nível de ruído (b).	8
2.3	Correção dos níveis de cinza de uma imagem proposta para redução de ruído.	8
2.4	(a) Um simples <i>thresholding</i> promove a criação de grandes desníveis na imagem, futuramente prejudicando a extração de bordas. (b) Resultado do ajuste de contraste usando a transformação apresentada em 2.3.	9
2.5	Ilustração das bases de uma <i>B-Spline</i> : (a) As bases $B_n(s)$ são multiplicadas pelos seus coeficientes x_n e (b) combinadas, geram a função $x(s)$ (Fonte: [1]).	10
2.6	Exemplo de uma curva em duas dimensões $r(s)$ (a), definida pelos pontos de controle em (b) (Fonte: [1]).	11
2.7	O rastreamento dinâmico do contorno é realizado em duas fases: a predição do contorno baseado em um modelo de evolução da curva e o refino da posição predita baseado nas características das imagens (Adaptado de [2]).	12
2.8	Em (a) temos a evolução da curva $r(s)$ do instante $k-1$ à k . Como descrito na equação 2.8, a evolução de b^n esta associada ao deslocamento dos adjacentes em $n + 1$ e $n - 1$ (b).	12
2.9	O filtro de Kalman propagando as incertezas de um processo Gaussiano, assumindo uma dinâmica linear. As funções de densidade, para este caso, são completamente descritas média e variância (Adaptado de [1]).	15
2.10	Propagação das incertezas usando o filtro Condensation. Neste caso mais geral, as funções densidade não são mais Gaussianas como aquelas propagadas pelo filtro de Kalman (Adaptado de [1]).	16
2.11	O modelo de observação, ilustrado pela sua função densidade (Fonte: [2]).	18
2.12	Dinâmica do filtro Condensation, ilustrando o efeito do modelo de evolução sobre as partículas, a re-amostragem e finalmente a influência das medições no processo (Adaptado de [1]).	18
3.1	(a) Modelo unidimensional de uma borda; (b) A derivada de Gaussiana em uma dimensão: o valor de σ influência sobre a suavização do sinal e conseqüentemente sobre a intensidade do desnível necessário para que seja detectado; (c) Resultado da detecção da borda em (a) (Fonte: [3]).	20
3.2	Resultados da convolução com uma DoG para σ iguais à (a) 1,2 (b) 0,3 (c) 0,8 (d) 0,8 com ajuste de contraste.	21
3.3	O deslocamento normal entre a curva proposta e a curva detectada pela extração de bordas. Nota-se que o deslocamento real tem componentes na direção normal e tangencial do contorno estimado.	22

3.4	Uma linha intercepta os pixels de uma imagem de maneira irregular, gerando efeitos não desejados.....	22
3.5	O uso do deslocamento normal para a obtenção de medidas. Nota-se que as linhas normais podem interceptar mais de uma borda ou mesmo borda alguma.....	23
3.6	A direção do gradiente funciona como uma pré-seleção das medidas. Para cada medida, B pixels são avaliados na ordem indicada. O somatório dos primeiros $B/2$ pixels é comparado com o dos últimos e desta forma a direção do gradiente é determinada.	23
3.7	Procedimento da obtenção do vetor de pseudo-medicoes \mathbf{v}'_j	23
4.1	Fluxografo do algoritmo de rastreamento. Após a iniciação do filtro pelo usuário, toma lugar a seqüência predição/correção do modelamento dinâmico desenvolvido nos capítulos anteriores.	24
4.2	Rastreamento de contorno sem filtragem estocástica. Os círculos grandes marcam as bordas detectadas enquanto que os círculos pequenos marcam os pontos de controle (centro da linha de procura).	30
4.3	Rastreamento sem restrição de medidas usando filtragem de Kalman. Em (a) percebe-se que, pelos vários graus de liberdade do esquema proposto, os pontos de controle acabam por se chocar e o rastreamento do contorno é conseqüentemente perdido.....	31
4.4	Rastreamento com restrição de medidas usando filtragem de Kalman. Em (a) é visto quadros de um ciclo cardíaco. Se um ponto de controle não encontra medidas para sua correção, ele é marcado como um círculo cinza.	32
4.5	Rastreamento com restrição de medidas usando o filtro Condensation.....	33
4.6	Rastreamento usando <i>Star</i> . Esta variante usa o filtro de Kalman e sua formulação encontra-se no Anexo.	34
4.7	O conjunto de pontos de controle selecionados manualmente para iniciação da curva. Nota-se que a distância entre pontos não é constante devido à imprecisão na iniciação.....	35
4.8	O contorno ideal do ventrículo esquerdo (em verde). As imperfeições na reconstrução das imagens são aspectos importantes ao rastreamento.	35
4.9	A diferença entre o contorno ideal (verde) e o estimado pelo rastreamento (amarelo). É notável a influência no rastreamento das imperfeições da representação do ventrículo.	35
I.1	O contorno do VE sendo representado por um elipsóide, cujos parâmetros são indicados na tabela I.1.	40
I.2	Representação dos parâmetros do elipsóide de ajuste. Nota-se que as coordenadas dos eixos espaciais estão invertidas em relação ao eixo das abcissas.	41

LISTA DE TABELAS

4.1	Parâmetros do algoritmo de rastreamento usado nos experimentos.	25
4.2	Tempo de execução de cada passo do rastreamento (figura 4.1), usando o filtro de Kalman. Onze pontos de controle são usados para representar a curva.	27
4.3	Tempo de execução de cada passo do rastreamento (figura 4.1), usando o filtro Condensation. Onze pontos de controle são usados para representar a curva.	27
4.4	Repetição da medida do tempo de execução de cada passo do rastreamento, usando o filtro de Kalman, para sete pontos de controle são usados para representar a curva.	27
4.5	Repetição da medida do tempo de execução de cada passo do rastreamento, usando o filtro Condensation, para sete pontos de controle são usados para representar a curva.	28
4.6	Repetição da medida do tempo de execução de cada passo do rastreamento, usando o filtro Condensation, para uma redução de 700 partículas (novo $N_p = 400$) e sete pontos de controle.....	28
I.1	Valores dos parâmetros normalizados do elipsóide que assemelha ao VE.....	41

LISTA DE SIMBOLOS

Índices

k	índice para tempo discreto
d	ordem do polinômio da <i>B-Spline</i>
s	parâmetro espacial da curva paramétrica
N	numero de pontos de controle
N_p	numero de partículas
E	extensão da linha normal de procura

Matrizes e Vetores

$\mathbf{0}$	matriz de zeros
$\mathbf{1}$	matriz onde todos valores são 1
\mathbf{A}	matriz de transformação do processo dinâmico em tempo discreto
$\mathbf{B}(s)$	vetor de funções base da <i>B-Spline</i>
\mathbf{H}	matriz de observação do filtro de Kalman
\mathbf{I}_k	matriz identidade $k \times k$
\mathbf{K}	Matriz de zeros
$\mathbf{n}(s)$	vetor normal da curva no ponto s
\mathbf{b}_n	pontos de controle para a curva <i>B-Spline</i>
\mathbf{Q}	vetor de pontos de controle
\mathbf{Q}^x	vetor de pontos de controle na coordenada x
\mathbf{Q}^y	vetor de pontos de controle na coordenada y
\mathbf{Q}	vetor de pontos de controle
$\mathbf{r}(s)$	curva paramétrica em duas dimensões
\mathbf{w}_k	vetor de ruído de medição, gaussiano e independente
\mathbf{r}_k	vetor de ruído de processo, gaussiano e independente
$\Delta \mathbf{x}$	diferença do vetor \mathbf{x} entre os instantes k e $k-1$
\mathbf{z}_k	vetor de observação do instante k
χ^k	historia do vetor de estados até o instante k
\mathbf{Z}	história do vetor de observações até o instante k
s_l	l -ésima partícula
π_l	peso da l -ésima partícula
\mathbf{S}	conjunto de partículas
\mathbf{W}	conjunto de pesos das partículas
\mathbf{C}_k	conjunto de medidas no instante k
\mathbf{Q}	matriz de covariância da incerteza do processo
\mathbf{R}	matriz de covariância da incerteza do ruído da medição
\mathbf{P}	matriz de covariância da incerteza das variáveis de estado
\mathbf{v}^n	vetor de medidas do n -ésimo ponto de controle
\mathbf{m}^n	ponto médio entre os pontos de controle em $n+1$ e $n-1$
\mathbf{p}_j	pseudo-medida relativa à j -ésima medida do vetor v_j

Operações

$\mathbf{x} \cdot \mathbf{y}$	Produto escalar dos vetores \mathbf{x} e \mathbf{y}
$\mathbf{A} \otimes \mathbf{B}$	Produto de Kronecker das matrizes \mathbf{A} e \mathbf{B}
$E(x)$	expectância da variável x

Sobrescritos

\hat{x}_k	Estimativa da quantidade x no instante k
$\hat{x}_{k k-1}$	Predição da quantidade x no instante $k-1$
\mathbf{A}^T	Transposta da matriz \mathbf{A}
\mathbf{A}^{-1}	Inversa de \mathbf{A}

Siglas

$N(\mu, \sigma^2)$	Distribuição normal com média μ e variância σ^2
VE	Ventrículo Esquerdo
IRM	Imageamento por ressonância magnética
CONDENSATION	<i>Conditional Density Propagation</i>
FDP	Função Densidade de Probabilidade
FDA	Função Densidade Acumulada de Probabilidade

1 INTRODUÇÃO

Os avanços das últimas décadas no campo da visão computacional possibilitam o desenvolvimento de ferramentas automáticas de diagnóstico médico, reforçadas por novas formas de imageamento como o imageamento por ressonância magnética.

1.1 RASTREAMENTO DE CONTORNOS

Nas décadas de 70 e 80 do século passado, os interesses em visão computacional estavam voltados basicamente ao desenvolvimento de algoritmos computacionais "multi-propósito", que realizavam a interpretação das imagens baseada em técnicas de processamento em baixo nível, isto é, a extração de contornos etc. [3, 4, 5]. Tais algoritmos são bastante eficientes mas no entanto apresentam soluções limitadas.

Com a evolução e amadurecimento das técnicas de rastreamento, conhecimento anterior sobre as estruturas interpretadas pelo computador começa a ser utilizado, mostrando diversos resultados convincentes [1]. Grande parte deste conhecimento anterior resume-se ao conhecimento geométrico das estruturas. Isto leva conseqüentemente a especialização dos algoritmos de rastreamento.

O conhecimento da classe geométrica dos objetos ou estruturas rastreadas aumenta drasticamente a robustez do rastreamento à perturbações do ambiente onde se encontra o objeto e das interferências da forma de imageamento utilizada. Isto pode ser interpretado como a restrição da procura, pela máquina, por formas já conhecidas do objeto rastreado. Uma das primeiras técnicas desenvolvidas que faz uso deste conhecimento anterior são as *snakes*, ou serpentes [6]. Trata-se basicamente de um contorno elástico que é casado com o objeto detectado na imagem. Restrições elásticas à deformação do contorno limitam a gama de deformações, tornando a segmentação robusta a ruídos e defeitos na representação do objeto na imagem.

Outra técnica conhecida, que também é aplicada à problemas de rastreamento é a "teoria de padrões", que trata conhecimento anterior em um contexto probabilístico, onde a forma detectada dos objetos rastreados é resultado de distorções aplicadas em uma forma base, ou molde. Neste caso, o problema não se resume a descobrir uma distribuição de probabilidade para as formas detectadas. É necessário também fusionar o conhecimento anterior com as observações feitas à partir de uma seqüência de imagens. Ferramentas usadas para realizar o fusão tratam-se dos filtros estocásticos.

Filtros estocásticos são conhecidos e amplamente usados em problemas de rastreamento [7, 8, 9, 2]. Eles serão apresentados posteriormente, quando serão modificados e usados na tarefa de fusão de dados. O desenvolvimento de técnicas de rastreamento e detecção encontra aplicabilidade em uma vasta variedade de problemas. Algumas delas são mostradas a seguir.

1.1.1 Aplicações

A ambição das primeiras décadas, de desenvolver ferramentas de visão de propósito geral foi gradualmente dando lugar à abordagens mais especializadas, como o que se verificou nas áreas de robótica, imageamento médico etc. Aliado ao desenvolvimento de "hardware" mais poderosos, o cenário da visão computacional se modificou bastante. O desenvolvimento das técnicas de rastreamento não só resolveu os problemas que inicialmente motivaram sua pesquisa, mas também criou diversas aplicações nas mais vastas áreas, como por exemplo, tecnologias de processamento de vídeo em geral.



Figura 1.1: A silhueta de um intruso pode guiar a câmera de monitoramento para melhor monitorar o ambiente (Fonte: [1]).

Monitoramento de segurança Em aplicações de monitoramento de segurança, um possível intruso é detectado e a câmera é dinamicamente reposicionada para uma melhor captura das atividades em determinado ambiente (ver figura 1.1).

Monitoramento de tráfego Em estradas, informações sobre acidentes, engarrafamentos ou mesmo leitura de placas podem ser obtidas automaticamente usando algoritmos de rastreamento (ver figura 1.2). Isto implica em aplicações imediatas, como o acionamento dos serviços de emergência mais rápido, a disponibilização imediata de informações sobre tráfego, entre outras.

Rastreamento em imageamento médico Diagnósticos médicos auxiliados por técnicas de análise de seqüências de imagens em tempo real (ver figura 1.3) aumentam a segurança e velocidade na obtenção do diagnóstico [1].

Uma modalidade bastante atraente é o imageamento por ressonância magnética (IRM), que apesar de ser cara, agrega as vantagens do ultra-som e do raio-X. Ela é discutida em seguida.

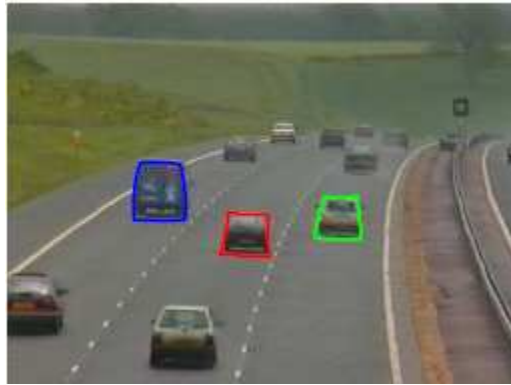


Figura 1.2: Informações sobre o tráfego em uma rodovia aumenta o tempo de resposta dos serviços de emergência ou informação (Fonte: [1]).

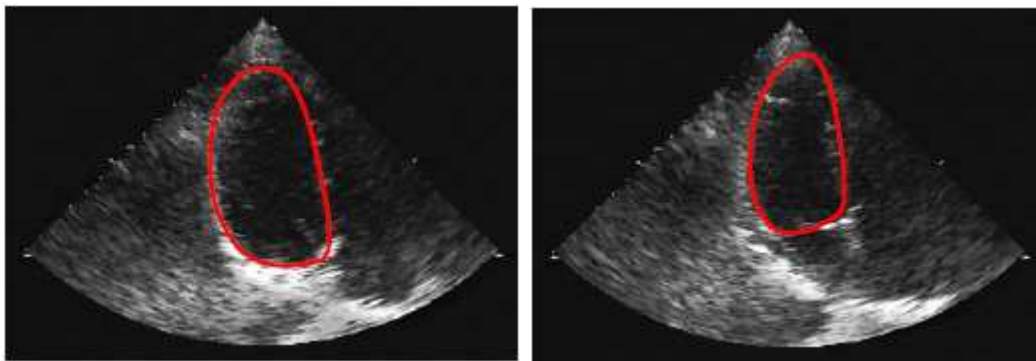


Figura 1.3: A dinâmica do coração pode ser reconstruída em tempo real por meio do rastreamento do contorno das cavidades cardíacas (Fonte: [1]).

1.2 IMAGEAMENTO POR RESSONÂNCIA MAGNÉTICA (IRM)

Atualmente, o imageamento por ressonância magnética (IRM) encontra-se como uma das ferramentas mais poderosas para imageamento. Este tipo de imageamento possui uma série de vantagens sobre outras técnicas, desde que é possível a visualização de estruturas anatômicas com alta relação sinal-ruído sem necessidade de exposição à radiação.

Como para cada paciente um grande volume de dados é gerado, técnicas de processamento e interpretação de seqüências de IRM automáticas são procuradas de modo a facilitar o diagnóstico em tempo real. Mais especificamente, a detecção de diversas doenças cardíacas pode ser realizada pela reconstrução e análise das funções cardíacas por meio de IRM [10].

O desenvolvimento de um método automático de reconstrução das funções cardíacas é um trabalho bastante laborioso. Mesmo que o IRM permita uma alta razão sinal-ruído em comparação com técnicas não radiantes como ultrasonografia, complicações surgem por distúrbios no fluxo sanguíneo, respiração e outras funções cardíacas naturais. Adicionando a presença de ruído e artefatos na reconstrução da imagem, a estrutura cardíaca não é representada perfeitamente.

O rastreamento robusto do contorno de estruturas como o ventrículo esquerdo (VE) em uma seqüência longa de imagens ($\approx 200-300$ quadros) representa um grande desafio (ver figura 1.4). Como disfunções cardíacas podem ser detectadas pelo contorno do VE, tal rastreamento se torna interessante na geração de diagnósticos.

Uma solução para tornar o rastreamento robusto às perturbações mencionadas acima é o modelamento estocástico. Outra meta deve ser a elaboração de um rastreamento que seja efetuado em tempo real, ou

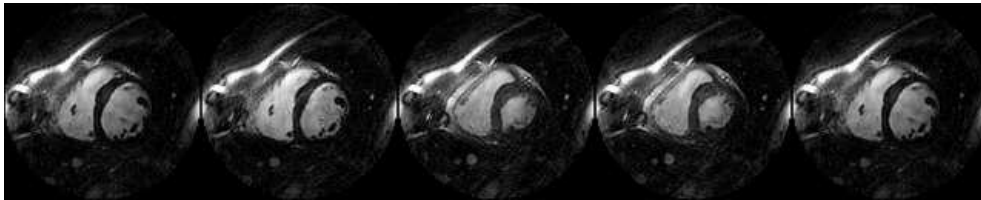


Figura 1.4: O imageamento por ressonância magnética permite a obtenção de imagens com baixa razão sinal ruído, fornecendo uma boa representação das estruturas anatômicas.

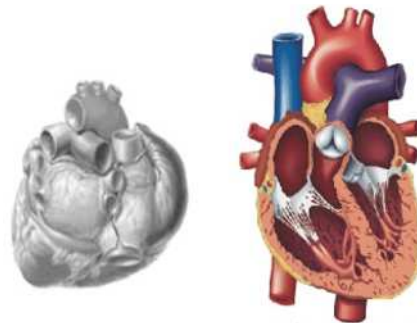


Figura 1.5: Figura ilustrativa da visão frontal do coração fechado e aberto. O ventrículo esquerdo em corte é apresentado na figura à direita (Fonte: Wikipédia).

próximo disto, de modo a permitir a obtenção de um diagnóstico preciso *em linha*, economizando tempo de utilização da máquina e do paciente.

1.3 MOTIVAÇÃO

O coração humano é um órgão cavitário, constituindo-se de uma bomba muscular cuja principal função é o impulsionamento do sangue para os vasos sanguíneos e daí para os órgãos e tecidos do organismo (figura 1.5). Este órgão é formado por quatro cavidades: átrio direito, átrio esquerdo, ventrículo direito e ventrículo esquerdo, sendo elas divididas pelo septo cardíaco.

O ventrículo esquerdo é a mais forte das cavidades do coração. Sua musculatura é a mais espessa já que trabalha para vencer a alta pressão da circulação sistêmica. Existem inúmeras patologias que acometem o ventrículo esquerdo e o coração como um todo. Os exames disponíveis atualmente falham em nos fornecer informações precisas sobre a anatomia e da dinâmica desse órgão. Por meio da visualização do contorno e dos batimentos em tempo real do coração, pode-se observar mais objetivamente a espessura da parede do ventrículo esquerdo, bem como mensurar o fluxo sanguíneo que circula na cavidade, detectando alterações advindas de doenças como a hipertensão arterial sistêmica (que causa pronunciado espessamento da parede dessa cavidade).

Pode-se ainda, através da medida do fluxo sanguíneo que passa pelo ventrículo esquerdo, diagnosticar condições que aumentem ou diminuam o mesmo, relacionadas a hipertensão pulmonar (doença essa causada pelo enfisema pulmonar por exemplo, que diminui o fluxo sanguíneo no pulmão e conseqüentemente no lado esquerdo do coração), doenças valvulares como insuficiência/estenose mitral, insuficiência/estenose aórticas, já que nesses casos há importante alteração no fluxo sanguíneo.

Além disso, a detecção de áreas infartadas na parede desse ventrículo é possível observando-se a sincronia dos batimentos e da contração da parede cardíaca em tempo real, facilitando assim o diagnóstico do infarto agudo do miocárdio.

1.4 OBJETIVOS DO PROJETO

O trabalho aqui realizado trata do rastreamento em tempo real do contorno do ventrículo esquerdo em imagens de ressonância magnética. Este trabalho explora a filtragem estocástica no contexto de rastreamento. A filtragem estocástica tem o propósito de modelar a incerteza na localização do verdadeiro contorno do ventrículo durante o rastreamento, de modo a tornar este rastreamento robusto à falhas e ambigüidades na detecção do contorno.

Mínimas restrições à deformação do contorno do ventrículo são supostas, fazendo com que o algoritmo de rastreamento seja facilmente adaptável à outras aplicações que envolvem formas geométricas pouco definidas.

Ao final, uma seqüência de teste avaliará o desempenho dos algoritmos propostos nos quesitos principais de robustez e tempo de processamento, visto que o rastreamento devera ser efetuado em tempo real, ou próximo disto.

1.5 APRESENTAÇÃO DO TRABALHO

Este trabalho é dividido em três partes principais. Inicialmente, uma revisão bibliográfica sobre o rastreamento de contornos é feita e então são formuladas as técnicas que serão usadas para o rastreamento. Posteriormente, as ferramentas usadas para detecção de contornos serão apresentadas e uma série de experimentos avaliará o algoritmo de rastreamento proposto. Por fim uma conclusão sobre o trabalho é apresentada, assim como as perspectivas de trabalho futuro.

2 FUNDAMENTAÇÃO TEÓRICA

Algumas técnicas de rastreamento de contornos revistas e a formulação do problema.

2.1 REVISÃO BIBLIOGRÁFICA

O rastreamento do contorno do VE em imagens de ressonância magnética não é um assunto novo na literatura. O processamento e interpretação das imagens de IRM do coração auxilia o diagnóstico de diversas disfunções e doenças cardíacas [10]. Mais especificamente, o grande número de trabalhos sobre a reconstrução do movimento do VE deve-se a precisão na visualização desta estrutura por meio do IRM. As técnicas de extração e representação do contorno do VE têm por fim gerar um conjunto de dados que um médico ou profissional da saúde possa interpretar [11].

Como mencionado na introdução, a análise cardíaca automática usando IRM é ainda um trabalho desafiador, devido à natural variabilidade das estruturas cardíacas entre pacientes ou à interferência da atividade natural de outros órgãos. Além disso, a presença de ruído e artefatos diminui a confiança às informações contidas nas imagens. Desta forma, somente a interpretação em "baixo nível" das imagens não apresenta bons resultados.

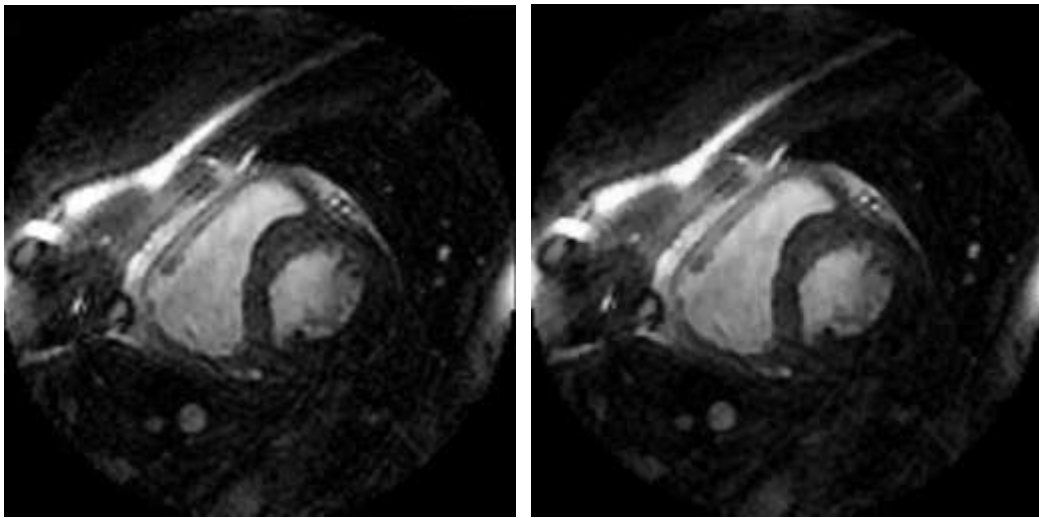
Técnicas que incorporam conhecimento anterior da dinâmica cardíaca esbarram na definição de um modelo que descreva uma estrutura orgânica. Estruturas orgânicas possuem características diferentes de paciente a paciente e um modelo descritivo deve tolerar tais variações.

Em diversos casos, modelos deformáveis foram desenvolvidos para tarefas de rastreamento [12, 13, 14, 15]. A técnica baseada em *snakes*, que fora previamente apresentada, tem amplo uso [16] e mostra bom desempenho na segmentação de estruturas cardíacas. Diversos modelos de elasticidade já desenvolvidos permitem boas representações geométricas de estruturas como a do VE [10, 16]. Para estimação do movimento do contorno do VE, técnicas baseadas na informação de contraste, textura ou mesmo fluxo ótico são utilizadas [17]. Apesar dos bons resultados, poucas são passíveis de serem usadas em uma aplicação em tempo real, devido ao grande esforço computacional envolvido.

Modelos ativos, que usam ferramentas estatísticas para rastreamento são técnicas que apresentam resultados promissores [18]. Os bons resultados obtidos usando tais modelos se devem à uma fase de treinamento: um conjunto de segmentações é obtida por treinamento, que incorpora o conhecimento da deformação espaço-temporal do objeto para o rastreamento. Da mesma maneira que esta fase de treinamento permite a obtenção de bons resultados, a maior deficiência desta técnica se deve exatamente à dependência deste treinamento. Além de ser uma tarefa laboriosa, quaisquer erros na fase de treinamento incapacitam o rastreamento.

O trabalho aqui proposto segue uma abordagem similar à [1], com modificações ao modelo dinâmico de deformação. O contorno rastreado é descrito por uma *B-Spline*, possuindo mínimas restrições de deformação. No entanto, restrições são propostas às medidas extraídas de cada quadro, com intuito de produzir um rastreamento estável. Nas próximas seções uma revisão sobre *B-Spline* é apresentada, juntamente com o modelo dinâmico proposto para a evolução do contorno do VE. Por fim, apresentaremos como os filtros estocásticos trabalham para aumentar a robustez do rastreamento.

Em paralelo ao trabalho do rastreamento, um trabalho para a diminuição do nível de ruído nas imagens de IRM é realizado, com intuito de auxiliar a tarefa de rastreamento. Diversos trabalhos com tal fim existem na literatura, sendo que novamente esbarramos nas restrições da execução em tempo real.



(a) Quadro original extraído da seqüência

(b) Quadro após aplicação do filtro Malik-Perona

Figura 2.1: Filtragem de ruído para IRM.

2.2 TRATAMENTO DAS IMAGENS DE IRM

Um dos primeiros tarefas com objetivo de melhorar a qualidade das imagens de IRM foi o estudo e implementação de filtros redutores de ruído. Para este objetivo, a difusão anisotrópica foi uma das técnicas que resultou na melhoria mais significativa das imagens.

A difusão anisotrópica é um algoritmo simples. É usada em tarefas primarias de visão, suavizando a imagem enquanto mantendo a definição dos contornos. Em imagens nas quais o gradiente do brilho devido ao ruído é menor que a das bordas verdadeiras e em que o nível de ruído não tem grande variação pela extensão da imagem, a difusão anisotrópica apresenta bons resultados. Bons resultados são conhecidos para a difusão anisotrópica na redução de ruído em imagens de ultrasonografia.

Uma característica das imagens de IRM é a variação temporal do nível de ruído, conforme ilustrado na figura 2.2. As condições descritas para um bom desempenho na redução de ruído por meio da difusão anisotrópica são atendidas, mas não para todos quadros da seqüência. Como isto significa esforço computacional feito sem necessidade, seu uso é eliminado por restrições de tempo de processamento. A figura 2.1 comprova que a difusão anisotrópica diminui notavelmente os níveis de ruído nas imagens de IRM.

Mostrando a necessidade da avaliação do nível de ruído temporalmente, uma análise histográfica é realizada. Da figura 2.2 percebemos a diferença entre os histogramas de um quadro corrompido ou não por ruído é bastante pequena. Como o ruído não se mostra explicitamente no histograma da imagem, partimos para outra abordagem.

Outra proposta é o ajuste dinâmico do contraste das imagens de IRM, de modo a tornar menos significativa a influência do ruído na detecção das bordas na imagem. O ajuste de contraste é uma técnica simples usada para acentuar os contornos das imagens. Uma transformação modifica os valores dos níveis de cinza da imagem original u por meio de uma transformação linear $T()$ (determinada na figura 2.3):

$$v = T(u)$$

Da figura 2.3, o valor do limiar L pode ser descoberto por meio da função densidade de probabilidade acumulada FDA, que por sua vez pode ser encontrado por meio do cálculo do histograma de cada imagem. O limiar L é o valor de nível de cinza para o qual a FDA seja menor que 0,9 (ou seja, que 90% dos pixels tenham nível de cinza menores que o limiar L). Isto foi determinado empiricamente, após aplicar a transformação mencionada para diversos valores de L . Um ajuste de contraste é usado ao lugar de um

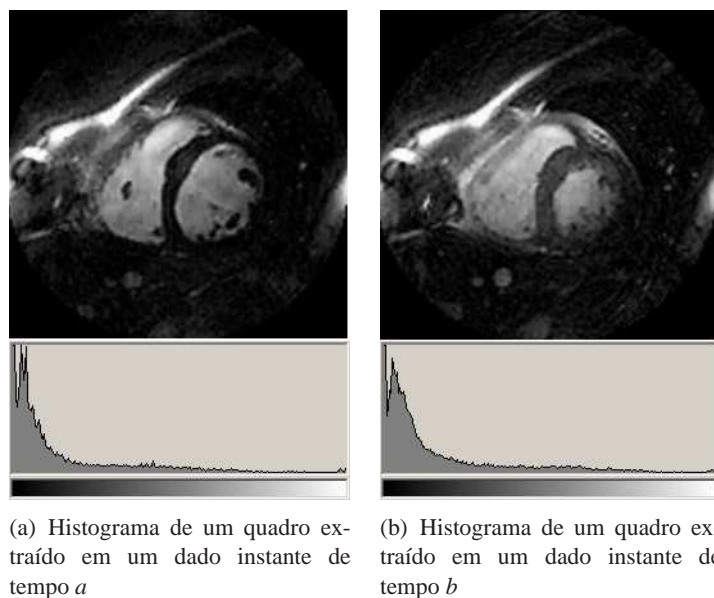


Figura 2.2: Histograma de um quadro com pouco ruído (a) e histograma de um quadro com alto nível de ruído (b).

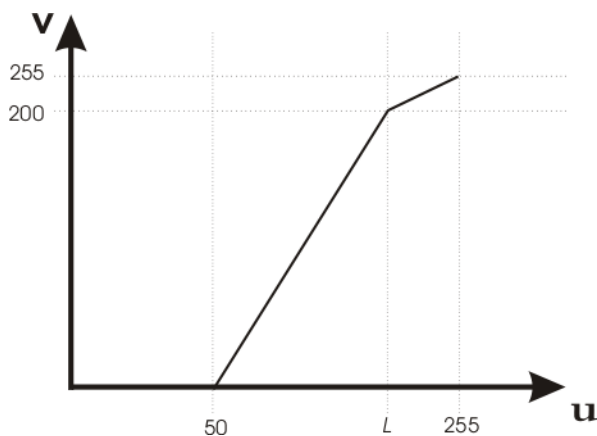


Figura 2.3: Correção dos níveis de cinza de uma imagem proposta para redução de ruído.

simples *threshold* pelo simples fato que este último geraria grandes desníveis no nível de cinza na imagem, futuramente prejudicando a extração de bordas. Os resultados da aplicação do ajuste de contraste são apresentados na figura 2.4.

2.3 B-SPLINES

Uma vasta teoria sobre *Splines* é encontrada na literatura [19]. Esta modalidade de representação de curvas ou superfícies é bastante conveniente, dada a estabilidade numérica em seu cálculo, seus baixos requerimentos de grau polinomial para um mesmo desempenho de ajuste. Vários algoritmos estão disponíveis para sua implementação atualmente, tornando-a bastante atraente.

Pelo fato do contorno ser completamente caracterizado um conjunto de pontos de controle, esta maneira de representar o contorno do VE produzido pelo rastreamento é bastante interessante. Uma curva $x(s)$ é construída como uma soma ponderada de N funções base $B_n(s)$ (desta forma, '*B*'-*Splines*), onde $n = 0, \dots, N - 1$. Da forma mais simples, cada função base consiste de d polinômios definidos em uma unidade

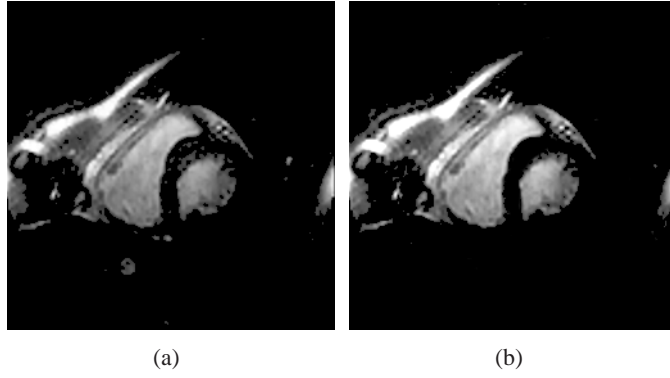


Figura 2.4: (a) Um simples *thresholding* promove a criação de grandes desníveis na imagem, futuramente prejudicando a extração de bordas. (b) Resultado do ajuste de contraste usando a transformação apresentada em 2.3.

de espaço no eixo s . Cada unidade de espaço é ligada por nós, o mais suave possível, tendo $s-2$ derivadas no ponto. A *B-Spline* tem portanto a seguinte forma:

$$x(s) = \sum_{n=0}^{N-1} x_n B_n(s) \quad (2.1)$$

em que x_n são os pesos aplicados as respectivas bases $B_n(s)$ (ver figura 2.5).

Por convenção, as funções base $B_n(s)$ somam para 1 em todos os pontos:

$$\sum_{n=0}^{N-1} B_n(s) = 1 \quad (2.2)$$

Na forma matricial, podemos expressar esta curva compactamente da forma abaixo:

$$x(s) = \mathbf{B}(s)^T \mathbf{Q}^x \quad (2.3)$$

em que $\mathbf{B}(s)$ é o vetor de funções base e \mathbf{Q}^x representa o vetor dos respectivos pesos x_n :

$$\mathbf{B}(s) = (B_0(s), B_1(s), \dots, B_{N-1}(s)), \quad \mathbf{Q}^x = \begin{pmatrix} x_0 \\ \vdots \\ x_{N-1} \end{pmatrix} \quad (2.4)$$

A correspondência entre as equações 2.3 e 2.1 é notada. Para representar uma curva em um plano, a curva é descrita parametricamente da forma (ver ilustração na figura 2.6):

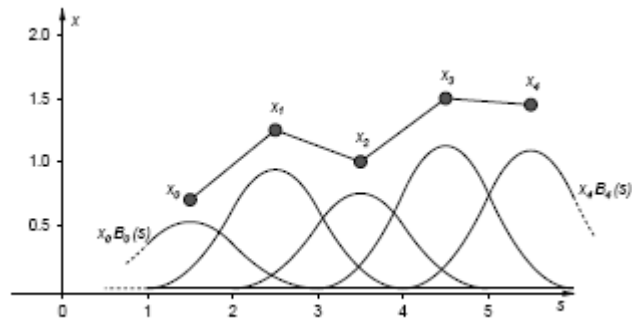
$$\mathbf{r}(s) = (x(s), y(s)) = U(s)\mathbf{Q} \quad (2.5)$$

em que

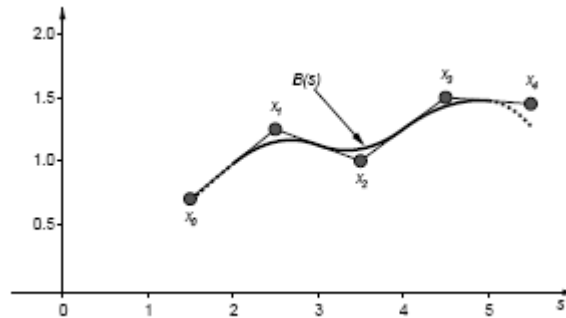
$$\mathbf{Q} = \begin{pmatrix} \mathbf{Q}^x \\ \mathbf{Q}^y \end{pmatrix} \quad (2.6)$$

e $U(s)$ é o delta de Kronecker de

$$U(s) = I_2 \otimes \mathbf{B}(s)^T = \begin{pmatrix} \mathbf{B}(s)^T & \mathbf{0} \\ \mathbf{0} & \mathbf{B}(s)^T \end{pmatrix} \quad (2.7)$$



(a)



(b)

Figura 2.5: Ilustração das bases de uma *B-Spline*: (a) As bases $B_n(s)$ são multiplicadas pelos seus coeficientes x_n e (b) combinadas, geram a função $x(s)$ (Fonte: [1]).

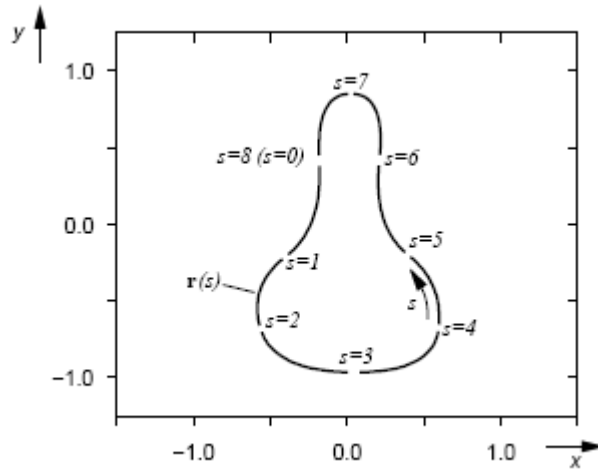
Como o contorno de interesse é um contorno fechado, uma *B-Spline* pode ser tornada "fechada" ou periódica repetindo os últimos dois pontos de controle. Deste modo, uma curva periódica e suave (possuindo a primeira derivada) é obtida. Definida a representação da curva rastreada, o próximo passo é definir o modelo dinâmico de evolução da mesma.

2.4 MODELAGEM DINÂMICA

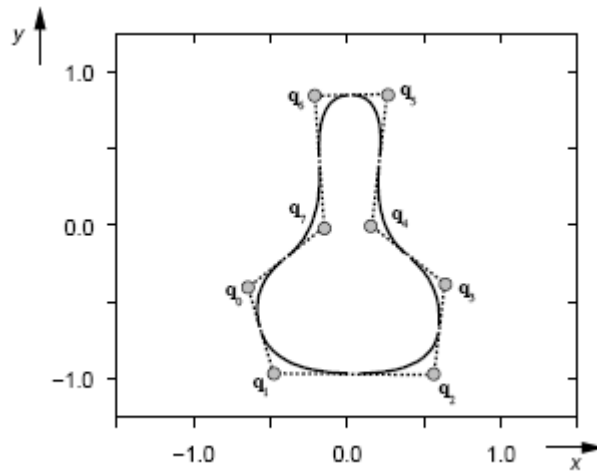
A proposta deste trabalho não é desenvolver um modelo deformável do movimento do ventrículo, pois tal modelo custaria bastante tempo para ser corretamente desenvolvido. De modo semelhante, uma *snake* não apresenta o grau de conhecimento anterior cujo acréscimo ao rastreamento era de interesse, nem um custo computacional aceitável para uma aplicação em tempo real.

A maneira mais poderosa de realizar o rastreamento seria combinar o modelamento dinâmico semelhante à uma *snake* com as formas geométricas de um modelo deformável. Este é o caminho ideal do modelamento dinâmico de um contorno para o rastreamento. Neste trabalho, o problema de rastreamento é abordado de uma maneira diferente: a dinâmica do contorno é elaborada com o mínimo de restrições à sua deformação, sendo que posteriormente restrições são aplicadas às medições com propósito de estabilizar a curva rastreada. Tal abordagem permite visa um rastreamento robusto, em tempo real, que seja capaz de funcionar durante seqüências longas.

O modelo dinâmico de rastreamento é elaborado como um processo de duas fases, no qual um modelo de evolução temporal prediz, em tempo discreto, o movimento do contorno de um instante para outro e posteriormente a posição predita é corrigida usando medições de características da imagem (ver figura



(a)



(b)

Figura 2.6: Exemplo de uma curva em duas dimensões $r(s)$ (a), definida pelos pontos de controle em (b) (Fonte: [1]).

2.7). Este é um paradigma bastante popular de rastreamento, sendo por esta razão que os filtros estocásticos são usados como mecanismo probabilístico de fusão destas duas fases.

As fases do modelo dinâmico, a evolução temporal da curva no tempo e o modelo de observação do contorno real são detalhados a seguir.

2.4.1 Evolução temporal

Como explicado anteriormente, o contorno rastreado em determinado instante k é completamente descrito, por meio de uma *B-Spline*, por N pontos de controle $\mathbf{b}^n = (x^n, y^n)$, tal que $n = \{1, \dots, N\}$.

$$\mathbf{b}_k^n = (x_k^n, y_k^n) = \begin{pmatrix} x_{k-1}^n + \alpha \Delta x_k^n + \beta (\Delta x_k^{n-1} + \Delta x_k^{n+1}) \\ y_{k-1}^n + \alpha \Delta y_k^n + \beta (\Delta y_k^{n-1} + \Delta y_k^{n+1}) \end{pmatrix}^T \quad (2.8)$$

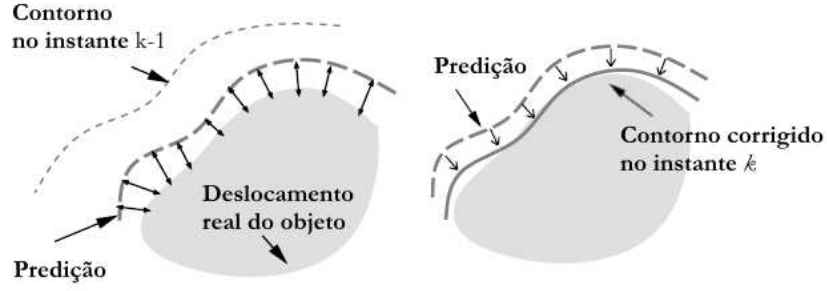


Figura 2.7: O rastreamento dinâmico do contorno é realizado em duas fases: a predição do contorno baseado em um modelo de evolução da curva e o refinamento da posição predita baseado nas características das imagens (Adaptado de [2]).

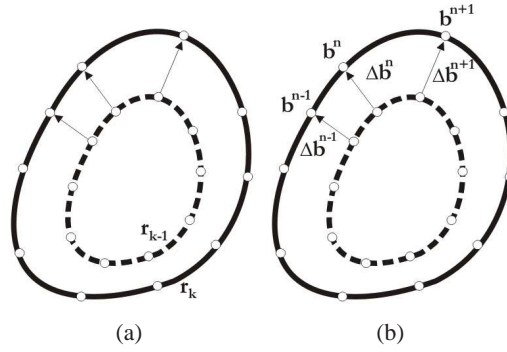


Figura 2.8: Em (a) temos a evolução da curva $r(s)$ do instante $k-1$ à k . Como descrito na equação 2.8, a evolução de b^n está associada ao deslocamento dos adjacentes em $n+1$ e $n-1$ (b).

em que $\Delta b^n = (\Delta x^n, \Delta y^n)$ são deslocamentos nas coordenadas x e y (calculado entre os quadros dos instantes k e $k-1$) e α e β são constantes reguladoras.

Como cada ponto de controle define o comportamento de sua respectiva região no coração, segue naturalmente que seu respectivo comportamento dinâmico tem ligação com seus pontos vizinhos. Esta correlação entre o movimento dos pontos de controle é descrita pelo modelo de evolução na equação 2.8 e ilustrada na figura 2.8.

O contorno, em um determinado instante k , é representado completamente por meio do vetor de estados \mathbf{x}_k , composto pelas coordenadas dos N pontos de controle $\mathbf{b}_k = (x_k, y_k)$ que definem a curva *B-Spline*, assim como os respectivos deslocamentos espaciais. A transição entre estados $k-1$ e k é linear e descrita matricialmente pela equação 2.14:

$$\mathbf{x}_k = (\xi^1 \quad \xi^2 \quad \xi^3 \quad \dots \quad \xi^N)^T \quad (2.9)$$

em que

$$\xi = (x \quad y \quad \Delta x \quad \Delta y) \quad (2.10)$$

A fase de predição discutida anteriormente baseia-se na formulação matricial de 2.8. Nele, as variáveis de estado \mathbf{x}_k são, no instante $k-1$, preditas para o instante k por meio do modelo de evolução proposto. A equação que governa este processo é:

$$\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} + \mathbf{r}_k \quad (2.11)$$

em que \mathbf{r}_k é um vetor de ruído branco, com distribuição $N(\mathbf{0}, \mathbf{Q})$ e \mathbf{A} é a matriz de transição do modelo

discreto de evolução dos parâmetros da curva estimada. O ruído branco serve para modelar a incerteza associada à predição.

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & \alpha & 0 & 0 & 0 & \beta & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & \beta & 0 \\ 0 & 1 & 0 & \alpha & 0 & 0 & 0 & \beta & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \beta \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \beta & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 1 & 0 & \alpha & 0 \\ 0 & 0 & 0 & \beta & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 1 & 0 & \alpha \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.12)$$

Desenvolvido o modelo de evolução do rastreamento, o próximo passo é definir o modelo de observação.

2.4.2 Modelo de observação

Um modelo de observação é a etapa que atualiza o contorno rastreado usando as informações colhidas do contorno real de cada novo quadro. O processo de detecção a cada quadro é uma etapa que trata de todas as dificuldades inerentes à forma de imageamento em questão, como ruído, artefatos entre outros efeitos, que podem mascarar o real contorno do VE. O objetivo é estimar o conjunto de pontos de controle que melhor descreva o contorno real, no passo atual k , levando em conta as incertezas associadas ao processo.

O detalhamento das técnicas para detecção do contorno é dado no próximo capítulo, quando desenvolvemos as ferramentas para executarmos o rastreamento. Pelo instante, assumimos que seja possível extrair um conjunto de medidas \mathbf{C} do contorno real, que atualize os pontos de controle estimados pelo modelo de evolução, para cada novo quadro da seqüência.

As medidas do conjunto \mathbf{C} são então processadas por cada filtro estocástico. O modelo de observação que corrige as predições é descrito da seguinte forma:

$$\mathbf{z}_k = \mathbf{H}\mathbf{x}_k + \mathbf{w}_k \quad (2.13)$$

em que \mathbf{z}_k representa o vetor de observações (um subconjunto de \mathbf{C}), \mathbf{w}_k é o vetor contendo ruído branco com distribuição $N(0, \mathbf{R})$. A matriz \mathbf{H} é a matriz de observação, que seleciona quais posições do vetor de estados são observadas.

Infelizmente, a ausência de restrições à deformação da curva não garante que os pontos de controle não se colidem ou se interlaceem devido à ruído ou perturbações nas imagens. Como proposta para evitar tais problemas, restrições ao conjunto de medidas são usadas. Podemos gerar um conjunto de *pseudo-medições* (projeções das medições reais) que restringem o deslocamento da curva e estabilizam o rastreamento. No capítulo seguinte a técnica é propriamente detalhada. As *pseudo-medições* são portanto usadas, ao lugar das medidas reais, para corrigir as posições estimadas pelo modelo de evolução.

Faz-se necessário, portanto, o estudo de um filtro estocástico que melhor fusione as observações e a estimativa do modelo de evolução, para um dado instante k .

2.5 FILTROS ESTOCÁSTICOS

De modo a trabalhar com as incertezas das observações colhidas a cada instante k e as fusionar com as previsões feitas pelo modelo de evolução, filtros estocásticos são usados para estimar, de maneira ótima, o contorno real do VE.

Devido ao seu uso na teoria de controle e rastreamento de alvos, o filtro de Kalman é consagrado na literatura [5]. As suas diversas vantagens como fácil implementação e reduzido esforço computacional, tornam-no amplamente usado em diversos problemas de visão computacional [20]. Seu desempenho aplicado para a solução de nosso problema é avaliado. No entanto, o filtro de Kalman não consegue lidar com as múltiplas medidas que surgem naturalmente, ao detectar o contorno na presença de ruído. Tal limitação afeta o desempenho do rastreamento ao lidar com eventuais ambigüidades.

Soluções sub-ótimas podem ser encontradas por meio do uso de filtros de partículas. Estes filtros são métodos de Monte Carlo sequenciais baseados em "partículas", isto é, representações discretas da função densidade de probabilidade. Com o intuito de lidar com não-linearidades que surgem no processo de rastreamento, o filtro Condensation é usado. Este filtro, proposto por Blake e Isard [1, 2], é uma variante do filtro de partículas SIR (*Sample Importance Resampling filter*) conhecido na literatura [9].

Em seguida, o problema é reescrito baseado nos dois filtros apresentados.

2.5.1 Filtro de Kalman

O filtro de Kalman como estimador linear é um caso especial, aplicado apenas à processos que podem ser modelados com densidades Gaussianas. Ele funciona baseado em dois passos: previsão e correção.

No primeiro passo, o modelo de evolução dos estados estima, no presente instante $k-1$, qual serão as variáveis de estado no próximo instante k . Isto pode ser pensado como uma difusão, na qual a incerteza das variáveis de estado \mathbf{x}_k se espalha sob a ação do modelo de evolução descrito anteriormente. Esta difusão é puramente linear para o caso Gaussiano e aumenta a incerteza do estado. Em seguida, o passo de correção realiza a incorporação das medidas obtidas no instante k , comparando-as com os valores preditos no instante anterior $k-1$. O efeito das observações \mathbf{z}_t é aumentar a probabilidade na vizinhança das medições. A figura 2.9 ilustra os passos descritos.

Deste modo, os pontos cruciais para o correto funcionamento do filtro de Kalman são a correta regulação das incertezas relacionadas ao processo de previsão e ao processo de medição, assim como a correta definição do processo de evolução do estado. Em seguida, é descrita a iniciação do filtro de Kalman, assim como os passos de previsão e correção.

Para a iniciação do filtro, é necessária a definição manual de um conjunto de pontos de controle para o contorno do primeiro quadro da seqüência, caracterizando o rastreamento como *semi-automático*. Esta iniciação manual é a única intervenção manual necessária para o funcionamento do algoritmo de rastreamento. Por meio de experimentação, os valores das incertezas mencionadas anteriormente são calculados e então armazenados nas matrizes de covariância \mathbf{Q} , \mathbf{R} e \mathbf{P}_0 , que correspondem respectivamente à incerteza do processo, a incerteza das medições e a incerteza inicial das variáveis de estado.

Desta forma, a previsão das variáveis de estado $\hat{\mathbf{x}}_k$ no instante $k-1$ pelo filtro de Kalman é baseada na equação do modelo de evolução 2.11:

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{A}\hat{\mathbf{x}}_{k-1} \quad (2.14)$$

Conseqüentemente, a incerteza do processo $\hat{\mathbf{P}}_k$ também passa pela difusão linear:

$$\hat{\mathbf{P}}_{k|k-1} = \mathbf{A}\hat{\mathbf{P}}_{k-1}\mathbf{A}^T + \mathbf{Q} \quad (2.15)$$

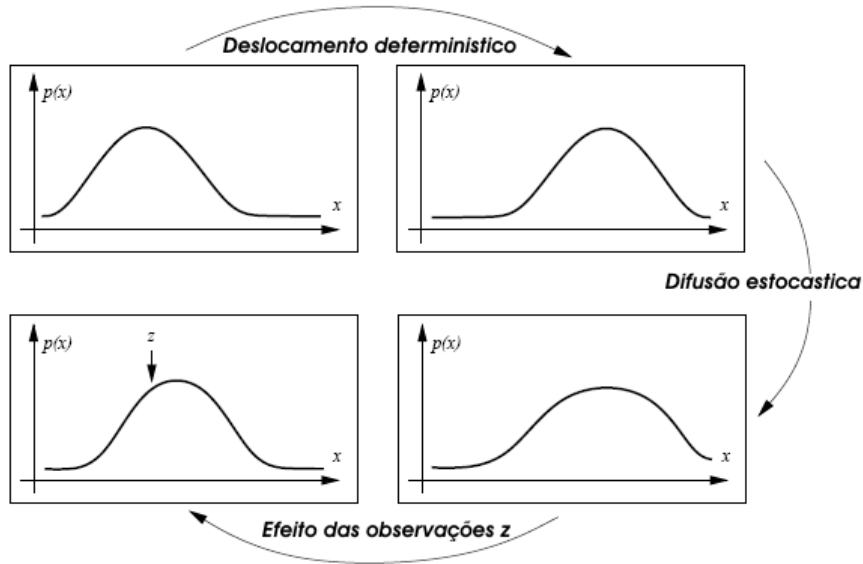


Figura 2.9: O filtro de Kalman propagando as incertezas de um processo Gaussiano, assumindo uma dinâmica linear. As funções de densidade, para este caso, são completamente descritas média e variância (Adaptado de [1]).

Finalmente, o passo de correção toma lugar. Como dito anteriormente, o objetivo é estimar o conjunto de pontos de controle que melhor descreva o contorno no passo atual k . Isto se faz corrigindo a estimativa $\hat{\mathbf{x}}_k$ baseada na detecção do contorno real no instante k . Nota-se que o filtro de Kalman não trabalha com múltiplas hipóteses para a posição do contorno detectado.

Para o filtro de Kalman, a matriz de observação \mathbf{H} é uma matriz $N_o \times N$ conforme ilustrada na equação 2.16 (N_o significa o número de medidas disponíveis em determinado instante):

$$\mathbf{H} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix} \quad (2.16)$$

A correção da estimativa $\hat{\mathbf{x}}_k$ baseada nas observações \mathbf{z}_k resulta então no estado \mathbf{x}_k conforme abaixo

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}\hat{\mathbf{x}}_{k|k-1}) \quad (2.17)$$

$$\mathbf{K}_k = \hat{\mathbf{P}}_{k|k-1}\mathbf{H}^T(\mathbf{H}\hat{\mathbf{P}}_{k|k-1}\mathbf{H}^T + \mathbf{R})^{-1} \quad (2.18)$$

A matriz de incerteza de estado \mathbf{P}_k é calculada por

$$\hat{\mathbf{P}}_k = (\mathbf{I} - \mathbf{K}_k\mathbf{H})\hat{\mathbf{P}}_{k|k-1} \quad (2.19)$$

Nota-se que os deslocamentos Δx e Δy são estimados pelo filtro de Kalman.

Como mencionado anteriormente, o vetor de observação \mathbf{z}_k , usado pelo filtro estocástico para correção do vetor de estados estimados é um subconjunto do conjunto de medidas \mathbf{C} . Como mais de uma medida pode estar disponível para a atualização de certo ponto de controle $\mathbf{b}^{(n)}$ e o filtro de Kalman não trata múltiplas hipóteses de medição, dentre as M medidas disponíveis, aquela mais próxima do valor predito $\hat{\mathbf{b}}^{(n)}$ é eleita para atualizar o ponto. Deste modo é suposto que o contorno realizou um deslocamento mínimo entre quadros. O vetor \mathbf{z}_k é então preenchido e a curva atualizada.

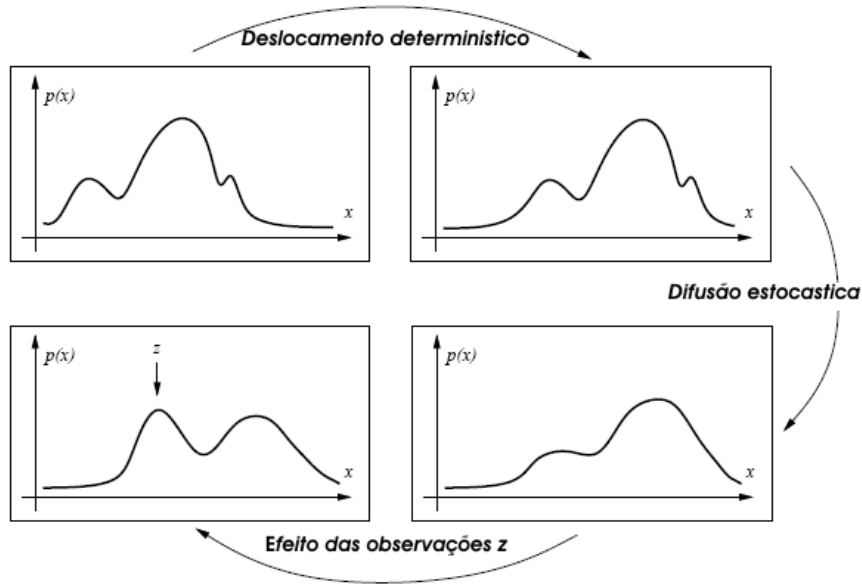


Figura 2.10: Propagação das incertezas usando o filtro Condensation. Neste caso mais geral, as funções densidade não são mais Gaussianas como aquelas propagadas pelo filtro de Kalman (Adaptado de [1]).

2.5.2 O filtro Condensation

Como já mencionado, o filtro de Kalman faz a suposição que a incerteza relacionada ao estado é sempre Gaussiana. No entanto, quando o rastreamento é realizado em um ambiente ruidoso, as observações realizadas tendem a tornar a distribuição das variáveis de estado não-Gaussianas. Neste contexto, o filtro Condensation é introduzido para atender à esta hipótese mais realista do problema de rastreamento. Seu algoritmo é consideravelmente mais simples que o do filtro de Kalman (que exige a solução das equações de Riccati) e, apesar de envolver um passo de re-amostragem, ele pode funcionar em tempo real.

Nenhuma suposição é feita sobre a forma da função densidade de probabilidade do estado \mathbf{x}_k . Sendo $\chi_k = \mathbf{x}_1, \dots, \mathbf{x}_k$ toda a história do vetor \mathbf{x}_k , o processo é modelado como uma seqüência temporal de Markov (da mesma forma que o filtro de Kalman):

$$p(\mathbf{x}_k | \chi_{k-1}) = p(\mathbf{x}_k | \mathbf{x}_{k-1}) \quad (2.20)$$

isto é, o novo estado é condicionado diretamente ao estado passado, independente da história anterior.

2.5.2.1 A realização de observações

Seja \mathbf{z}_k as observações no instante k e $Z_k = \mathbf{z}_1, \dots, \mathbf{z}_k$ o conjunto de medições até o instante k , assumimos que elas são independentes tanto ao processo dinâmico quanto à elas mesmas:

$$p(Z_k | \chi_k) = \prod_{i=1}^k p(\mathbf{z}_i | x_i) \quad (2.21)$$

O processo de observação é expresso probabilisticamente:

$$p(Z_{k-1}, \mathbf{x}_k | \chi_{k-1}) = p(\mathbf{x}_k | \chi_{k-1}) \prod_{i=1}^{k-1} p(\mathbf{z}_i | \mathbf{x}_i) \quad (2.22)$$

A forma da função $p(\mathbf{z} | \mathbf{x})$, ilustrada na figura 2.11, consiste na sobreposição de Gaussianas, com média centrada em cada medida z e variância r^2 . Nota-se na figura a equiprobabilidade das medições \mathbf{z}_k . O

parâmetro r^2 representa a variância da distribuição associada à cada medida \mathbf{z}_k . Outro detalhe importante é um valor residual de probabilidade, que é aplicado à cada medida. Este valor permite o surgimento do efeito de "tunelamento", ou seja, a probabilidade de um contorno na posição \mathbf{z} que esteve presente em instantes anteriores e não é detectada em determinado quadro não desaparece completamente. Isto aumenta a capacidade de recuperação de eventuais perdas de rastreamento.

2.5.2.2 Propagação do processo

Dado que a função densidade do processo no instante k seja descrita por uma cadeia de Markov, a regra de propagação do estado é da forma:

$$p(\mathbf{x}_k|Z_k) = k_t p(\mathbf{z}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{k-1}|Z_{k-1}) \quad (2.23)$$

A equação 2.23 é simplesmente a regra de Bayes, inferindo a função densidade a partir das observações. O termo $p(\mathbf{x}_k|Z_{k-1})$ representa a predição da distribuição anterior $p(\mathbf{x}_{k-1}|Z_{k-1})$. A multiplicação pela densidade da observação $p(\mathbf{z}_k|\mathbf{x}_k)$ corrige a distribuição predita usando as medições z_k . Como as observações não são Gaussianas, a função $p(\mathbf{x}_k|Z_k)$ torna-se não-Gaussiana também. O problema então se resume à como aplicar esta filtragem não linear de maneira computacionalmente eficiente. A proposta de usar amostras, ou "partículas" é descrita a seguir.

2.5.2.3 Filtragem por re-amostragem

A filtragem baseada em partículas é feita gerando randomicamente variações do estado, chamadas partículas, de modo que a distribuição destas partículas aproxima a densidade $p(\mathbf{x}_{k-1}|\mathbf{z}_{k-1})$. Estas partículas compõe um conjunto de amostras $\mathbf{S}_{k-1} = \{\mathbf{s}_{k-1}^{(1)}, \dots, \mathbf{s}_{k-1}^{(N_p)}\}$, onde N_p representa o numero de partículas. Outro conjunto $\mathbf{W}_{k-1} = \{\pi_{k-1}^{(1)}, \dots, \pi_{k-1}^{(N_p)}\}$ contem os pesos π^i de cada i -ésima partícula de modo que:

$$\pi^i = \frac{p_z(\mathbf{s}^{(i)})}{\sum_{j=1}^N p_z(\mathbf{s}^{(j)})} \quad (2.24)$$

em que $p_z(\mathbf{x})$ é a função densidade de observação. Conforme N_p cresce, melhor a distribuição $p(\mathbf{x}_{k-1}|\mathbf{z}_{k-1})$ é aproximada. O valor do vetor de estados \mathbf{x}_{k-1} pode ser expresso como a média de $\mathbf{s}_{k-1}^{(i)}$, ponderada pelos pesos $\pi_{k-1}^{(i)}$:

$$\mathbf{x}_{k-1} \approx \frac{\sum_{n=1}^N \mathbf{s}^{(n)} p_z(\mathbf{s}^{(n)})}{\sum_{n=1}^N p_z(\mathbf{s}^{(n)})} \quad (2.25)$$

O processo iterativo aplicado aos conjuntos de partículas é ilustrado na figura 2.12. Nesta figura, no topo encontra-se o conjunto $\{(\mathbf{s}_{k-1}^{(n)}, \pi_{k-1}^{(n)}), n = 1, \dots, N_p\}$ de partículas e função densidade referente ao passo $k-1$. A primeira operação é a amostragem (com substituição) das partículas N_p vezes, baseado em seus pesos. Nota-se que partículas com pesos grandes podem ser escolhidas várias vezes. Outras podem nunca serem escolhidas. Inicialmente, as partículas passam pelo processo de difusão, conforme ilustrado na figura 2.10, onde cada elemento é sujeito ao passo preditivo ditado pela equação 2.11 (no caso de um modelo de evolução linear). As amostras então se separam pois o vetor erro \mathbf{r}_k é decorrelacionado e cada partícula sofre sua própria difusão. Nesta fase a correção usando medidas é feita, gerando os pesos $\pi_k^{(n)}$ por meio da densidade de observação. Isto resulta no conjunto $\{(\mathbf{s}_k^{(n)}, \pi_k^{(n)})\}$, que ilustra o estado no instante k . A equação 2.25 é usada para estimar as variáveis de estado \mathbf{x}_k .

O pseudo-código do algoritmo pode ser encontrado em [2].

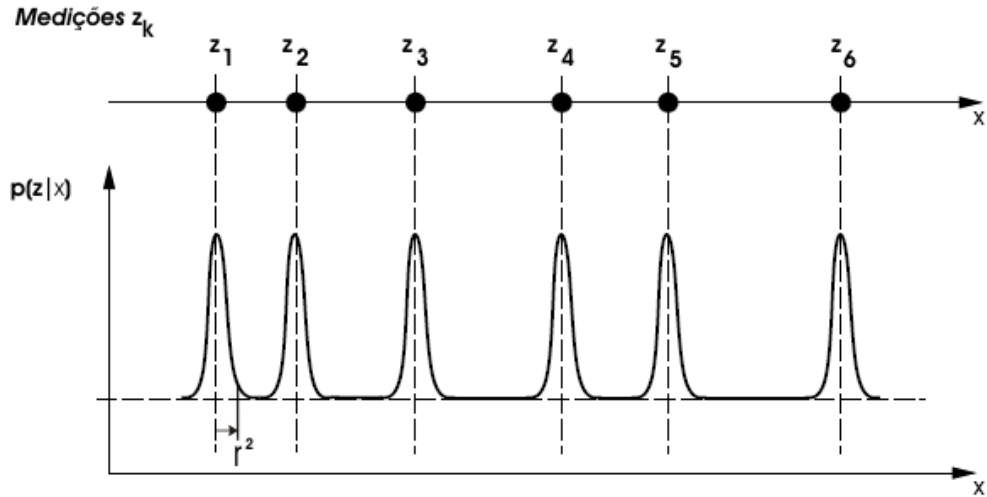


Figura 2.11: O modelo de observação, ilustrado pela sua função densidade (Fonte: [2]).

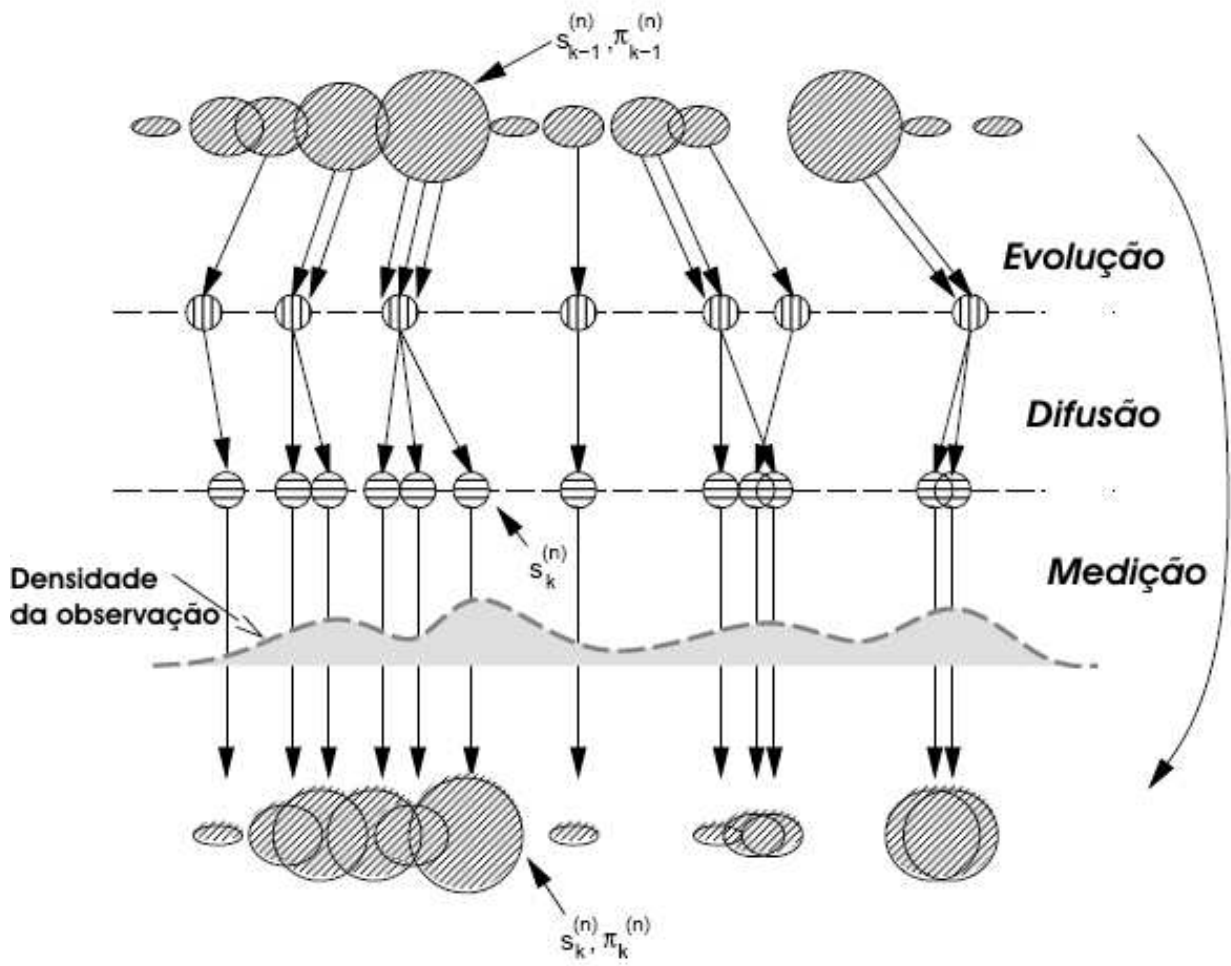


Figura 2.12: Dinâmica do filtro Condensation, ilustrando o efeito do modelo de evolução sobre as partículas, a re-amostragem e finalmente a influência das medições no processo (Adaptado de [1]).

3 DESENVOLVIMENTO

3.1 INTRODUÇÃO

Nos capítulos anteriores, a formulação do problema de rastreamento foi dada e as técnicas que serão utilizadas, discutidas. Neste capítulo as técnicas de detecção do contorno nas imagens serão desenvolvidas e as restrições que serão aplicadas às medidas serão formuladas.

3.2 TÉCNICAS PARA EXTRAÇÃO DE BORDAS

A cada quadro da seqüência de IRM o contorno real deve ser detectado para a correção da estimativa do contorno dada pelo modelo de evolução. A posição do contorno nas imagens é encontrada por meio da detecção de bordas. A detecção de bordas em imagens corrompidas por ruído foi um dos primeiros focos de interesse em visão computacional. Trabalhos como de Canny [3, 4] que propõe um método ótimo para extração de bordas se tornaram referência na área. Contribuições sobre o assunto ainda são encontradas em trabalhos recentes, como o excelente trabalho sobre o filtro SUSAN [21].

Segundo Canny, a extração de bordas em uma imagem começa pela convolução com a máscara de uma derivada de Gaussiana (em inglês, *Derivative of Gaussian* ou DoG). Em uma dimensão, como ilustra a figura 3.1(b), a DoG consiste na convolução de uma Gaussiana com centro na origem e variância σ^2 com uma função do tipo degrau. O sigma da Gaussiana está relacionado ao grau de suavização; quanto maior o seu valor, maior será a suavização aplicada à imagem. A resposta em frequência do filtro é da seguinte forma:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x-x_0)^2 + (y-y_0)^2}{2\sigma^2}} \quad (3.1)$$

em que x_0 e y_0 são as coordenadas onde se aplica a máscara e σ o fator de suavização. O valor de σ é determinado empiricamente.

Neste problema, outros procedimentos como histerese e supressão não-máxima não são necessários. Isto se justifica pelo fato da imagem de gradiente guardar várias características sobre a imagem original que são posteriormente perdidas nos procedimentos do filtro de Canny. Para uma seqüência de IRM, os resultados da convolução com uma máscara de uma derivada de Gaussiana para diferentes valores de σ são mostrados na figura 3.2. Dela concluímos que o valor ideal para σ encontra-se por volta de 0,8. Nota-se que em 3.2(a), as bordas ruidosas são praticamente eliminadas mas a posição da borda verdadeira não é precisa. Em 3.2(b), várias bordas espúrias são detectadas. Em 3.2(d) percebe-se como o ajuste de contraste melhora a precisão da detecção de bordas.

Na seção a seguir mostra-se como a técnica de deslocamento normal nos permite aperfeiçoar a extração de bordas, realizando reduzindo o problema em somente uma dimensão.

3.3 DESLOCAMENTO NORMAL PARA AJUSTE DO CONTORNO

O uso do deslocamento normal para estimar o deslocamento de uma curva é amplamente conhecido em visão computacional. O conceito é bastante simples: supondo que $r_c(s)$ seja o contorno identificado por meio da detecção de bordas e $r(s)$ o contorno estimado no instante atual, o deslocamento total $r_c(s)$ e

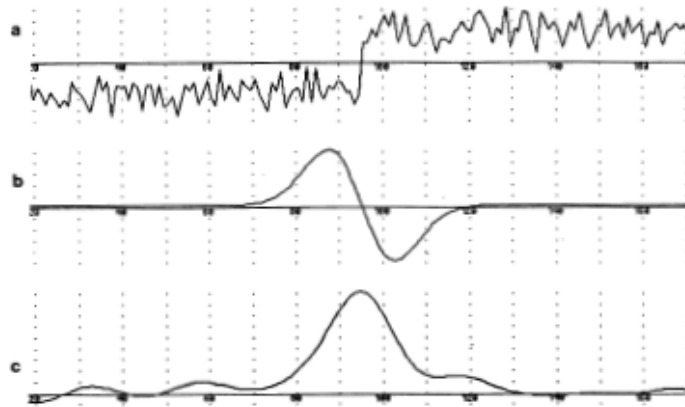


Figura 3.1: (a) Modelo unidimensional de uma borda; (b) A derivada de Gaussiana em uma dimensão: o valor de σ influencia sobre a suavização do sinal e conseqüentemente sobre a intensidade do desnível necessário para que seja detectado; (c) Resultado da detecção da borda em (a) (Fonte: [3]).

$r(s)$ é descrito como uma soma de vetores de componentes na direção normal e tangencial à curva $r(s)$. A figura 3.3 traz uma ilustração do processo. Supondo nula a componente tangencial, pode-se calcular o deslocamento simplesmente pela distância normal entre as curvas.

Conforme discutido na seção anterior, o processo de extração de bordas pode ser otimizado. Fazendo a procura das bordas somente na direção normal ao contorno, grande tempo computacional é salvo. Para tal tarefa, é necessário prestar atenção à detalhes como a interseção irregular de uma linha nos pixels de uma imagem (figura 3.4). Este é um problema conhecido em computação gráfica e costumeiramente é resolvido pelo uso de "anti-aliasing". Outra forma mais simples, mas menos eficiente computacionalmente é a convolução com a mascara DoG apenas nos pixels que são interceptados pela linha normal, que gera um resultado equivalente (já que a convolução é uma operação linear). Por meio do modulo do gradiente da imagem avaliado sobre uma linha normal, a posição das bordas é então determinada como sendo os máximos locais encontrados em determinada extensão E .

Dado um novo quadro da seqüência IRM, N linhas normais $l^{(n)}$, tal que $n = \{1, \dots, N\}$, interceptando os pontos de controle da curva predita $\hat{\mathbf{b}}_k$ são traçadas (figura 3.5). Máximos locais da imagem de gradiente são procurados ao longo destas linhas, por uma determinada extensão E . Para cada ponto de controle, um conjunto de M medidas $\mathbf{v}_j^{(n)}$ onde $j = \{0, \dots, M^{(n)}\}$ é gerado. A união dos N conjuntos formam o conjunto de medidas C_k no instante k .

Como o rastreamento esta submetido à ruído, artefatos, zonas de baixo contraste ou mesmo estruturas adjacentes tenham se aproximado suficientemente para serem detectadas, é natural que mais de uma borda seja encontrada. No pior cenário, apenas bordas espúrias são detectadas e o contorno real não aparece. O algoritmo de rastreamento deve necessariamente tratar tais eventos.

Como a informação da posição das bordas na vizinhança do contorno não é suficiente para manter o rastreamento em um ambiente altamente ruidoso, mais conhecimento sobre o verdadeiro contorno deve ser utilizado. Como a imagem de gradiente onde as bordas são detectadas consiste apenas do módulo do gradiente, a informação sobre a direção do gradiente não esta presente. A direção do gradiente indica se em uma determinada direção, os níveis de cinza variaram de alta para baixa intensidade ou vice-versa. Quando se procura a borda em uma determinada linha, é natural que o contorno em dois quadros seguidos possua a mesma direção. Desta forma, grande parte das bordas espúrias pode ser descartada por não se assemelharem à borda do verdadeiro contorno do objeto. Como a detecção da direção do gradiente não pode ser reaproveitada do calculo anterior da imagem de gradientes (para um caso mais geral, a direção

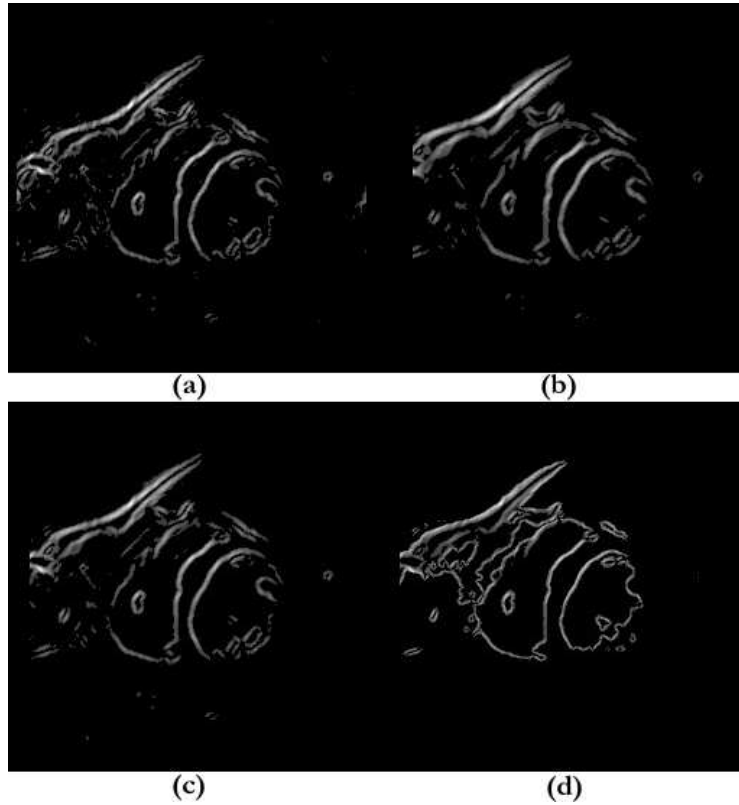


Figura 3.2: Resultados da convolução com uma DoG para σ iguais à (a) 1,2 (b) 0,3 (c) 0,8 (d) 0,8 com ajuste de contraste.

pode se conforme o contorno se deforma), é feita a análise da vizinhança de cada borda detectada (ver figura 3.6). A direção do gradiente de cada medida é calculada e comparada com a do verdadeiro contorno. Caso aponte outra direção, a medida é descartada.

Infelizmente, a simples procura em adjacências normais ao contorno relativo ao quadro anterior não elimina outro problema: a possibilidade de colapso ou interlaceamento dos pontos de controle da curva. Algumas restrições às medições são propostas para resolver tais problemas.

3.4 RESTRIÇÕES APLICADAS ÀS MEDIDAS

Durante o rastreamento de seqüências longas de imagens (possuindo mais de 10 segundos de duração), surge a necessidade de evitar o colapso dos pontos de controle. As restrições mencionadas nos capítulos anteriores são aplicadas ao processo de medição para estabilizar a curva e evitar o colapso dos pontos de controle.

Idealmente, é esperado que os pontos de controle mantenham uma distancia relativamente constante entre si durante o rastreamento. Ao invés de forçar tal comportamento pelo modelo de evolução, leves restrições são aplicadas ao conjunto de medidas C . Inicialmente, a normal $\mathbf{n}^{(n_0)}$ é calculada na posição do ponto de controle n_0 . Em seguida, o ponto médio $\mathbf{m}^{(n_0)}$ entre os pontos de controle adjacentes em $n_0 + 1$ e $n_0 - 1$ é calculado. A linha $I^{(n_0)}$ é então definida como sendo a reta que intercepta o ponto $\mathbf{m}^{(n_0)}$ e possui inclinação dada por $\mathbf{n}^{(n_0)}$. Sendo $\mathbf{v}_j = C_j^{(n_0)}$ o subconjunto de M_{n_0} medidas relativas ao ponto n_0 , as projeções d_j de cada medida na reta $I^{(n_0)}$ são calculadas segundo o produto escalar:

$$d_j = \langle \mathbf{v}_j - \mathbf{m}^{(n_0)}, \mathbf{n}^{(n_0)} \rangle$$

Tais projeções d_j formam pseudo-medidas \mathbf{p}_j da forma:

$$\mathbf{p}_j = \mathbf{m}^{(n_0)} + d_j \mathbf{n}^{(n_0)}$$

Finalmente, um conjunto \mathbf{v}'_j das pseudo-medidas \mathbf{p}_j é gerado contendo tais projeções. A união dos N conjuntos de pseudo-medições torna-se D e este conjunto é então usado no passo de observação do filtro estocástico. Com respeito à nomenclatura utilizada anteriormente, o conjunto \mathbf{v}'_j corresponde ao vetor \mathbf{z} de observação. A figura 3.7 ilustra o procedimento. A eficácia do rastreamento com e sem restrições é avaliada na seção a seguir, quando os resultados experimentais são apresentados.

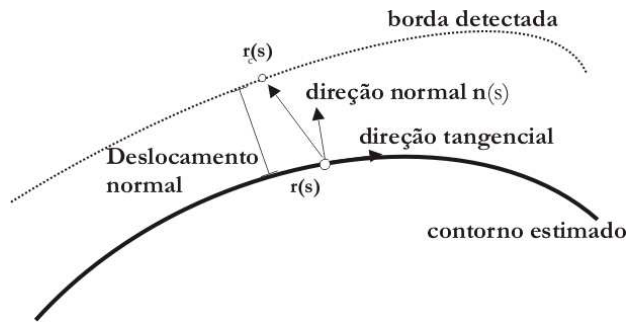


Figura 3.3: O deslocamento normal entre a curva proposta e a curva detectada pela extração de bordas. Nota-se que o deslocamento real tem componentes na direção normal e tangencial do contorno estimado.

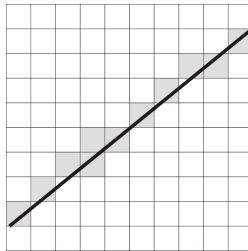


Figura 3.4: Uma linha intercepta os pixels de uma imagem de maneira irregular, gerando efeitos não desejados.

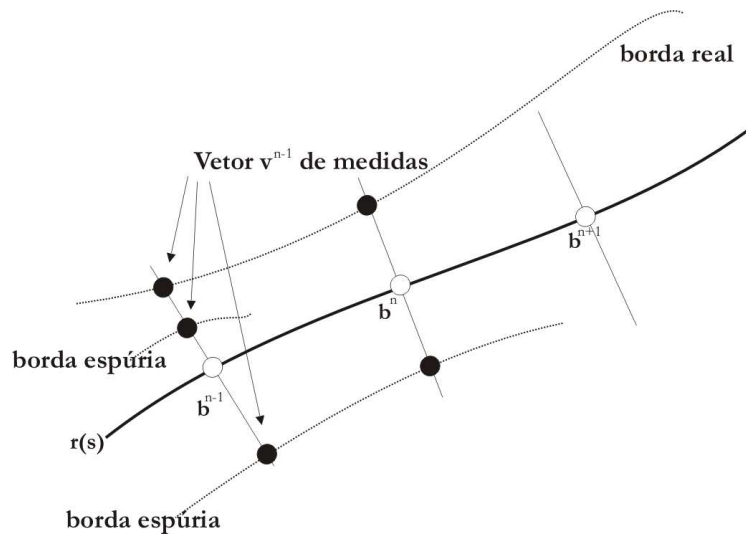


Figura 3.5: O uso do deslocamento normal para a obtenção de medidas. Nota-se que as linhas normais podem interceptar mais de uma borda ou mesmo borda alguma.

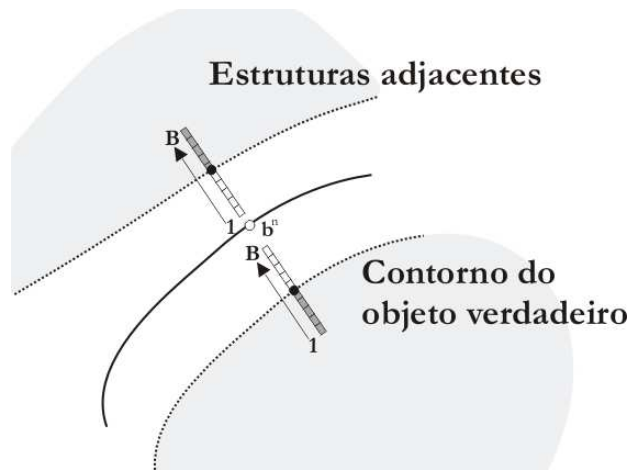


Figura 3.6: A direção do gradiente funciona como uma pré-seleção das medidas. Para cada medida, B pixels são avaliados na ordem indicada. O somatório dos primeiros $B/2$ pixels é comparado com o dos últimos e desta forma a direção do gradiente é determinada.

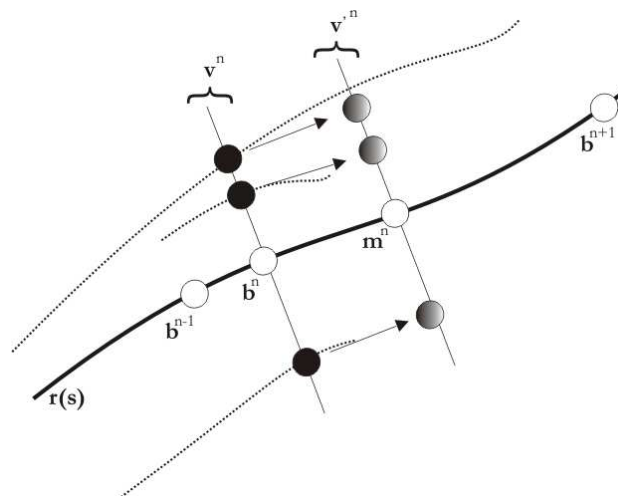


Figura 3.7: Procedimento da obtenção do vetor de pseudo-medicoes \mathbf{v}^j_j .

4 RESULTADOS EXPERIMENTAIS

4.1 INTRODUÇÃO

Os experimentos têm como propósito avaliar o desempenho das técnicas apresentadas nos capítulos anteriores. Foi usada uma seqüência de IRM gentilmente cedida pelo professor Krishna Nayak, da University of Southern California (USC). Ela consiste de uma seqüência de 8 segundos, à uma taxa de atualização de 60Hz.

O algoritmo desenvolvido consiste dos seguintes passos, descritos na figura 4.1. O projeto do algoritmo foi realizado usando Microsoft Visual C++ versão 6, usando a biblioteca OpenCV (**Open** Computer Vision library, disponível no endereço <http://www.intel.com/software/products/ipp/index.htm>). Quatro experimentos são conduzidos de maneira à:

- Avaliar o rastreamento sem a filtragem estocástica;
- Verificar o rastreamento com e sem restrições de medidas utilizando o filtro de Kalman;
- Avaliar o desempenho do filtro Condensation e o comparar com o filtro de Kalman;

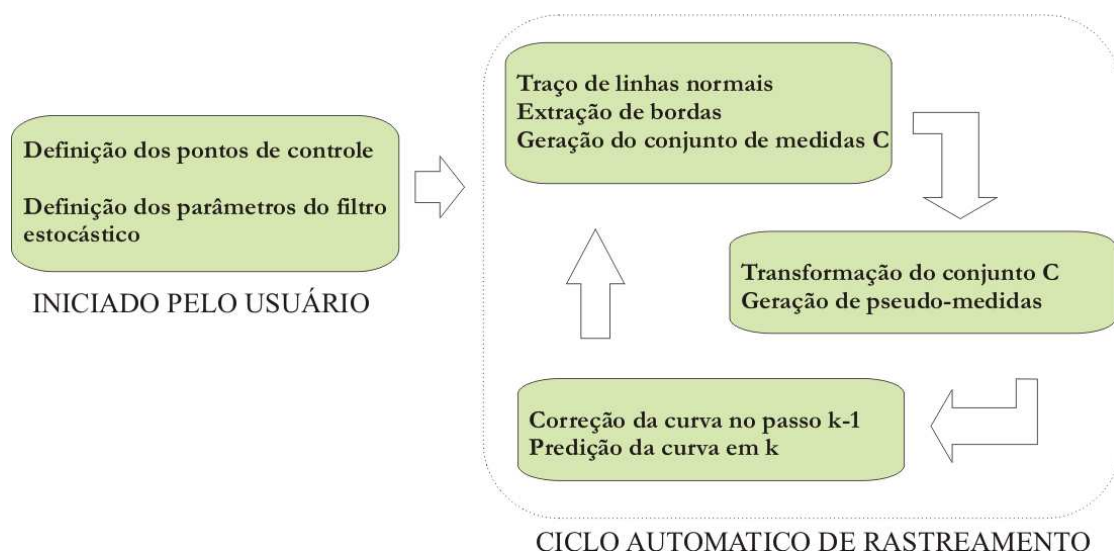


Figura 4.1: Fluxograma do algoritmo de rastreamento. Após a iniciação do filtro pelo usuário, toma lugar a seqüência predição/correção do modelamento dinâmico desenvolvido nos capítulos anteriores.

Para melhor ilustrar o rastreamento, uma expansão da imagem original de 128×128 para 256×256 (usando interpolação linear) é realizada. O ônus que esta expansão acarreta será devidamente discutido quando tratarmos do esforço computacional realizado em cada etapa do rastreamento.

Uma análise visual da seqüência de IRM nos mostra a necessidade do rastreamento lidar com deslocamentos rápidos do contorno, já que o deslocamento entre quadros do contorno do VE no momento da contração do músculo do coração, mesmo à uma taxa de atualização de 60Hz, é bastante rápido. Os filtros estocásticos devem então estar bem regulados. Um mesmo conjunto de parâmetros, discriminado na tabela 4.1 foi usado nos experimentos para que os resultados fossem comparados sem viés.

Os valores dos parâmetros de incerteza dos filtros estocásticos são especificados para cada filtro usado. Os resultados dos experimentos estipulados acima são apresentados nas seções seguintes.

Tabela 4.1: Parâmetros do algoritmo de rastreamento usado nos experimentos.

Descrição do parâmetro	Valor
Ordem do polinômio da <i>B-spline</i>	3
Limiar L do ajuste de contraste	92 (para escala de níveis de cinza entre 0 e 255)
Numero de pontos de controle N	11 (ao menos que seja especificado outro valor)
α	0,1
β	0,05
Valor de σ para mascara DoG	0,8
Tamanho E das linhas de procura	20

4.2 IMPLEMENTAÇÃO SEM FILTRAGEM ESTOCÁSTICA

Inicialmente, o rastreamento é realizado considerando apenas o deslocamento normal, sem filtragem estocástica. Nesta implementação quer-se mostrar que um rastreamento simplesmente determinístico falha por não tratar as incertezas associadas ao problema. Analisando a figura 4.2, percebe-se que o rastreamento do contorno é perdido decorridos 0,3 segundos. A seqüência de quadros de 4.2(a) ilustra o rastreamento do contorno do VE usando simplesmente o deslocamento normal para a estimação do contorno à cada quadro. Em 4.2(b) ilustra como o interlaceamento dos pontos de controle embaraça a *B-Spline*.

4.3 IMPLEMENTAÇÃO BASEADA NO FILTRO DE KALMAN

Dois experimentos testam o rastreamento com e sem o passo de restrição das medidas. O filtro é iniciado provendo as informações sobre o contorno inicial e as incertezas do processo. O contorno do VE é iniciado por um conjunto de 11 pontos de controle definidos manualmente (figura 4.7). As matrizes \mathbf{Q} , \mathbf{R} e \mathbf{P}_0 de dimensões 11×11 , que descrevem respectivamente a incerteza do processo, a incerteza das medições e a incerteza sobre o estado inicial tem a forma abaixo:

$$\mathbf{Q} = 0,1^2 \mathbf{I}$$

$$\mathbf{R} = 3,0^2 \mathbf{I}$$

$$\mathbf{P}_0 = 1,0^2 \mathbf{I}$$

onde \mathbf{I} tem dimensão apropriada 11×11 .

Sem restrições de medidas, decorridos 0,8 segundos, o rastreamento do contorno do VE é perdido (ver figura 4.3). Os pontos não obedecem a uma distancia segura e acabam por se entrelaçar e colidir. Em relação com a seqüência sem filtragem estocástica, o rastreamento usando o filtro de Kalman sem restrições, apesar de não ter sucesso foi sustentado por muito mais tempo.

No experimento (figura 4.4) seguinte nota-se que mesmo não restringindo o movimento dos pontos de controle, o rastreamento é mantido quando as restrições às medições são usadas. Nota-se também que pelo uso da restrição de medidas, a posição dos pontos de controle se distribui ao longo da seqüência, corrigindo a distribuição irregular da iniciação. Mesmo que em determinadas regiões a diferença entre o contorno estimado e o contorno real seja grande, o rastreamento se mostrou estável o suficiente durante toda a seqüência. Este erro do rastreamento é resultado de uma série de fatores como a falta de pontos de controle para correta representação da geometria do contorno do VE, imprecisão na descrição do contorno real pela *B-Spline*, perturbações temporárias do rastreamento por estruturas adjacentes ou ruído, etc.

Uma possível solução para melhorar o desempenho do filtro de Kalman seria a criação de um filtro adaptativo onde os parâmetros que descrevem as incertezas do processo balanceassem as modificações nas condições do rastreamento, como aumento do ruído, perda do rastreamento etc.

Como o aumento do número de pontos de controle tem implicações na estabilidade do rastreamento, seria interessante a procura de uma descrição do contorno rastreado mais precisa. Como a *B-Spline* segue proximamente os pontos e controle mas não os interpola ("fecho convexo"), outra forma de descrição do contorno, que interpolasse os pontos de controle aumentaria a precisão do rastreamento.

4.4 IMPLEMENTAÇÃO BASEADA NO FILTRO CONDENSATION

O uso do filtro Condensation tem como principal vantagem a consideração de múltiplas hipóteses no passo de observação. No rastreamento com filtragem de Kalman, uma vez que o contorno é perdido raramente ele é restabelecido. Tal problema advém da ausência de restrições à deformação da curva: uma vez que um ponto de controle é atraído por medidas ruidosas, ele pode distanciar-se perigosamente do contorno real.

Já pelo rastreamento usando o filtro Condensation, múltiplas medidas são sempre consideradas. Como se pode verificar em alguns quadros extraídos da seqüência do rastreamento (figura 4.5), em diversas ocasiões o contorno rastreado é perturbado por estruturas adjacentes, que o afasta do contorno real. Este efeito aparentemente indesejado reflete como o rastreamento interpreta mais de uma medida para um ponto de controle: cada medida representa equiprovavelmente o contorno real. Deste fato advém a robustez do filtro Condensation, onde o contorno real é considerado juntamente com o ruído.

Outra característica notável é a boa capacidade de recuperação: certos instantes mostram o rastreamento sendo perdido em alguns quadros e recuperado em seguida. Os resultados apresentados na figura 4.5 foram obtidos usando a incerteza de processo e medição ditada abaixo:

$$\begin{aligned} \mathbf{Q} &= 15^2 \mathbf{I} \\ r^2 &= 2^2 \\ \mathbf{P}_0 &= 2\mathbf{Q} \end{aligned}$$

em que o parâmetro r^2 representa a variância de cada Gaussiana ilustrada na figura 2.11. Foi usado $N_p = 1000$ partículas neste experimento. Um número de partículas acima de 400 se mostrou necessário para esta aplicação, pois abaixo deste valor alguns efeitos indesejados podem ocorrer.

O experimento de avaliação deste filtro usa uma quantidade menor de pontos de controle que em outros experimentos simplesmente para uma melhor ilustração do rastreamento. Pode-se verificar que o rastreamento usando múltiplas hipóteses não representa o contorno real tão bem quanto o filtro de Kalman, já que todas as medidas são consideradas para a atualização de cada ponto de controle. Por outro lado, a probabilidade de sucesso do rastreamento é bem maior, visto que a capacidade de recuperação da perda do rastreamento é maior.

4.5 ESFORÇO COMPUTACIONAL

Para avaliar o esforço computacional do algoritmo desenvolvido, o tempo necessário para execução de cada passo da figura 4.1 é mensurado. O computador usado para este experimento foi um computador pessoal com processador Pentium M 1,73GHz. Como a taxa de atualização das imagens é 60Hz, espera-se que o processamento de cada quadro funcione em uma janela menor que $1/60 = 16,666$ ms.

Tabela 4.2: Tempo de execução de cada passo do rastreamento (figura 4.1), usando o filtro de Kalman. Onze pontos de controle são usados para representar a curva.

	Tarefa	Tempo exigido (ms)
1	Traço das linhas normais, extração de bordas, obtenção de Medidas	0,228
2	Regularização das medições (Aplicação de restrições às medidas)	0,002
3	Passo de correção do filtro de Kalman	0,306
4	Passo de predição do filtro de Kalman	0,076
5	Carregamento de outro quadro e detecção bordas	49,431
	Total	50,043

Tabela 4.3: Tempo de execução de cada passo do rastreamento (figura 4.1), usando o filtro Condensation. Onze pontos de controle são usados para representar a curva.

	Tarefa	Tempo exigido (ms)
1	Traço das linhas normais, extração de bordas, obtenção de Medidas	0,229
2	Regularização das medições (Aplicação de restrições às medidas)	0,003
3	Atribuição de pesos às partículas do filtro Condensation (usando as pseudo-medições)	7,746
4	Re-amostragem das partículas e predição pelo modelo de evolução	6,144
5	Carregamento de outro quadro e detecção bordas	51,341
	Total	65,552

Cabe lembrar que a extração de bordas não fora otimizada, pois a geração da imagem de gradiente inteira é útil para a ilustração do rastreamento. A expansão da imagem em duas vezes implica também em tempo de calculo extra mas sem este aumento a visualização da imagem seria bastante difícil. O peso computacional dos dois filtros estocásticos é avaliado para duas formas de inicialização, uma com 11 pontos de controle e outra com 7 pontos. O conjunto maior de 11 pontos é inicialmente usado. Posteriormente diminuindo-se a resolução da representação do contorno, a influência da escolha dos pontos de controle no custo computacional é avaliada.

Cinco experimentos foram realizados. A meta é testar cada passo do algoritmo de rastreamento e medir o efeito da variação do número de pontos de controle, o desempenho de cada filtro estocástico e, especificamente, o efeito da variação do número de partículas do filtro Condensation. As tabelas 4.2, 4.3, 4.4, 4.5 e 4.6 apresentam o tempo exigido para execução de cada passo do algoritmo.

Tabela 4.4: Repetição da medida do tempo de execução de cada passo do rastreamento, usando o filtro de Kalman, para sete pontos de controle são usados para representar a curva.

Tarefa	Tempo exigido (ms)
1	0,151
2	0,001
3	0,137
4	0,021
5	45,575
Total	45,885

Tabela 4.5: Repetição da medida do tempo de execução de cada passo do rastreamento, usando o filtro Condensation, para sete pontos de controle são usados para representar a curva.

Tarefa	Tempo exigido (ms)
1	0,214
2	0,002
3	5,070
4	3,284
5	45,575
Total	54,146

Tabela 4.6: Repetição da medida do tempo de execução de cada passo do rastreamento, usando o filtro Condensation, para uma redução de 700 partículas (novo $N_p = 400$) e sete pontos de controle.

Tarefa	Tempo exigido (ms)
1	0,232
2	0,002
3	1,977
4	1,547
5	45,575
Total	54,146

Analisando os dados obtidos, percebe-se que a redução dos pontos de controle não acarreta em uma grande economia de processamento. Qualitativamente, tal redução acarreta no aumento do erro da representação do contorno, que se distancia do contorno real por falta de pontos suficientes para representação da curva.

Já quando o número de partículas é diminuído, uma economia drástica é feita, diminuindo 1,7 vezes o tempo de processamento necessário. Como neste caso não se verifica grandes diferenças no desempenho de rastreamento entre 1000 e 400 partículas, deve-se regular o filtro para que ele funcione bem com o mínimo de partículas, pois isto tem um efeito direto no seu custo computacional.

Como pode ser notado, o maior custo computacional está ligado à detecção de bordas nas imagens. A otimização deste processo se mostra crucial, pois caso contrario um custo excessivo é exigido, especialmente para uma aplicação em tempo real. Outro problema relacionado à tempo de processamento refere-se as funções da biblioteca *OpenCV* usadas para implementação do algoritmo. O uso desta biblioteca para a realização do projeto não permite a realização de certas tarefas em um espaço de tempo aceitável. Algumas destas tarefas são:

- **A mostra de uma imagem em uma janela** A função *cvShowImage()* é bastante lenta, implicando em um atraso para a simples ilustração das imagens processadas. O uso de outras maneiras de mostrar uma imagem se mostra necessário.
- **O registro da seqüência rastreada em vídeo** Salvar um arquivo em vídeo usando *OpenCV* é também bastante demorado, além de esbarrar no problema da seleção de um *codec* que permita a visualização da seqüência posteriormente.

Tais problemas deverão ser levados em conta quando desenvolvendo um algoritmo de rastreamento que funcione verdadeiramente em tempo real. Por fim pode-se concluir que o rastreamento pode funcionar em tempo real, à 60 Hz, já que o tempo de execução dos passos do algoritmo de rastreamento somam (com

exceção da detecção de bordas) 0,4 ms e 10 ms, para os filtros de Kalman e Condensation respectivamente. Com a execução das modificações propostas para otimizar o processo de detecção de bordas, o algoritmo de rastreamento atende às exigências de tempo de processamento.

4.6 AVALIAÇÃO DA METODOLOGIA PROPOSTA

As restrições aplicadas às medidas associadas à filtragem estocástica foram bem sucedidas, ao permitir o rastreamento do contorno do VE durante vários ciclos do coração. Como avaliação final do rastreamento, o erro entre o contorno ideal e o contorno obtido deve ser avaliado.

Como mencionado na introdução, problemas como poças de sangue e zonas de baixo contraste nas imagens deformam a representação do ventrículo nas imagens de IRM (figura 4.8). Um bom algoritmo de rastreamento deve vencer tais imperfeições e reproduzir da melhor maneira o contorno real do ventrículo. Isto é conseguido somente por meio da introdução de conhecimento anterior no rastreamento. Infelizmente, o modelo aqui proposto faz apenas leves suposições sobre o movimento da parede do coração nas regiões onde se localizam os pontos de controle e não do contorno do coração como um todo, limitando assim os resultados. Pode-se enxergar claramente nos vários quadros da figura 4.9 que as imperfeições das imagens não são corrigidas pelo esquema de rastreamento proposto.

Entretanto, a introdução de conhecimento anterior para o rastreamento do VE já foi discutida anteriormente. Esta proposta esbarra nas diferenças anatômicas entre pacientes e a geração de um modelo dinâmico de deformação é complexa e demanda tempo. Uma saída encontrada para esta dificuldade é parcialmente encontrada na segmentação por *snakes*. Ajustando corretamente os modelos elásticos de uma *snake*, bons resultados são obtidos na correção das imperfeições das imagens (ver a seqüência de teste apresentada em [10]). Infelizmente esta técnica envolve iterações para atingir uma curva ótima, dificultando bastante seu uso em tempo real.

Alguns testes bem sucedidos mostram que a incorporação do conhecimento anterior das características geométricas do ventrículo ajudam bastante a estabilidade do rastreamento (ver figura 4.6). Como o ventrículo possui um forma aproximadamente elipsoidal, pode-se definir separadamente os vetores normais \mathbf{n} de cada ponto de controle (baseado no elipsóide que melhor representa o contorno em dado instante) e restringir seus graus de liberdade. Aliado ao uso da regularização das medidas para manter uma distribuição mais homogênea dos pontos, tal solução permite um aumento dos número de pontos de controle sem perda de estabilidade durante rastreamento. Esta variante do algoritmo proposto apelidada de 'Star' mostrou um melhor desempenho ao passo que se restringe à uma forma específica do ventrículo. Sua formulação é apresentada em anexo.

Finalmente, é importante lembrar alguns detalhes sobre a seqüência de IRM que estava disponível para a realização do experimento. A seqüência de IRM possui tamanho 128 por 128 e, portanto fora necessário a expansão para 256 por 256 para uma melhor visualização do rastreamento. Para isto, uma interpolação linear dos pixels foi feita. Um sério problema relacionado à seqüência era o fato dela estar compactada em AVI. Uma compressão em tal tipo de imageamento pode implicar no aumento drásticas no nível de ruído das imagens, além de deformação na representação de estruturas na imagem. Outras seqüências de teste são portanto necessárias para testar mais amplamente a técnica de rastreamento desenvolvida.

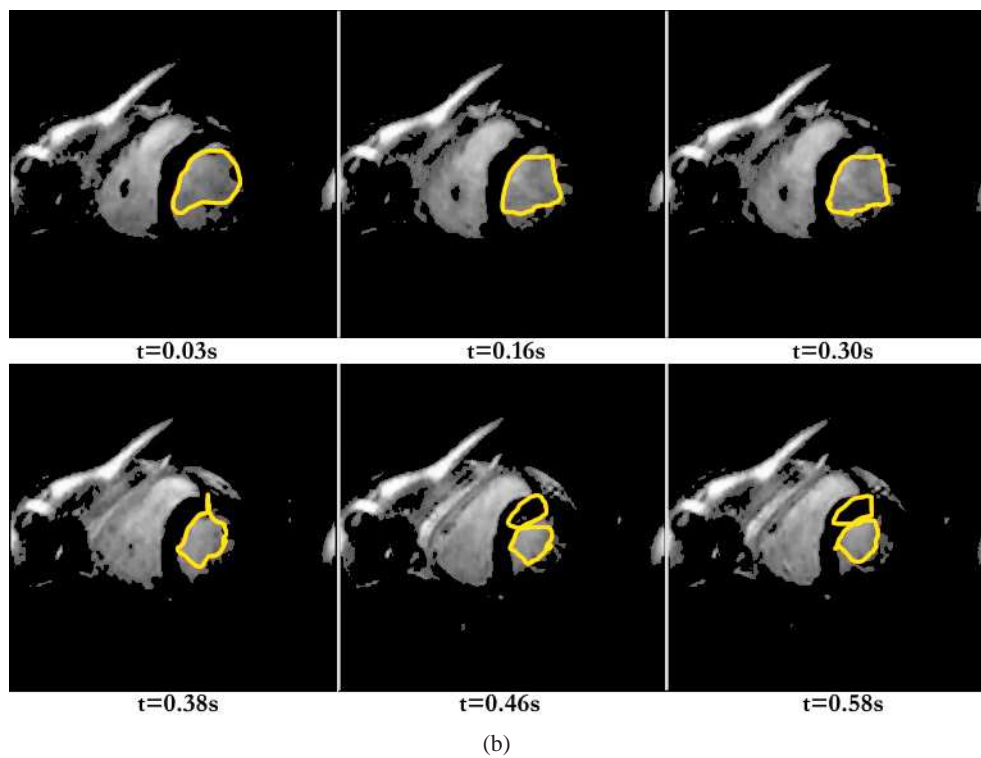
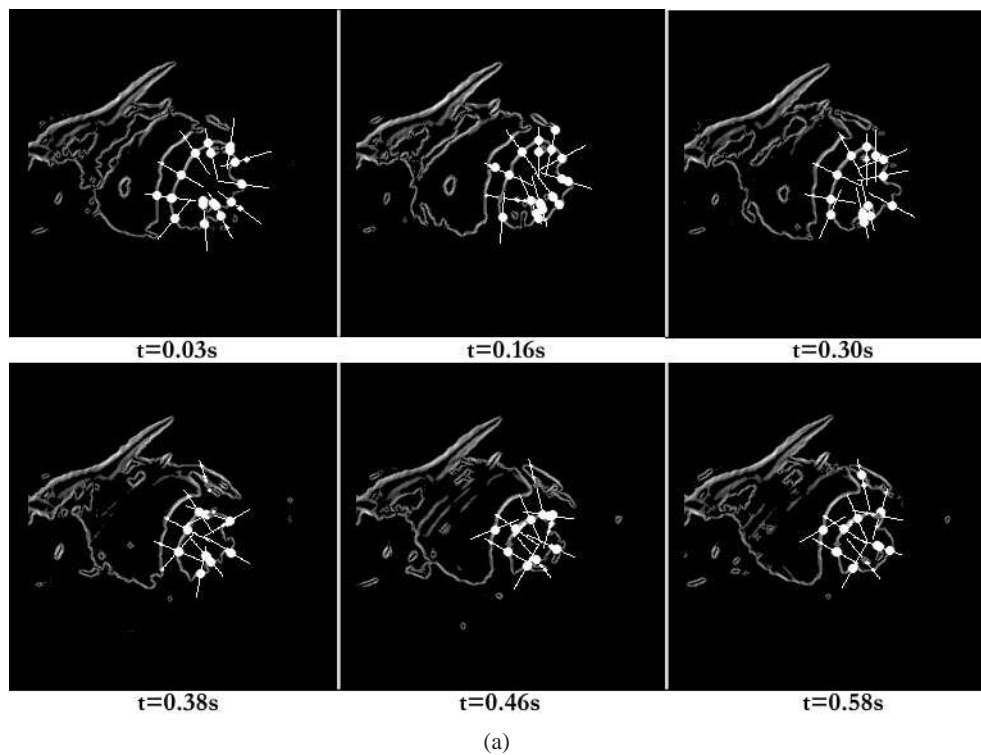


Figura 4.2: Rastreamento de contorno sem filtragem estocástica. Os círculos grandes marcam as bordas detectadas enquanto que os círculos pequenos marcam os pontos de controle (centro da linha de procura).

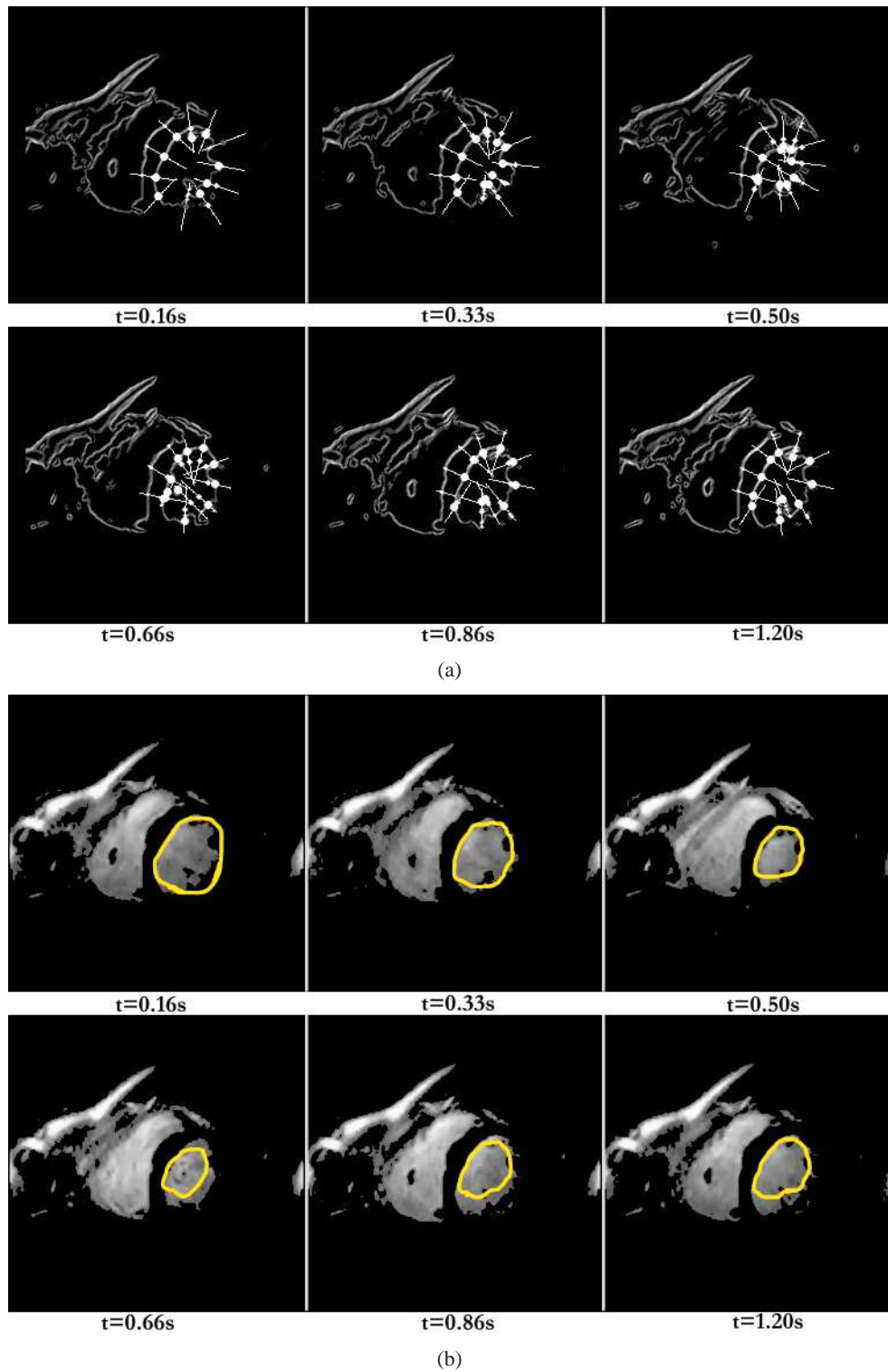


Figura 4.3: Rastreamento sem restrição de medidas usando filtragem de Kalman. Em (a) percebe-se que, pelos vários graus de liberdade do esquema proposto, os pontos de controle acabam por se chocar e o rastreamento do contorno é conseqüentemente perdido.

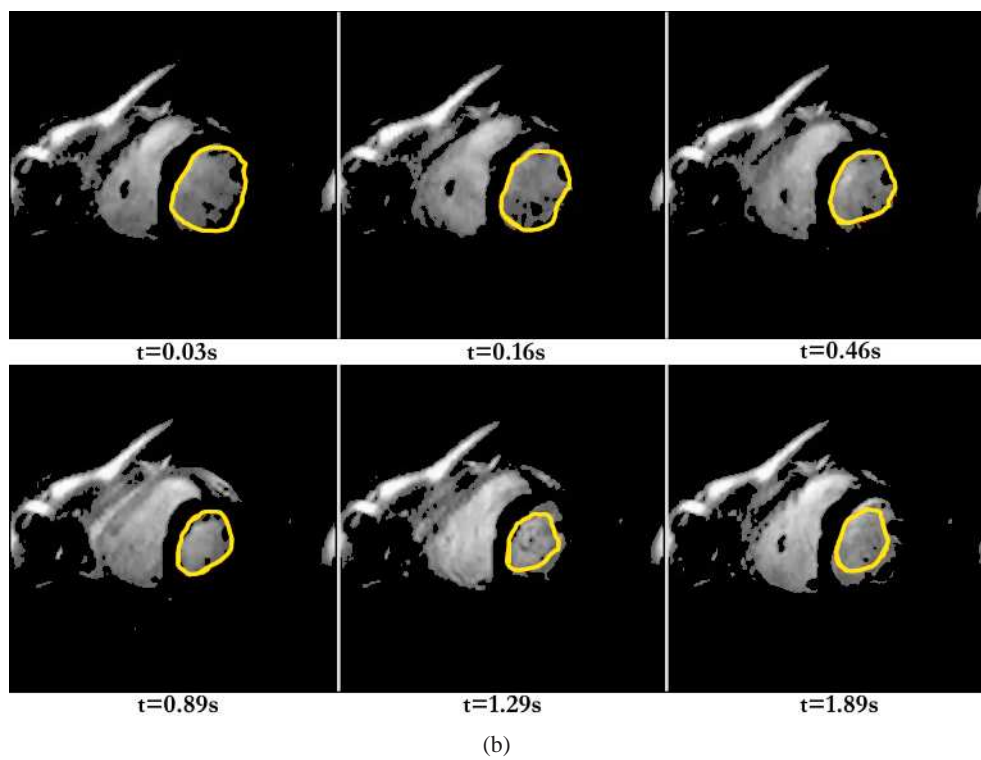
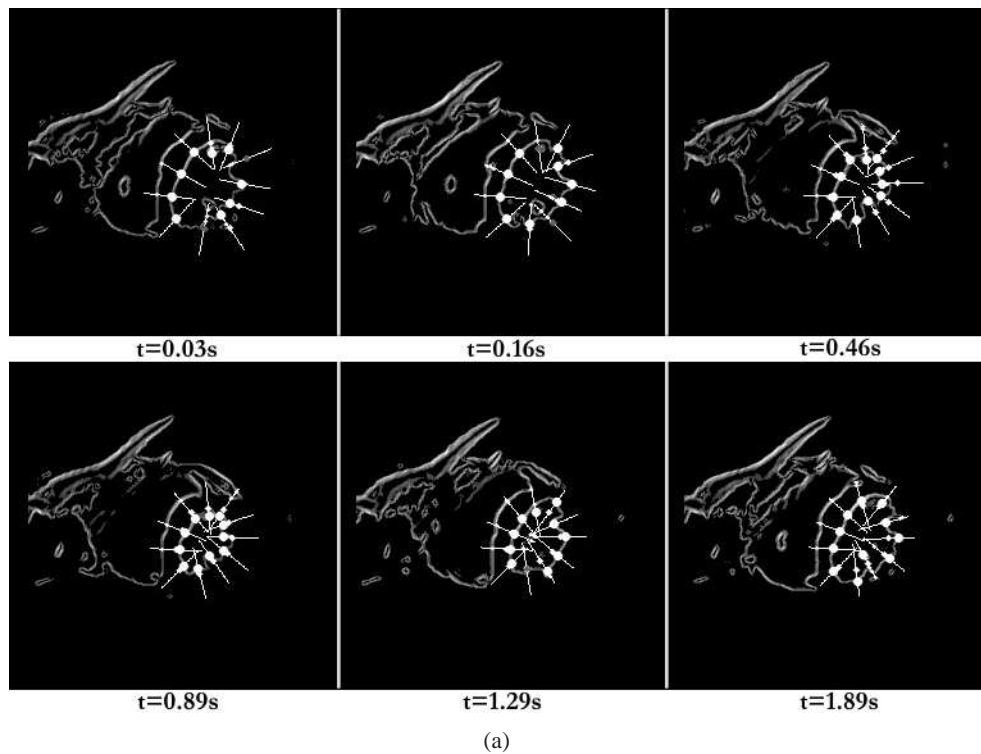


Figura 4.4: Rastreamento com restrição de medidas usando filtragem de Kalman. Em (a) é visto quadros de um ciclo cardíaco. Se um ponto de controle não encontra medidas para sua correção, ele é marcado como um círculo cinza.

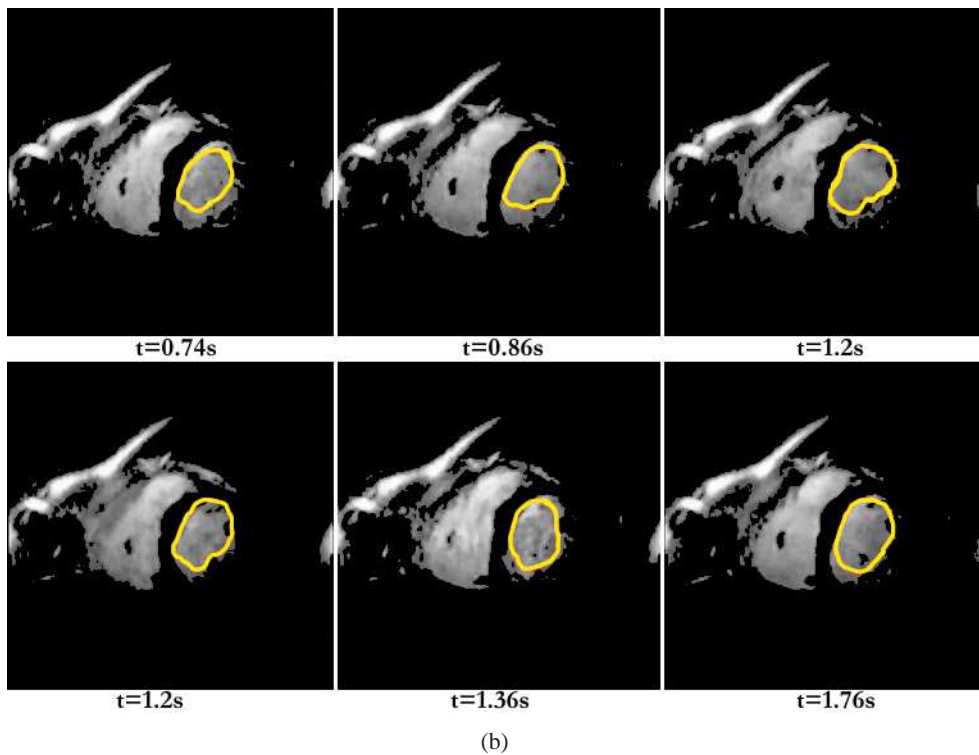
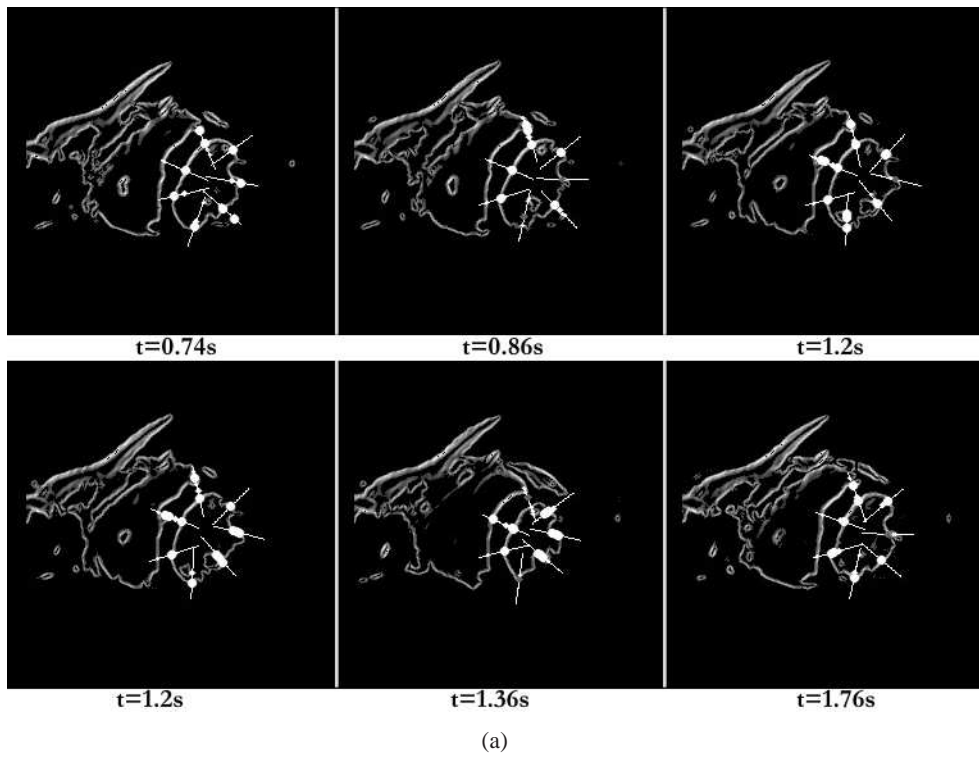


Figura 4.5: Rastreamento com restrição de medidas usando o filtro Condensation.

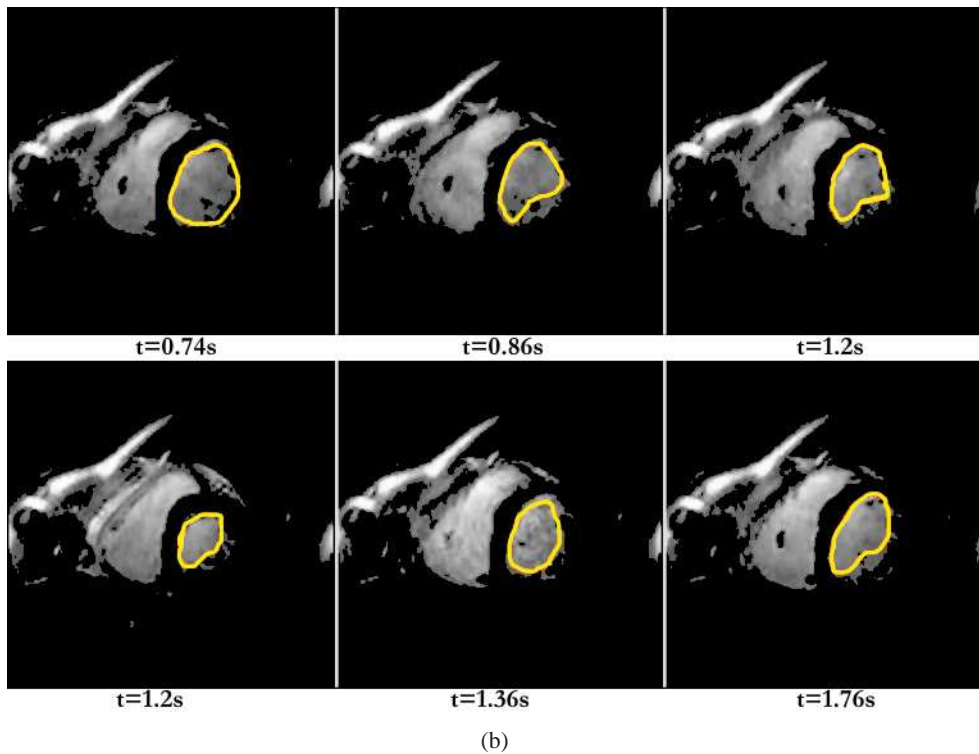
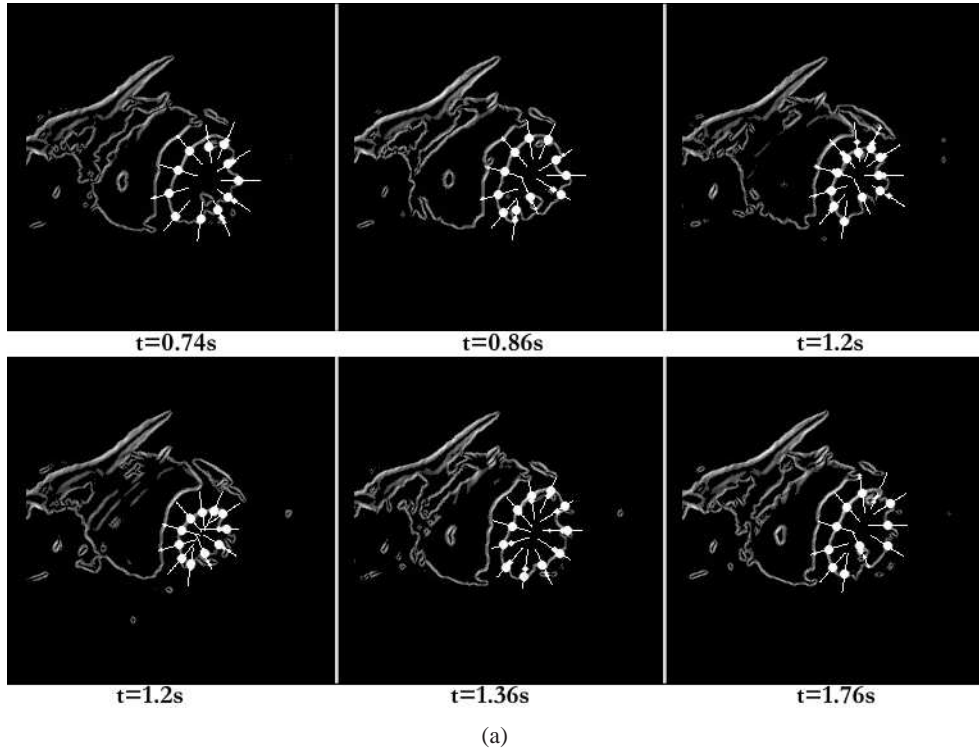


Figura 4.6: Rastreamento usando *Star*. Esta variante usa o filtro de Kalman e sua formulação encontra-se no Anexo.

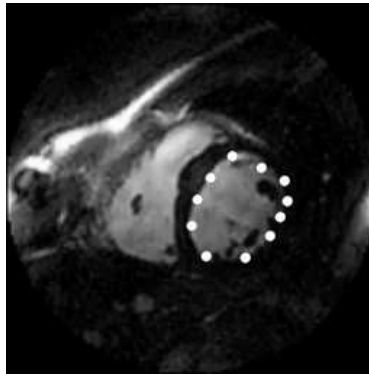


Figura 4.7: O conjunto de pontos de controle selecionados manualmente para iniciação da curva. Nota-se que a distância entre pontos não é constante devido à imprecisão na iniciação.

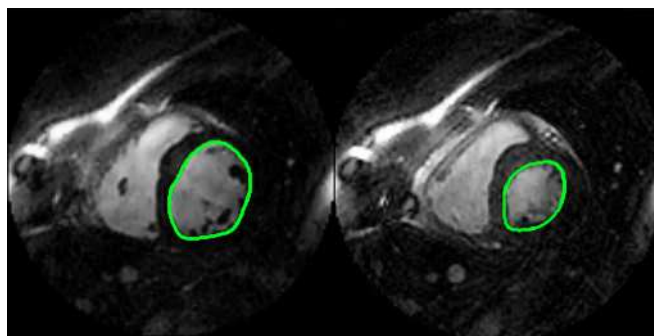


Figura 4.8: O contorno ideal do ventrículo esquerdo (em verde). As imperfeições na reconstrução das imagens são aspectos importantes ao rastreamento.

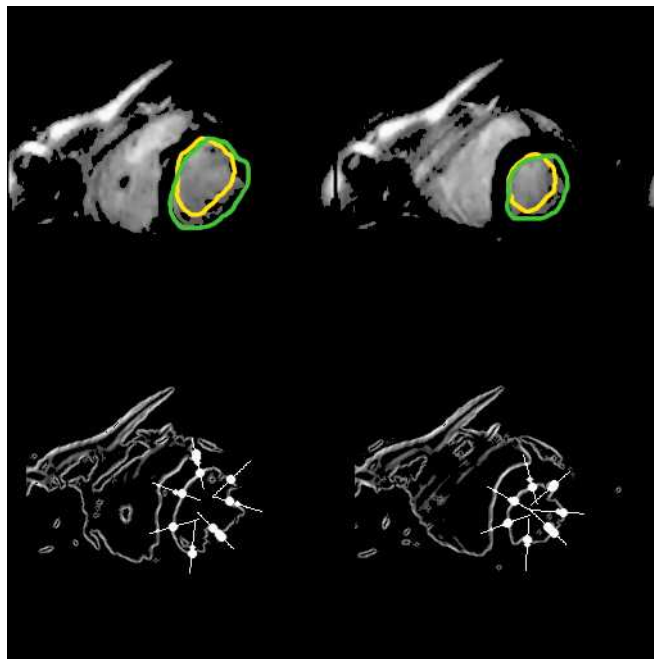


Figura 4.9: A diferença entre o contorno ideal (verde) e o estimado pelo rastreamento (amarelo). É notável a influência no rastreamento das imperfeições da representação do ventrículo.

5 CONCLUSÃO

O objetivo deste trabalho é explorar a filtragem estocástica no rastreamento do contorno do ventrículo esquerdo, em tempo real. O contorno é representado por uma *B-Spline* e modelado com mínimas restrições de deformação. Pode-se interpretar este rastreamento aqui realizado como o rastreamento de cada ponto de controle separadamente, que serve para traçar uma *B-Spline*.

Os resultados dos experimentos apresentados na seção anterior mostram o sucesso das restrições usadas para manter a estabilidade do rastreamento de seqüências longas. Esta estabilidade era necessária visto que o rastreamento deve se manter durante vários ciclos do batimento do coração para que dados relevantes sejam colhidos.

A grande portabilidade que o algoritmo desenvolvido proporciona baseia-se na quase ausência de restrições à deformação do objeto rastreado. Infelizmente os problemas que isto acarreta não se restringem à simples perda de rastreamento. Uma limitação mais grave se verifica quando o rastreamento não consegue vencer as imperfeições da representação pelas imagens de IRM do contorno real do VE. Este último defeito, mais grave, apenas pode ser tratado usando conhecimento anterior sobre a forma geométrica do ventrículo e seu comportamento dinâmico.

A seguir são discutidas algumas linhas de pesquisa que o trabalho aqui realizado nos permite enxergar.

5.1 TRABALHOS FUTUROS

Diversas novas frentes de trabalho que abrange o rastreamento do contorno do VE podem ser vistas. Um grande trabalho ainda deve ser feito na formulação do modelo dinâmico usado no rastreamento, visto que nossa revisão bibliográfica conta com poucos trabalhos realizados no assunto que utilizam modelos de deformação para estruturas anatômicas complexas como o coração.

Como pudemos perceber na literatura, a maior parte dos trabalhos usa *snakes* para realizar a segmentação. Esta técnica produz bons resultados mas não usa grande conhecimento sobre a dinâmica do coração. A vantagem do uso de outros modelos dinâmicos de deformação é a melhor aplicabilidade sobre uma gama maior de situações, tornando desnecessária a formulação de novos modelos de deformação para cada novo problema, como quando se usam *snakes*.

Outro aspecto do trabalho que necessita de trabalho é a transformação da curva estimada pelo rastreamento em dados que possam ser utilizados por um profissional da saúde. Atualmente é pouco evidente o comportamento de uma determinada região do coração. Isto impossibilita a detecção das mais simples disfunções, como o infarto de determinadas regiões.

Finalmente outro campo de trabalho encontra-se no tratamento das imagens de ressonância para concerto de falhas e redução de ruído. Mesmo que para o problema de rastreamento o melhoramento das imagens não seja necessário, quando se busca fazer o processamento das imagens em lote bastante tempo computacional é disponível e melhores resultados podem ser obtidos se as imagens são pré-tratadas.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] BLAKE, A.; ISARD, M. *Active Contours*. [S.l.]: Springer, 2002.
- [2] ISARD, M.; BLAKE, A. Contour tracking by stochastic propagation of conditional density. In: . [S.l.: s.n.], 1998.
- [3] CANNY, J. F. *Finding edges and lines in images*. Dissertação (Mestrado) — MIT Cambridge, 1983.
- [4] CANNY, J. F. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Machine Intell*, PAMI 8, p. 679–698, 1986.
- [5] BAR-SHALOM, Y.; FORTMANN, T. E. *Tracking and Data Association*. [S.l.]: Academic Press, 1988.
- [6] KASS, M. W.; TERZOPOULOS, D. Snakes: Active contour models. *1st International Conference on Computer Vision*, p. 259–268, 1987.
- [7] LI, P.; ZHANG, T.; PECE, A. E. C. Visual contour tracking based on particle filters. *Image Vision Comput*, v. 1, 2003.
- [8] GAO, J.; KOSAKA, A.; KAK, A. C. A multi-kalman filtering approach for video tracking of human-delineated objects in cluttered environments. In: . [S.l.: s.n.], 2004.
- [9] ARULAMPALAM, M. S. et al. A tutorial on particle filters for online nonlinear non-gaussian bayesian tracking. In: . [S.l.: s.n.], 2002. v. 50.
- [10] GOTARDO, P. F. U. et al. A new deformable model for boundary tracking in cardiac mri and its application to the detection of intra-ventricular dyssynchrony. In: *Proceedings of IEEE Computer Society Conference on Computer Vision and Patter Recognition (CVPR)*. [S.l.]: IEEE Computer Society, 2006.
- [11] PARK, J.; METAXAS, D. Deformable models with parameter functions for cardiac motion analysis from tagged mri data. *Transactions on Medical Imaging*, IEEE, v. 15, n. 3, p. 278–289, June 1996.
- [12] GEIGER, D. et al. Dynamic programming for detecting, tracking and matching deformable contours. In: *Pattern Analysis and Machine Intelligence*. [S.l.: s.n.], 1995. v. 17.
- [13] TERZOPOULOS, D.; MCINERNEY, T. Deformable models in medical image analysis. *Workshop on Mathematical Methods in Biomedical Image Analysis (MMBIA '96)*, IEEE, 1996.
- [14] XU, C. *Deformable Models With Application To Human Cerebral Cortex Reconstruction From Magnetic Resonance Images*. Tese (Doutorado) — Johns Hopkins University, January 1999.
- [15] XU, C.; PRINCE, J. L. Reconstruction of human cerebral cortex from magnetic resonance images. *Transactions on Medical Imaging*, IEEE, v. 18, n. 6, p. 467–480, June 1999.
- [16] XU, C.; PRINCE, J. L. Snakes, shapes, and gradient vector flow. *Transactions on Image Processing*, v. 7, n. 3, p. 359–369, March 1998.
- [17] MEYER, F. G.; SINUSAS, A. J.; DUNCAN, J. S. Tracking myocardial deformation using phase contrast mr velocity fields: a stochastic approach. In: . [S.l.: s.n.], 1996. p. 453 465.
- [18] COOTES, T.; TAYLOR, C.; GRAHAM, J. Active shape models: Their training and application. *CVIU*, IEEE, v. 61, n. 1, p. 38–59, January 1995.
- [19] KITSON, F. L. An algorithm for curve and surface fitting using b-splines. In: . [S.l.: s.n.], 1989.

- [20] SALMOND, D. Target tracking: Introduction and kalman tracking filters. *IEE International Seminar Target Tracking*, p. 16–17, October 2001.
- [21] SMITH, S. M.; BRADY, J. M. Susan - a new approach to low level image processing. *UK Defense Research Agency*, v. 1995.

I. A VARIANTE 'STAR'

Esta variante do rastreamento proposto caracteriza-se pela redução do grau de liberdade de movimento dos pontos de controle. Enquanto que anteriormente o vetor normal, calculado para verificar o deslocamento normal entre quadros era avaliado à cada quadro levando em conta apenas a posição dos pontos de controle adjacentes, no algoritmo 'Star' ele é baseado em uma forma geométrica esperada para o ventrículo esquerdo.

Levando em consideração que o ventrículo esquerdo pode ser aproximado por um elipsóide (ver trabalho da elaboração de um modelo paramétrico para o VE em [11]), podemos calcular o vetor normal \mathbf{n} dos N pontos de controle baseado no vetor normal de um elipsóide que melhor represente o ventrículo no dado instante.

Um elipsóide $\mathbf{e}(s)$ pode ser definido parametricamente em duas dimensões como o vetor:

$$\mathbf{e}(s) = (a\cos(s) + x_0, b\sin(s + \theta) + y_0) \quad (\text{I.1})$$

em que a e b são o tamanho dos semi-eixos do elipsóide, x_0 e y_0 são o centro do elipsóide na imagem e θ é a inclinação do elipsóide. A figura I.2 ilustra esta representação. Ajustando o elipsóide para alguns quadros da seqüência de imagens de IRM, os valores dos parâmetros da equação I.1 foram estimados (ver tabela I.1). Cabe lembrar que as coordenadas da imagem são invertidas em relação às abscissas.

Finalmente, os N vetores normais $\mathbf{n}(s)$ são calculados como sendo as normais do elipsóide avaliadas em N pontos igualmente espaçados. A normal do elipsóide pode ser calculado da transposta do vetor tangente $\mathbf{Tg}(s)$, que por sua vez é a derivada de $\mathbf{e}(s)$:

$$\mathbf{Tg}(s) = (-a\sin(s), b\cos(s + \theta) + b\theta) \quad (\text{I.2})$$

assim $\mathbf{n}(s)$ tornam-se:

$$\mathbf{n}(s) = (-\mathbf{Tg}_y, \mathbf{Tg}_x) = -(b\cos(s + \theta) + b\theta, -a\sin(s)) \quad (\text{I.3})$$

A inversão do vetor é simplesmente para fazer com que o vetor aponte para fora do elipsóide, por nossa definição. Por fim, as N normais relativas aos pontos de controle do contorno rastreado são avaliadas em espaços igualmente espaçados do parâmetro s , tal que $s = [0, \dots, 2\pi[$. Nota-se que a ordem dos pontos de controle deve ser a mesma do parâmetro s do elipsóide (ou seja, a distribuição é feita no sentido anti-horário).

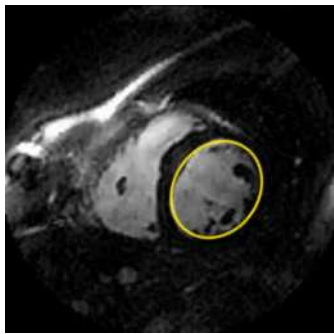


Figura I.1: O contorno do VE sendo representado por um elipsóide, cujos parâmetros são indicados na tabela I.1.

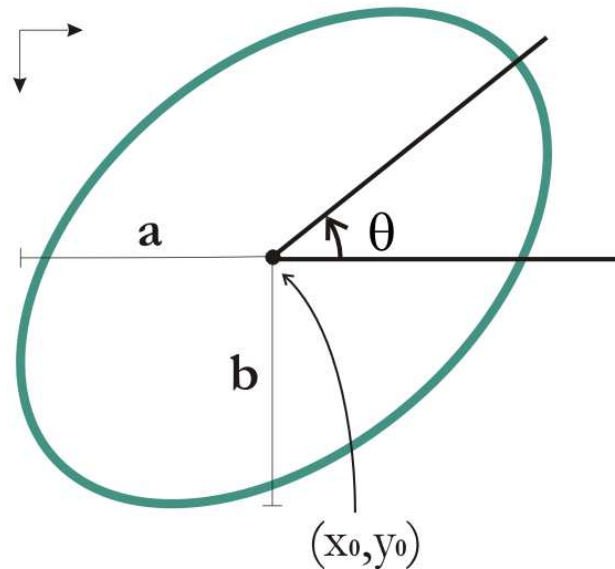


Figura I.2: Representação dos parâmetros do elipsóide de ajuste. Nota-se que as coordenadas dos eixos espaciais estão invertidas em relação ao eixo das abcissas.

Tabela I.1: Valores dos parâmetros normalizados do elipsóide que assemelha ao VE.

Parâmetro	Valor
a	2
b	3
x_0	153
y_0	140
θ	$-\pi/10$

Os resultados produzidos na figura 4.6 foram obtidos ao se usar valores de \mathbf{R} iguais à 1 ao lugar de 10, para reduzir a inércia do contorno representado durante a contração do coração, quando o contorno real se move rapidamente entre quadros. Nota-se também que mesmo melhorando a precisão na representação do contorno, ainda as imperfeições das imagens prejudicam bastante esta representação.

O algoritmo 'Star' funciona atualmente baseado na hipótese que o elipsóide que melhor representa um determinado quadro da seqüência possa ser usado para descrever as normais dos outros quadros da seqüência. Infelizmente tal hipótese apenas sustenta determinados casos. Uma proposta de melhoramento do algoritmo 'Star' é a estimação dos parâmetros a e b dinamicamente, baseado na frequência do batimento. Quando rastreando seqüências longas, o ponto central do elipsóide também é afetado pela respiração e deve ser melhor estimado. Ao fim, o ajuste de um elipsóide baseado nos pontos de controle pode ser feito para produzir resultados mais acessíveis.

II. DESCRIÇÃO DO CONTEÚDO DO CD

II.1 PASTAS DO CD

O conteúdo disponibilizado no CD está distribuído em quatro pastas:

- Documentos - onde encontra-se o relatório final de curso em PDF;
- Simulações - onde encontra-se o código usado nas simulações descritas, em linguagem C;
- Apresentação - as transparências e vídeos apresentados durante a defesa;
- Etc - a biblioteca OpenCV necessária para a execução das simulações.

A pasta 'Simulações' possui quatro sub-pastas: a pasta 'Dummy', a pasta 'Kalman', a pasta 'Condensation' e a pasta 'Star'. Cada uma contém as simulações relativas aos tópicos discutidos no capítulo 4.

A seguir tratamos da regulação dos parâmetros dos filtros estocásticos usados. As simulações estão munidas de *flags*, que servem para pausar a simulação em determinados pontos, ilustrando os passos executados no rastreamento e permitindo um melhor entendimento do funcionamento do mesmo.

De cada simulação, a sequência rastreada será salva em um arquivo *.avi* em caráter ilustrativo e uma nota sobre o tempo de cálculo no arquivo 'temps calcul.txt'.

SIMULAÇÃO DUMMY

Uma simulação de caráter ilustrativo, mostrando como se comporta o rastreamento quando não se tratam as incertezas.

SIMULAÇÃO USANDO O FILTRO DE KALMAN

No arquivo 'init.cfg', podem ser modificados os valores das incertezas das matrizes \mathbf{Q} , \mathbf{R} e \mathbf{P}_0 estipuladas anteriormente no texto, assim como os *flags* descritos acima. As coordenadas dos pontos de controle que iniciam a curva encontram-se no arquivo 'type.yml'. Dois conjuntos de pontos de controle estão disponíveis, com 11 e 7 pontos respectivamente. O arquivo padrão usa 11 pontos.

SIMULAÇÃO USANDO O FILTRO CONDENSATION

Como na simulação usando Kalman, as coordenadas dos pontos de controle que iniciam a curva encontram-se no arquivo 'type.yml'. O valor da variância de observação \mathbf{r} pode ser modificado no arquivo 'init.cfg', juntamente com o controle dos *flags*.

SIMULAÇÃO USANDO STAR

Como este algoritmo funciona baseado no filtro de Kalman, a regulação de seus parâmetros é bastante semelhante. Os parâmetros da elipsóide não são acessíveis pois foram regulados especificamente para a seqüência de teste presente.