

Medição de velocidade explorando campos consecutivos de um vídeo e detecção de contorno pelo método “Level Set”

Hermes A. Magalhães e Marcelo C. Ramalho

Discentes do PPGEE

Universidade Federal de Minas Gerais

Av. Antônio Carlos 6627, Belo Horizonte, MG 31270-010

hermesaguiar@terra.com.br, mccherem@yahoo.com.br

Hani C. Yehia e Antônio de Pádua Braga

Departamento de Engenharia Eletrônica

Universidade Federal de Minas Gerais

Av. Antônio Carlos 6627, Belo Horizonte, MG 31270-010

{hani, apbraga}@cpdee.ufmg.br

Abstract—This document was written with the purpose of providing a method for velocity measurement in image sequences. The displacement is measured with the analysis of two consecutive fields with the application of the Level Set method [5] and the energy function of Mumford and Shah [7].

Resumo—Este trabalho tem como objetivo apresentar e demonstrar um método de medição de velocidade a partir de um vídeo. A análise será feita nos campos consecutivos de um quadro do vídeo. Será utilizada uma ferramenta difundida na área de segmentação de imagens, a teoria dos Level Sets [5] com o funcional de energia de Mumford and Shah [7].

I. INTRODUÇÃO

Medição de velocidade de veículos é um importante ramo de estudo e pesquisa. Neste sentido, vários métodos já foram implementados. Existem no mercado muitos sensores que objetivam medição precisa de velocidade. Entre os principais estão os sensores indutivos, piezelétricos, laser e ultrassom. Os dois primeiros apresentam a desvantagem de a instalação necessitar de obras no asfalto, para a colocação dos sensores. A manutenção e a calibração também necessitam de interrupção da via. Os medidores de ultrassom e laser são menos invasivos na instalação e manutenção, mas apresentam um custo elevado.

Integrado com os equipamentos de medição, estão equipamentos de registro. Câmeras fotográficas ou de vídeo são acionadas quando os sensores determinam velocidades acima do permitido.

Um campo interessante nesta área é a utilização dos próprios equipamentos de registro para medição de velocidade. Com uma câmera de vídeo instalada, a medição e registro seriam feitos automaticamente a partir da sequência de imagens gerada. Esta proposta apresenta como vantagem um método não invasivo, uma vez que a instalação e manutenção não necessitam de interrupção da via. Além disto, há a diminuição do número de equipamentos, abaixando o custo do sistema.

Para este fim seria necessário a detecção e o rastreamento do veículo em quadros consecutivos do vídeo analisado. Na literatura são propostas quatro abordagens para aplicações em tráfego, descritas a seguir.

a) Baseada em modelos: A ênfase nestes estudos é a recuperação de trajetórias e desenvolvimento de modelos

com grande acurácia para um pequeno número de veículos [9]. O maior ponto fraco desta abordagem é que ela se baseia em modelos geométricos detalhados dos objetos. Não é realista desenvolver modelos geométricos para todos os veículos que surgem na via de tráfego [9], além de ser um método sensível à escala dos objetos. Nesta abordagem, diversos trabalhos foram propostos utilizando modelos 3D de objetos [14], [10], [11].

b) Baseada em regiões: Utilizam técnicas de subtração de imagem para segmentar o fundo do objeto em movimento. A partir da imagem subtraída, que fornece a informação do veículo, técnicas de correlação cruzada rastreiam o objeto pelos quadros consecutivos. Para compensar efeitos de iluminação e condições climáticas, filtros adaptativos são utilizados para estimar o fundo da imagem [12], [13]. Para tráfego intenso, esta solução pode mesclar veículos diferentes em uma mesma região devido à oclusão parcial.

c) Baseada em contornos ativos: Os contornos ativos são utilizados na literatura para segmentação de imagens, com diferentes objetivos. A função objetivo pode, por exemplo, expressar a solução do contorno com maior gradiente [3], ou a separação de regiões homogêneas da imagem [6]. Paragios [4] propôs uma técnica para o rastreamento de objetos em movimento através da análise de dois quadros consecutivos. Esta solução foi capaz de rastrear múltiplos objetos em cenas complexas, além de rastrear objetos não rígidos. Os problemas de oclusão e sensibilidade à iluminação permanecem [9], mas é possível a detecção de objetos em diferentes escalas [4].

d) Baseada em características: Outra abordagem para o rastreamento é a identificação de sub-características dos veículos, na qual abandona-se a idéia de rastrear o veículo como um todo, procurando rastrear partes discerníveis dele. A vantagem é ajuste automático da técnica a variações da iluminação ambiente e a oclusão parcial pela seleção das características favoráveis [9]. Mas o agrupamento de características pertencentes ao mesmo objeto é um problema de difícil solução [15]. Em uma abordagem alternativa, Garibotto et al. [8] propôs uma técnica para medição de velocidade baseada no reconhecimento e rastreamento das placas dos veículos.

A proposta deste trabalho é a de investigar de forma

preliminar o desempenho e as características da ferramenta de contornos ativos para a detecção dos objetos em movimento na cena. As imagens utilizadas não apresentam grande complexidade como as utilizadas por Paragios [4]. Por esta razão, a função objetivo utilizada foi baseada no funcional de energia de Mumford and Shah [7], que separa regiões homogêneas da imagem. O objetivo não é a completa solução do problema de medição de velocidade, mas a familiarização com o método de contornos ativos implementado através da teoria dos “Level Sets”, com suas variáveis, com possíveis problemas e com a complexidade computacional envolvida. Os resultados obtidos a seguir contribuem para futuras aplicações em cenários mais complexos.

II. METODOLOGIA

Nesta seção será apresentada, de forma abreviada, a teoria sobre as ferramentas utilizadas, detalhes sobre a implementação e o método de medição de velocidade. Foi utilizado o ambiente de processamento MATLAB. O trabalho proposto utilizou ferramentas do toolbox de processamento de imagens da mathworks e o toolbox de Level Set, implementado por Melvin [1].

A. Teoria “Level Set”

A teoria dos “Level Sets” [5] consiste em detectar o contorno de uma imagem a partir da adaptação de uma superfície ϕ , formulada no plano da imagem (nível $\phi = 0$ da superfície), conforme Fig. 1. Com esta teoria, mudanças de topologias são facilmente tratadas [4].

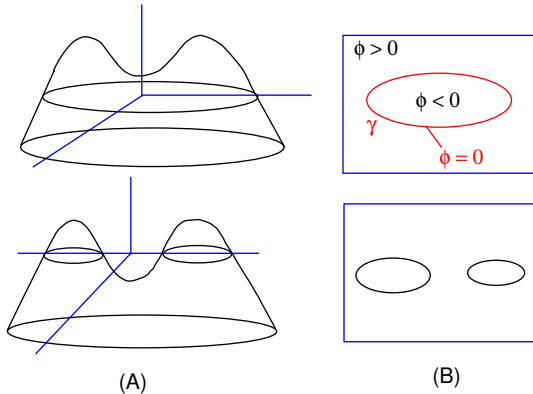


Fig. 1. Superfície formada sobre o plano da imagem. A coluna (A) mostra a adaptação da superfície e a coluna (B) demonstra o resultado no plano da imagem, com pouca mudança na superfície, conseguem-se grandes modificações topológicas na curva no plano da imagem.

A formulação consiste em adaptar a superfície a fim de minimizar uma função objetivo definida. Como mencionado na introdução, existem várias formulações diferentes para a função objetivo [3], [6], [4].

O funcional de energia utilizado neste trabalho foi proposto por Mumford and Shah [7]. A Fig. 1B apresenta uma representação do plano da imagem, com os respectivos valores da função ϕ , dentro e fora da curva no nível zero. Acompanhe a formulação a seguir tomando como base esta figura.

$$\Psi(\phi) = \int_{x,y \in \phi > 0} (I - M1(\phi))^2 H(\phi) + \int_{x,y \in \phi < 0} (I - M2(\phi))^2 (1 - H(\phi)) + \mu \int_{x,y \in \phi = 0} |\nabla H(\phi)| \quad (1)$$

na qual:

- H – Função degrau em $\phi = 0$;
- $M1$ – Média dos valores dos pixels dentro da curva γ ;
- $M2$ – Média dos valores dos pixels fora da curva γ ;
- I – Valores dos pixels da imagem;
- μ – Ajuste do Comprimento;
- x, y – Coordenadas dos pixels da imagem;
- $\int_{\phi=0} |\nabla H(\phi)|$ – Comprimento da curva γ .

Minimizando-se a superfície conforme a função objetivo em (1), consegue-se a curva com menor comprimento possível que mantenha as imagens formadas, dentro e fora da curva, homogêneas. Os dois primeiros termos são relativos ao caráter homogêneo das imagens, respectivamente. O terceiro termo é relativo ao comprimento da curva.

A taxa de atualização da função ϕ tem que fazer a função Ψ caminhar na direção contrária ao gradiente $\nabla \Psi$. Com isto:

$$\frac{\partial}{\partial t} \phi = \delta_\epsilon(\phi) \left[\mu \operatorname{div} \frac{\nabla \phi}{|\nabla \phi|} - (I - M1)^2 + (I - M2)^2 \right] \quad (2)$$

na qual: δ_ϵ – Delta de Dirac aproximado com largura ϵ .

O parâmetro δ_ϵ diminui a complexidade da atualização da função ϕ . Somente valores perto da curva γ ($\phi = 0$) são testados e atualizados. O valor de ϵ é um parâmetro do algoritmo, juntamente com o passo temporal ∂t e o fator de ajuste de comprimento μ .

B. Aplicação em Medição de Velocidade

1) *Varredura Entrelaçada*: A varredura de vídeo entrelaçada, assim como sua frequência de percurso vertical, escolhida originalmente como função da frequência da rede de energia elétrica, foram definidas para evitar cintilação (efeito flicker) [2]. Tais escolhas foram em função da limitação de banda de frequência disponível e alto custo dos circuitos osciladores na época da criação dos primeiros padrões de transmissão de sinais de vídeo. O padrão consiste em transmitir alternadamente linhas complementares pares e ímpares (denominadas campos de um quadro de vídeo), que para o caso do sinal NTSC usado neste trabalho repetem-se a uma taxa de 60 campos por segundo (i.e. 16,7ms entre campos).

2) *Medição de Velocidade - efeito “pente”*: Aplicando-se a detecção de borda do objeto em movimento em dois campos consecutivos de um quadro de vídeo, consegue-se medir a sua velocidade. Observe a Fig. 2. Nela podemos observar o efeito “pente” (ou “comb-effect”) causado pelo deslocamento do objeto entre dois quadros sucessivos de linhas pares e ímpares. Separando as linhas pares das ímpares obtemos as imagens da direita na mesma Fig. 2, onde se vê claramente o objeto e seu deslocamento ocorrido no intervalo de 1/60 segundos.

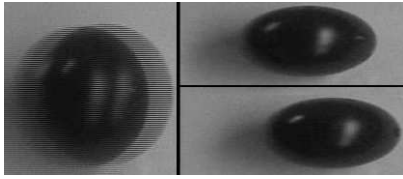


Fig. 2. Exemplo de um quadro. Os dois campos são separados para a detecção do contorno.

A primeira etapa na implementação do método de curvas de nível ou “Level Set” é a definição de uma superfície ϕ . A superfície adotada neste trabalho é uma hiper-elipsóide, obtida modificando-se a proporção entre os eixos de um parabolóide para tornarem-se proporcionais à Região de Interesse (ROI - “Region Of Interest”), onde encontram-se os objetos em movimento. Para determinação da ROI que englobe apenas o objeto em movimento, rejeitando a influência de imagens de fundo como objetos imóveis, texturas, etc., executa-se o cálculo da diferença (em valor absoluto) entre os campos consecutivo par e ímpar da mesma imagem de vídeo, conforme mostrado na Fig. 3. Assume-se aqui que as condições de iluminação não irão se modificar significativamente para os objetos de fundo no período de 16,67ms, o que é perfeitamente razoável. A seguir, a ROI é determinada estabelecendo-se um limiar de interesse em porcentagem da intensidade máxima na imagem diferença. Esta ROI é mostrada como o retângulo interno plotado sobre o objeto na imagem diferença da Fig. 3. Utilizando-se esta ROI prévia determina-se uma ROI definitiva em seu redor (retângulo externo na Fig. 3, com dimensões maiores em torno de 20%) capaz de compensar efeitos de sombra e menor contraste (efeito visível no lado esquerdo do objeto neste exemplo). Isto porque foi observado experimentalmente que uma redução simples no limiar de interesse tem o efeito de aumentar excessivamente a ROI, de forma a abarcar toda a região de sombra, o que aumenta excessivamente o tempo de processamento, devido à instabilidade da curva de bordas nas regiões de sombra como veremos mais adiante, sem a devida compensação sob a forma de melhores resultados. O nível zero da hiper-elipsóide também está mostrado, plotado sobre os campos par e ímpar na Fig. 3.

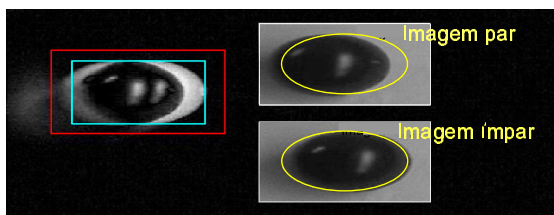


Fig. 3. Região de Interesse (ROI - “Region Of Interest”) e nível zero da hiper-elipsóide inicial ϕ (seqüência “bolapreta3”).

III. RESULTADOS

Para avaliação do método, diversas seqüências de vídeo de objetos em movimento de diferentes formas, cores (que resultam em condições de diferentes contrastes quando transformadas para tons de cinza) e velocidades foram filmadas sobre fundo homogêneo, utilizando-se uma filmadora

digital de 1 CCD com varredura entrelaçada, formato de vídeo DVCPRO de 25Mbps e “aspect ratio” de 3/4. O sinal de vídeo foi transferido via porta digital IEEE1394 (“FireWire”) para um PC com CPU Pentium IV de 1,4GHz, 256MB de RAM e Hard Disk SCSI ultra DMA 160, sendo o material de vídeo posteriormente compilado por edição não linear utilizando o software Studio DV da Pinnacle. Foram selecionadas seqüências de vídeo significativas de situações diversas como mudanças na iluminação e velocidade de abertura do “shutter”.

A partir da observação da evolução do objeto quadro a quadro no campo de visão da câmera, foi feita uma contagem da quantidade de quadros (separados no tempo de 1/30 segundos) necessária para que o objeto percorresse toda extensão horizontal do campo de visão da câmera (neste caso fixado em 0,5 metros). Devido à rapidez do deslocamento dos objetos frente à câmera com relação à taxa de varredura de vídeo, esta forma de medição quadro a quadro mostrou-se sujeita a uma margem de erro de ± 1 quadro, colocada na última coluna da Tabela 1 sob a forma de \pm m/s e porcentagem. A medida de velocidade quadro a quadro, aqui usada como medida de controle foi feita para todas as seqüências selecionadas.

Seu resultado foi colocado na coluna “Medidas na direção horizontal” da Tabela 1 para posterior comparação com o método aqui proposto, mostrado nas colunas “Velocidade calculada” na mesma tabela. Para cada seqüência de vídeo selecionada, foi pinçado um quadro de vídeo intermediário (com todo o objeto visível) de dimensão 640×480 pixels, que por sua vez passou pelo processamento aqui proposto para cálculo da velocidade instantânea. Estes quadros intermediários estão mostrados na Fig. 6, com as respectivas bordas detectadas e linhas de fronteira horizontais e verticais, i. e., coordenadas máximo de borda, média dos pixels pertencentes à borda, e mínimo de borda, evidenciando de forma gráfica o desempenho relatado na Tabela 1 para o algoritmo. Note como a sombra dos objetos (sempre à esquerda nos exemplos mostrados) degrada a medida de velocidade: enquanto o cálculo do deslocamento feito pela coordenada dos máximos (no sentido da direita, ou seja, lado sem sombra) nos dá um índice de acerto de 91,3%, o deslocamento medido utilizando-se a coordenada mínima (no sentido da esquerda) apresenta um índice de acertos de apenas 39,1%, já desprezados aqueles casos em que o valor de velocidade é nulo, i. e., a sombra do objeto está parcialmente fora da ROI. O índice de acerto pela média das coordenadas dos pixels pertencentes à borda também é afetado pela instabilidade das curvas de borda nas sombras, resultando em um índice de acerto intermediário de 52,2%.

Observou-se durante o experimento que a correta escolha da região de interesse (ou ROI - Region Of Interest) é de suma importância para acelerar a convergência do algoritmo “Level Set”, levando a melhores resultados com menor custo computacional. A Tabela 1 traz a informação de quantidade de iterações necessárias para se alcançar resultados dentro da faixa de tolerância, bem como o tempo médio em segundos

por iteração, que é função do tamanho da ROI. Observe que 76,2% das seqüências convergiram em menos de 5 iterações, totalizando 15 segundos de tempo de processamento. Aplicando-se o algoritmo “Level Set” sobre toda a imagem, ou seja, sem a seleção de uma ROI, cada iteração levou em média 8 segundos, e nenhuma seqüência convergiu em menos de 40 iterações.

O processamento pelo método “Level Set” requereu o ajuste de uma série de parâmetros, como a definição da própria função ϕ , a largura da função delta aproximada δ_ϵ , o valor do passo temporal numérico Δt aproximador de ∂t e o fator de decremento μ do comprimento total da curva de nível γ no nível $\phi = 0$. Os resultados mostrados aqui foram alcançados com todos estes parâmetros fixos, com exceção do passo temporal Δt , que foi sucessivamente incrementado a cada nova iteração de um “fator multiplicador de passo temporal”, i.e., a cada iteração $k + 1$, $\Delta t_{k+1} = Fator \cdot \Delta t_k$. Este artifício foi implementado porque se observou uma diminuição gradativa na rapidez com que a curva de nível zero do hiper-elipsóide se aproximava da borda do objeto, e também porque o critério de parada era baseado no número de iterações, limitando assim o valor final de Δt . A utilização do fator multiplicativo resultou em uma gradação mais “linear” do deslocamento da curva γ em direção à borda do objeto. Os fatores utilizados para as seqüências estão descritos na Tabela 1.

Quando o objeto em movimento apresenta um bom contraste com relação à imagem parada de fundo, o limiar de interesse é bem seletivo na determinação de uma ROI correta, como podemos ver na Fig. 4. Neste caso, a área de interesse claramente selecionou a região que envolve tanto o objeto quanto a sua sombra. A convergência em casos como este foi rápida e as bordas foram corretamente identificadas (veja na Fig. 6 as seqüências que convergiram em 5 iterações).

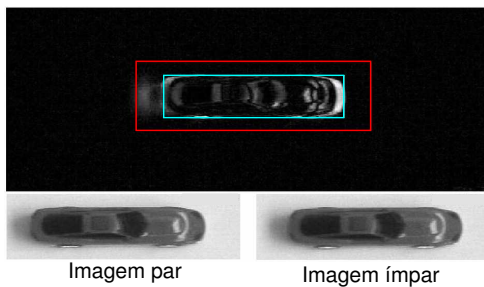


Fig. 4. Imagem com bom contraste e definição: diferença entre campos e seleção da ROI para um limiar de interesse em 50%.

No entanto fatores como iluminação, velocidade do “shutter”, a própria cor e o índice de reflexão dos objetos podem influenciar negativamente na correta seleção da ROI ou a convergência na detecção das bordas do objeto. A Fig. 5 mostra como a diferença na velocidade de abertura do “shutter” altera a nitidez da imagem, o que dificulta a detecção das bordas pelo algoritmo “Level Set”.

A Fig. 7 mostra como o brilho excessivo associado ao baixo contraste “polariza” a seleção da ROI para a seqüência “bolaazulbrilhante1”, ao utilizarmos o mesmo limiar de

interesse usado nas seqüências mostradas na Fig. 4.

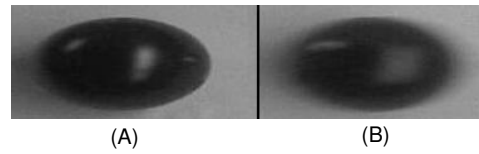


Fig. 5. Comparação entre diferentes velocidades de exposição. Em (A): exposição rápida. Em (B): exposição lenta.

Note que a seleção de ROI com este limiar despreza informação relevante das verdadeiras bordas do objeto, fazendo com que o algoritmo “Level Set” busque as bordas das regiões de alto brilho, naturalmente mais difusas. Note para o exemplo mostrado que no alto e à direita da imagem par foram detectados trechos das verdadeiras bordas que não são visíveis na imagem ímpar (devido ao deslocamento do objeto para a direita), o que levou a uma medida incorreta de velocidade.

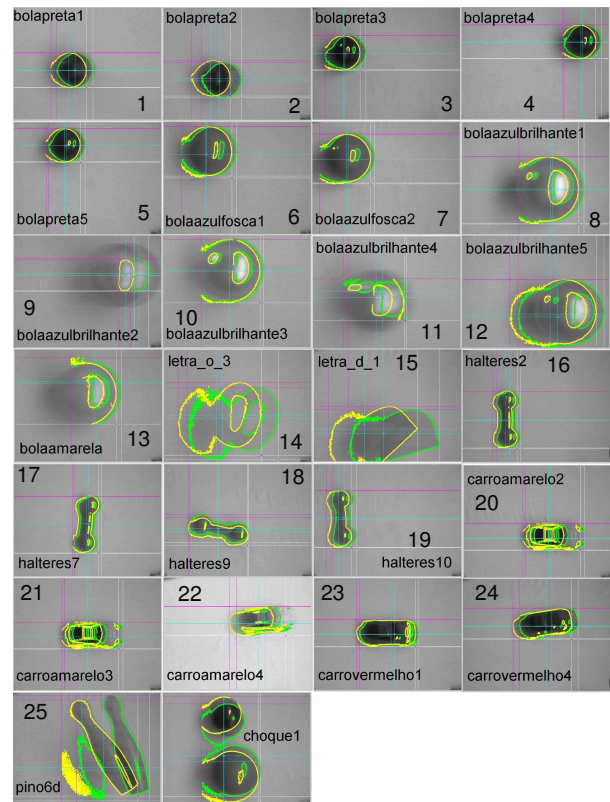


Fig. 6. Resultado gráfico para as seqüências constantes da Tabela 1 e da seqüência “choque1” que envolve dois objetos na cena.

Para um limiar de interesse de 30%, i. e., para uma fronteira menor de intensidade dos pixels, abaixo da qual a imagem diferença é considerada irrelevante para o processamento, a seleção da ROI e a detecção das bordas se dão de forma correta, como pode ser visto na segunda imagem diferença da Fig. 7. A cor do objeto neste caso causou também um resultado nítido de menor contraste, assim como em outras seqüências como “bolaazulbrilhante” de 3 a 5, “Bolaamarela”, “letra-O-3” e “letra-D-1”. Note que para estas seqüências a convergência do “Level Set” foi mais demorada. Curiosamente, a seqüência “bolaazulbrilhante2”,

TABELA I

RESULTADO DE MEDIÇÕES DE VELOCIDADE POR CAMPOS
CONSECUTIVOS DE UM QUADRO DE VÍDEO EMPREGANDO DETECÇÃO
DE BORDA PELO MÉTODO “LEVEL SET”.

Imagem Denominação (Fig. 6)	Velocidade calculada (m/s)			Custo computacional		Medidas na direção horizontal
	Componente X (horizontal)			Tempo médio por iteração (s) por campo	Quant. de iterações	Velocidade p/ comparação m/s \pm m/s (%) (ref. a ± 1 quadro)
	Média	Máximo	Mínimo			
1	1,24	1,35	1,16	0,43	5	1,25 \pm 0,21 (17)
2	1,82	2,22	1,99	0,56	5	2,14 \pm 0,62 (29)
3	0,29	1,21	0,00	0,54	5	0,94 \pm 0,12 (13)
4	0,27	1,08	0,21	0,50	5	1,15 \pm 0,18 (15)
5	0,33	0,96	0,51	0,37	5	0,94 \pm 0,12 (13)
6	0,30	0,96	0,00	1,13	5	1,00 \pm 0,13 (13)
7	0,38	0,82	0,00	0,87	5	0,88 \pm 0,10 (12)
8*	0,42	1,13	0,00	1,41	10	1,07 \pm 0,15 (14)
9	2,67	2,97	2,32	0,34	5	1,88 \pm 0,95 (51)
10*	0,04	0,66	0,00	1,79	10	0,68 \pm 0,06 (9)
11	-1,40	0,33	-0,97	1,02	10	2,14 \pm 0,62 (29)
12*	1,45	1,58	1,72	3,05	10	1,67 \pm 0,37 (22)
13**	0,58	1,00	0,00	0,80	20	1,00 \pm 0,13 (13)
14	1,29	3,39	1,94	3,97	10	3,00 \pm 1,21 (40)
15	3,91	4,54	4,05	4,43	10	3,75 \pm 1,90 (51)
16	0,70	0,72	0,84	0,48	5	0,68 \pm 0,06 (9)
17	0,42	0,52	0,35	0,47	5	0,48 \pm 0,03 (6)
18	0,20	0,14	0,21	1,09	5	0,34 \pm 0,02 (6)
19	1,15	1,03	1,20	0,40	5	1,00 \pm 0,13 (13)
20	0,50	0,82	0,57	0,76	5	0,94 \pm 0,12 (13)
21	0,61	0,83	0,53	1,01	5	0,83 \pm 0,09 (11)
22	1,44	1,41	1,47	1,38	5	1,50 \pm 0,30 (20)
23	0,69	0,57	0,62	0,77	5	0,54 \pm 0,04 (7)
24	1,18	1,31	1,34	1,18	5	1,36 \pm 0,25 (18)
25**	4,40	2,35	3,03	6,16	20	3,75 \pm 1,90 (51)

Contagem na componente X pelos máximos: 91,3% de acerto
Contagem na componente X pelas médias: 52,2% de acerto
Contagem na componente X pelos mínimos: 39,1% de acerto
Multiplicador do passo temporal: 1,2 (exceto em **: 1,4)
Limiar de interesse: 0,5 (exceto em *: 0,3)
Custo computacional: 76,2% das seqüências convergiram em 5 iterações, totalizando 15s de processamento.

mesmo tendo selecionado uma região de interesse em torno apenas do brilho do objeto, conseguiu com este brilho calcular corretamente a velocidade do mesmo. Como o objeto é uma bola, a imagem do brilho manteve-se homogênea. Isto aliado ao fato de a ROI não ter envolvido as bordas reais do objeto proporcionou o bom resultado alcançado, mesmo com um limiar de interesse de 50%.

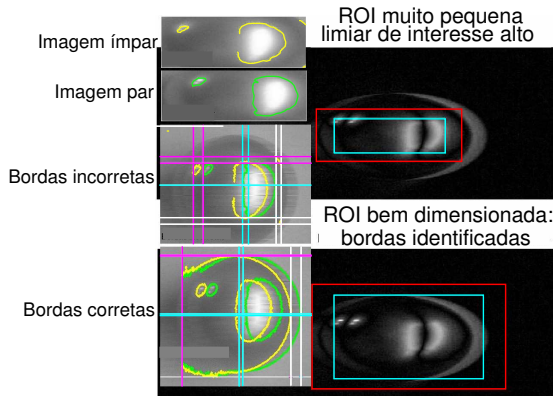


Fig. 7. O limiar de interesse deve estar em função do contraste da imagem: limiares de interesse alto (50%) e baixo (30%) e sua influência na determinação da ROI (seqüência “bolaazulbrilhante1”).

Vale ressaltar o alto tempo de convergência observado na Tabela 1 para a seqüência “Pino6d”, de aproximadamente 6 segundos por iteração. Esta duração é função do grande tamanho da região de interesse selecionada, mostrada na Fig. 8. Se tivéssemos selecionado regiões de interesse cujas bordas não necessariamente fossem paralelas às bordas do quadro de imagem, talvez pudéssemos reduzir a ROI para o entorno dos objetos, reduzindo o tempo de processamento.

Isto aliado ao estabelecimento de uma hiper-elipsóide de eixos igualmente oblíquos com certeza levariam a uma redução no custo computacional, pois as curvas de bordas convergiriam em menos iterações.

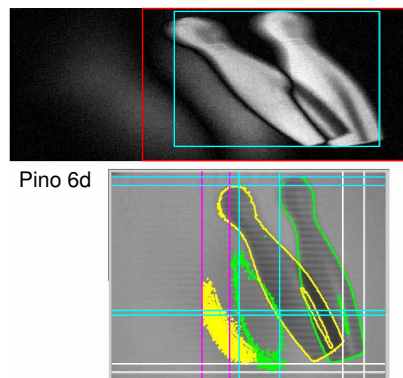


Fig. 8. ROI com bordas horizontal e vertical e hiper-elipsóides iniciais com eixos igualmente horizontal e vertical levam à seleção de ROI com área maior que o necessário e excessivo tempo de processamento (seqüência “pino6d”).

IV. CONCLUSÃO

Neste trabalho, o processamento para detecção dos contornos pelo “Level Set” se fez sobre a ROI da imagem original que havia sido filmada sobre um fundo homogêneo. Esta característica proporciona uma imagem muito simplificada em relação às condições reais, podendo ser resolvida por métodos mais simples (como cálculo do gradiente da imagem). No caso de um fundo contendo objetos parados ou texturas que estejam de alguma forma aparentes na ROI selecionada, com certeza o contorno destes objetos será evidenciada pelo “Level Set” implementado, inviabilizando a medição de velocidade do objeto de interesse. Para que isto não ocorra, uma sugestão é trabalhar com duas imagens diferença consecutivas, formadas por três campos seguidos (par-ímpar-par ou ímpar-par-ímpar) de dois quadros sucessivos. Outra saída é utilizar a função objetivo desenvolvido por Paragios [4]. Deste modo, apenas objetos em movimentos serão considerados para cálculo dos contornos. Contudo, a aplicação do método nas imagens com fundo homogêneo destacou problemas que serão encontrados no desenvolvimento de soluções gerais e robustas. A presença da sombra, que se move com o objeto, mas de uma forma não rígida, será um problema a ser solucionado. As condições de iluminação e contraste é outro fator importante, uma vez que a análise da diferença entre dois quadros (utilizados por muitos métodos) depende de um deslocamento que provoque uma não uniformidade nas regiões [4].

Para a aplicação de vias públicas reais, alguns problemas serão encontrados. Os parâmetros utilizados devem ser adaptados de forma automática e a correta calibração da câmera é um fator essencial. Para a correta transformação de pixels/s em m/s é necessário uma correspondência entre o plano da via pública e o plano da imagem. Pelo ângulo da câmera utilizado no trabalho (normal à via), uma solução seria a adoção de marcas com espaçamento conhecido no

asfalto. A correspondência entre pixels e metros poderia ser retirada pela análise direta.

Com relação aos parâmetros, observou-se que a variação dinâmica do passo temporal de iteração Δt apresentou evidente melhoria na taxa de convergência da curva γ . Isto suscita a necessidade de se definir no futuro uma regra clara de variação de Δt , que seja dinamicamente atrelada a medidas de taxa de convergência extraídas do próprio algoritmo, como o gradiente ou mesmo as medidas sucessivas de homogeneidade $M1(\phi)$ e $M2(\phi)$ das imagens dentro e fora da curva γ . A sensibilidade observada da seleção de uma ROI pelo limiar de interesse também sugere que uma medida de limiar mais adequada possa ser extraída automaticamente da própria imagem, através de cálculos que meçam seu contraste. Uma sugestão seria o achatamento da função densidade de probabilidade (ou variância) da intensidade dos pixels da imagem diferença.

Uma possível abordagem para a solução da influência negativa das sombras dos objetos sobre a correta determinação de seu contorno pode ser percebida se notarmos a diferença entre a topologia assumida pela curva γ em contornos reais (pontos bem comportados e em linha suave) e a topologia assumida nas sombras (pontos dispersos no espaço). Um classificador, talvez baseado em redes neurais, pode ser usado para classificar os pontos da curva γ em pertencentes ou não à borda do objeto, descartando assim da medida de velocidade pontos que com certeza não irão contribuir para sua correta determinação.

Apesar dos problemas encontrados, foi alcançada uma taxa de acerto considerável de 91,3% se for considerado o cálculo pela coordenada de maior valor (lado oposto à sombra do objeto). Esta confiança nos permitiu aplicar este algoritmo com sucesso, por exemplo, na seqüência de desaceleração do objeto “Letra-O” mostrado na Fig. 9, cuja curva de velocidade instantânea apresentada foi calculada sempre pelo valor máximo da coordenada horizontal. Para todos os quadros as curvas de nível convergiram em até 5 iterações de 1 segundo cada. Considerando que o MATLAB tem reconhecidamente um tempo de cálculo muito superior ao alcançável por uma implementação por programação direta, esta técnica mostra-se promissora para aplicações em tempo real.

Trabalhos Futuros

Dentre os possíveis trabalhos está a análise do método em imagens complexas e a possível utilização de diferentes funções objetivos, como a de Paragios [4]. Outras abordagens, como a adoção de rastreamento de características, também deverão ser implementadas para uma comparação de desempenho e eficiência computacional.

Para resolver a interferência da iluminação nos objetos, uma possibilidade seria classificar as curvas de contorno como pertencentes a sombras, reflexos e partes do objeto usando medidas de comprimento, dispersão no espaço, intensidade, comportamento físico de um quadro para outro, etc. A partir desta classificação, a medida de velocidade

passaria a considerar apenas curvas de contorno pertencentes a partes do objeto, reduzindo a margem de erro.

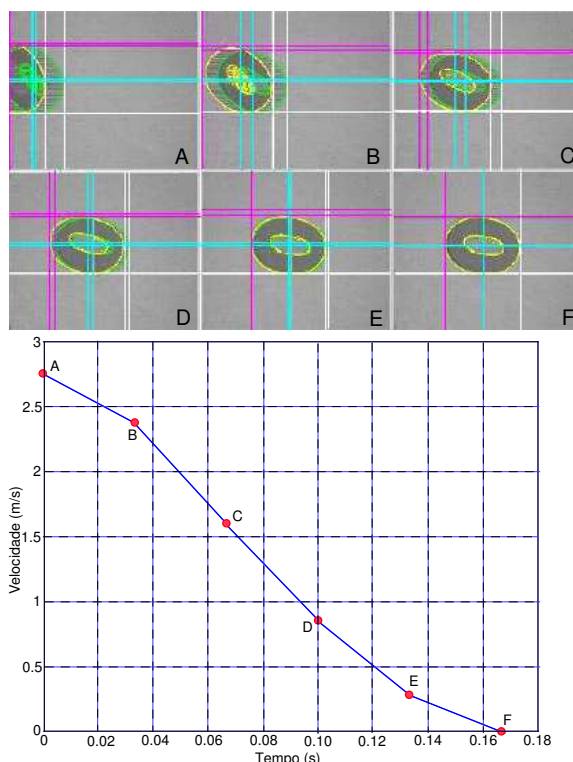


Fig. 9. Seqüência do objeto “Letra-O” em desaceleração e a velocidade instantânea quadro a quadro utilizando o método proposto.

REFERENCIAS BIBLIOGRÁFICAS

- [1] M. M. Michael Melvin, *Level Set Toolbox*, [http://www.mpcs.ryerson.ca/~rpyke/imgproc/](http://www.mpcs.ryerson.ca/~mmelvin@scs.ryerson.ca/http://www.mpcs.ryerson.ca/~rpyke/imgproc/).
- [2] N. P. Paragios, “Geodesic Active Regions and Level Set Methods: Contributions and Applications in Artificial Vision,” *Ph.D. Dissertation*, INRIA Sophia Antipolis, Jan 00, France.
- [3] R. G. Goldenberg, R. K. Kimmel, E. R. Rivlin, M. R. Rudzsky, “Fast Geodesic Active Contours,” *IEEE Transactions on Image Processing*, Vol. 10, No. 10, October 2001.
- [4] N. P. Paragios, R. D. Deriche, “Geodesic Active Contours and Level Sets for the Detection and Tracking of Moving Objects,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 22, No. 3, March 2000.
- [5] S. O. Osher, J. S. Sethian, “Fronts propagating with curvature-dependent speed: Algorithms based on the Hamilton-Jacobi formulation,” *Journal of Computational Physics*, 1988.
- [6] V. C. Caselles, R. K. Kimmel, G. S.apiro “Geodesic Active Contours,” *International Journal of Computer Vision*.
- [7] L. V. Vese, T. C. Chan, “A Multiphase Level Set Framework for Image Segmentation using the Mumford and Shah Model,” *UCLA C.A.M. Report 01-25*, Sept. 2001.
- [8] G. Garibotto, P. Castello, E. Del Ninno, P. Pedrazzi, G. Zan. “Speedvision: speed measurement by license plate reading and tracking,” *Intelligent Transportation Systems, 2001. Proceedings*, Sept. 2001.
- [9] H. A. Magalhães. “Classificação de veículos por fusão da análise de perfil magnético de sensores a laço indutivo com características visuais obtidas de câmeras de vídeo,” Texto de qualificação submetido ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Minas Gerais, Aug. 2005.
- [10] K. Baker, G. Sullivan. “Performance assessment of model-based tracking,” *Proceedings of the IEEE Workshop on Applications of Computer Vision*, 1992.
- [11] G. Sullivan. “Visual Interpretation of known objects in constrained scenes,” *Proceedings Phil. Trans. Roy. Soc.*, 1992.

- [12] G. Sullivan. "Moving object recognition using an adaptive background memory;" *Proceedings Cappellini, V. (Ed.), Time Varying Image Processing and Moving Objects Recognition*, 1990.
- [13] M. Kilger. "A shadow handler in a video-based real-time traffic monitoring system," *IEEE Workshop on Applications of Computer Vision*, 1992.
- [14] D. Koller, J. Weber, J. Malik. "Robust multiple car tracking with occlusion reasoning;" *ECCV*, 1994.
- [15] B. Coifman, D. Beymer, P. McLauchlan, J. Malik. "A real-time computer vision system for vehicle tracking and traffic surveillance;" *Proceedings of the 1996 ITE annual Conference*, 1996.