

# Desenvolvimento de rastreamento de máscaras em vídeo para simulador de treinamento

## *Development of video masking tracking for training*

Antonio Valerio Netto<sup>1</sup>

<sup>1</sup> Centro das Indústrias do Estado de São Paulo, CIESP, Brasil.  
avnetto@hotmail.com

**Abstract.** *The purpose of this article is to report the development of the image processing algorithms present in the Editing module of an interactive multimedia training system for public and private security area. This module aims to set up a scene that allows the user to interact with the video. Basically, from a main video, masks are inserted to trace the objects of interest so that at the moment of interaction with these objects the scene is taken to another situation (scene). Each mask is a closed outline that contains information about which secondary video should be displayed if there is interaction at any point within this mask.*

**Resumo.** O objetivo deste artigo é relatar o desenvolvimento dos algoritmos de processamento de imagens presentes no módulo de Edição de um sistema de multimídia interativo de treinamento para área de segurança pública e privada. Esse módulo tem como objetivo a montagem de uma cena que permita a interação do usuário com o vídeo. Basicamente, a partir de um vídeo principal, são inseridas máscaras para rastreamento dos objetos de interesse para que no momento da interação com estes objetos a cena seja conduzida para outra situação (cena). Cada máscara é um contorno fechado que contém informação sobre qual vídeo secundário deverá ser exibido caso haja interação em qualquer ponto dentro desta máscara.

### I. INTRODUÇÃO

O projeto o qual o artigo está relacionado teve como meta desenvolver um sistema de multimídia interativo de treinamento na área de segurança pública e privada. O propósito foi criar um sistema que pudesse “transportar” os profissionais da área de segurança para uma realidade mais próxima do cotidiano de trabalho, permitindo que os mesmos tenham um treinamento continuado e uma formação mais adequada. Muitas vezes, esses profissionais são colocados em situações de treinamento de tiro onde os alvos a serem alvejados são fixos ou imóveis, não transmitindo a sensação de realidade com a qual esse profissional irá realmente se defrontar no momento em que estiver prestando seus serviços no dia-a-dia. O projeto busca gerar uma evolução nos equipamentos de treinamento levando maior interatividade e sentimento tanto de imersão quanto de envolvimento.

Em linhas gerais, o funcionamento do simulador se baseia na exibição, em um anteparo de metal, de um filme 2D com situações reais do cotidiano do agente de

segurança, tais como assaltos, invasões entre outros, utilizando personagens em tamanho real (1:1). Isto produz por meio das imagens e dos sons emitidos, uma sensação de presença no momento em que a situação está ocorrendo. Para realizar a interação com a cena é utilizada uma arma real carregada com projéteis de silicone ou óleo de mamona. O simulador possui um sistema de captura de sensores que identifica a posição que este disparo do tiro atingiu a imagem projetada por um projetor de 3500 lux em um anteparo de metal. O agente terá que tomar decisões, tais como o momento exato de sacar a arma e atirar; e posteriormente a sua ação, ocorrerá uma reação do filme de acordo com o resultado do disparo. Isto é, após efetuar o disparo, o sistema será responsável por definir qual a nova sequência de imagens a ser mostrada na tela, dependendo da ação tomada pelo agente.

Um dos pontos inovadores desse projeto está nos algoritmos computacionais da área de processamento de imagens utilizados para geração das máscaras automáticas que identificam as regiões fatais e não fatais dos personagens no decorrer do filme 2D que está sendo exibido. Além disso, existe um desafio tecnológico na área de multimídia interativa relacionado à realização das sequências (ramificação do filme 2D) sem que ocorra cortes abruptos, isto é, a passagem do filme principal para uma sequência determinada, deve ser realizada sem que ocorra a perda do contexto da cena. O *software* que realiza todas estas tarefas se comunica com o *hardware* do simulador a partir da interação com o agente de segurança. Toda a interação e gerenciamento de informações são controlados por meio desse *software*, o qual é dividido em dois módulos: Edição e Controle.

As atividades apresentadas neste artigo fazem parte do desenvolvimento dos algoritmos de processamento de imagens presentes no módulo de Edição. Esse módulo tem como objetivo a montagem de uma cena que permita a interação do usuário com o vídeo. Basicamente, a partir de um vídeo principal, são inseridas máscaras para rastreamento dos objetos de interesse para que no momento da interação com estes objetos, a cena seja conduzida para outra situação (cena). Cada máscara é um contorno fechado que contém informação sobre qual vídeo secundário deverá ser exibido caso haja interação em qualquer ponto dentro desta máscara. As informações sobre as máscaras e ramificações são armazenadas em um arquivo com extensão \*.VMP. O conjunto dos vídeos (principal e continuações) e o arquivo contendo as informações sobre as máscaras e ramificações (relações entre máscaras e vídeos) são chamados de cena.

Os algoritmos para rastreamento de pessoas têm despertado interesse, seja para o caso de vídeos de segurança, com foco no controle de fluxo de pessoas, de investigação ou estudos comportamentais. Em geral, a maioria dos métodos para detecção de movimento em sequências de imagens envolve a subtração de sucessivos *frames* [2] [3] [4]. No caso do rastreamento de máscaras em vídeo foram estudados diferentes linhas de pesquisas. Existem métodos que envolvem abordagens estatísticas de características da imagem [1]; outros que relacionam a deformação do contorno com princípios baseados na conservação e teoria Hamilton-Jacobi, encontrados no método *Level-Set* [8]; também foi estudado o método que utiliza a abordagem de fluxo óptico [12] para calcular a velocidade da curva durante o vídeo, além de outros métodos.

Em métodos de simples subtração, a intensidade de cada *pixel* em pontos adjacentes no tempo é subtraída de cada um; resultados com valores diferentes de zero

indicam que algo mudou na imagem. Isso geralmente é interpretado como movimento ao invés de efeito de iluminação. Infelizmente para vídeos onde o objeto apresenta deformações no tempo (objetos não rígidos), o rastreamento baseado apenas na subtração de *frames* junto com um contorno poligonal do objeto é falho. Devido a esse problema, decidiu-se optar por um rastreamento realizado a partir de uma máscara deformável que utiliza o conceito de contornos ativos, mais precisamente o método chamado *snakes* [5] [6].

Detalhes sobre o método *snakes* podem ser vistos na seção 2. Na seção 3 é apresentada a metodologia de trabalho e são descritos os resultados obtidos com a parte prática do projeto. Por fim, na seção 4 são realizadas as considerações finais.

## II. MÉTODO DE CONSTRUÇÃO DE MÁSCARAS

O rastreamento é baseado em seguir determinado padrão numa sequência de imagens ou *frames* de um vídeo. Dado uma imagem o padrão que será rastreado é determinado por um contorno do objeto de interesse, que neste caso chamaremos de máscara. A máscara é inicializada por um conjunto de pontos distribuídos em volta do objeto de interesse. Esses pontos são interpolados por uma interpolação bilinear, a qual resulta numa curva poligonal fechada. Devido à necessidade de rastrear uma máscara deformável (o formato do objeto de interesse pode variar *frame a frame*) não podemos trabalhar apenas com um contorno poligonal. Para obtermos um contorno deformável e exato do objeto de interesse, nos aplicamos o método chamado Contorno Ativo ou *snakes* a partir do contorno poligonal inicial. Detalhe do método *snakes* será detalhado a seguir.

### A. Contorno Ativo ou Snakes

Referente ao método para construção de máscaras, optou-se por trabalhar com os modelos de Contorno Ativo, também conhecido como *snakes*, que foram introduzidos por Kass, Witkin e Terzoupoulos em 1987 [5]. A ideia geral do modelo é a utilização de uma curva de minimização de energia para extrair características importantes de uma imagem. Essa energia associada à curva é definida de forma que ela seja mínima quando a curva se encontrar sobre uma região com as características que se deseja extrair, dessa forma a função de energia passa a funcionar como uma função objetivo em um problema de otimização. As aplicações de modelos de contorno ativo que mais se destacam se encontram nas áreas de segmentação e rastreamento. Geralmente, as curvas são projetadas por meio de bordas por forças de potenciais, as quais são definidas pelo gradiente negativa de uma função potencial. Forças adicionais, tais como forças de pressão, junto com forças de potencial englobam as chamadas forças externas que agem na função energia. Há também as forças internas, de elasticidade e rigidez, as quais servem para manter a curva junto ao objeto de interesse segmentado.

Uma *snake* paramétrica 2D tradicional é uma curva do tipo  $x(s) = (x(s), y(s))$ , parametrizada pelo comprimento de arcos. A *snake* move-se através do domínio espacial de uma imagem minimizando a seguinte equação de energia [11]:

$$E = \int_0^1 \frac{1}{2} \left[ \alpha |\mathbf{x}'(s)|^2 + \beta |\mathbf{x}''(s)|^2 \right] + E_{ext}(\mathbf{x}(s)) ds \quad (1)$$

sendo que o termo da energia do primeiro colchete da equação define a energia interna da *snake*. Os parâmetros  $\alpha$  e  $\beta$  controlam a tensão e a rigidez da *snake* respectivamente. A função da energia externa  $E_{ext}$  é normalmente definida como:

$$E_{ext}(x, y) = - |\nabla [G_{\sigma}(x, y) * I(x, y)]|^2 \quad (2)$$

sendo  $I(x, y)$  a intensidade da imagem (imagem em níveis de cinzas) e  $G_{\sigma}(x, y)$  a função Gaussiana 2D com desvio padrão  $\sigma$ . A *snakes* que minimiza  $E$  deve satisfazer a Equação de Euler dada por:

$$\alpha \mathbf{x}''(s) - \beta \mathbf{x}''''(s) - \nabla E_{ext}(\mathbf{x}(s)) = 0 \quad (3)$$

Isto pode ser visto como uma equação de balanços de forças:

$$F_{int} + F_{ext}^{(p)} = 0 \quad (4)$$

Diferente da abordagem considerada por Kass [5], o método de contorno ativo usando o método de Fluxo de Vetores Gradientes (conhecida como GVF) [11] utiliza um campo de vetores atrativos  $v(u, v) = (u(x, y), v(x, y))$  para dirigir o contorno sobre algumas das dificuldades encontradas na segmentação de formas como, por exemplo, problemas associados com a inicialização e “pobre” convergência das concavidades da forma segmentada.

Neste contexto,  $f(x, y) = -E_{ext}(x, y)$  é o mapa de bordas, e  $g$  é a função decrescente da magnitude do gradiente definida por [13].

$$g(|\nabla f|) = \exp\left(-\frac{|\nabla f|}{k}\right) \quad (5)$$

sendo  $K$  uma constante positiva que controla a suavidade do campo resultante. Os cálculos para as variações da *snakes* são novamente aplicados a partir das seguintes equações de Euler:

$$g^2 u - (1 - g)(u - f_x) = 0 \quad (6)$$

$$g^2 v - (1 - g)(v - f_y) = 0 \quad (7)$$

resolvendo as equações acima é obtido o campo de fluxo de vetores gradientes  $(u, v)$  que minimiza a função energia. A seguir, um exemplo da implementação do método *snakes* para segmentar a forma de interesse (Figuras 1a e 1b) é apresentado. Resultados da obtenção da *snakes* sem e com a utilização do método de Fluxo de Vetores Gradientes (Figuras 2a- 2b 2c) podem ser comparados.

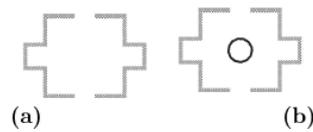


Fig. 1. Forma a ser segmentada em (a) e, em (b) com a snake inicial em azul.

Conforme pode ser conferida nas Figuras 2(c), 2(f) e 2(i), a *snakes* final teve como resultado satisfatório quando utilizou a abordagem com o campo GVF, ao contrário das outras duas abordagens que utilizaram a força baseada na Transformada Distância e a força baseada no campo gradiente, respectivamente.

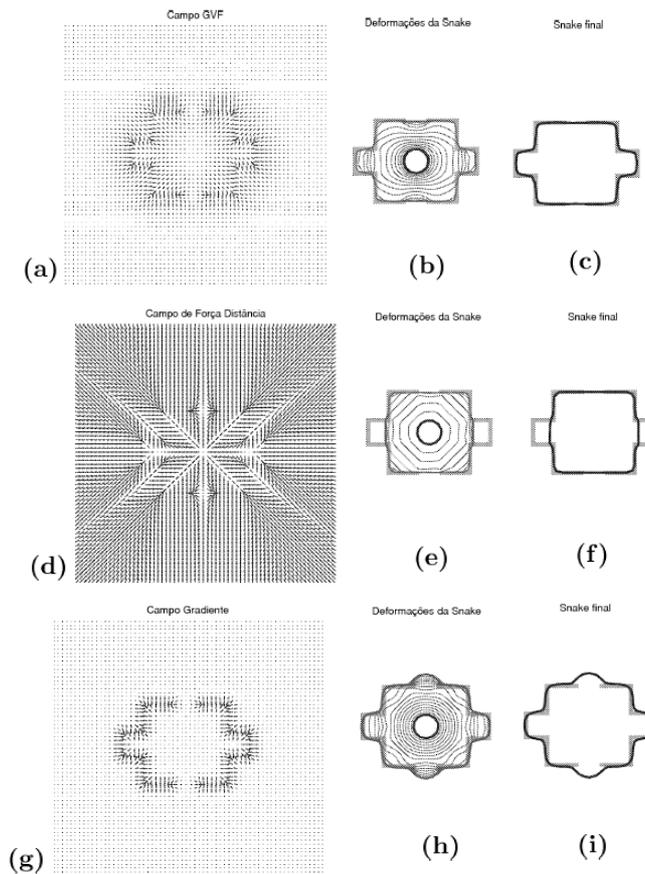


Fig. 2. Resultados para a segmentação da forma da Figura 1a: usando a abordagem do campo GVF em (a), a snake sofre deformações conforme (b), até obter sua forma final em (c); usando a abordagem da força baseada na Transformada Distância em (d), a *snake* sofre deformações conforme (e), até obter sua forma final em (f); e usando a abordagem do campo gradiente em (g), a *snake* sofre deformações conforme (h), até obter sua forma final em (i).

## B. Fluxo óptico

Com relação ao método para rastreamento de objetos em vídeos foram estudados dois métodos. O de Fluxo Óptico e a detecção do movimento saliente.

O método de Fluxo Óptico aplicado em uma sequência de imagens extrai parâmetros que revelam o comportamento do ambiente numa imagem. Mais especificamente, é possível rastrear um objeto através do seu campo de movimento 2D

quando comparamos um frame com outro [2][16]. No trabalho de Lucas-Kanade [14], fluxo óptico é entendido como uma distribuição de velocidades aparentes do movimento de padrões de brilho numa imagem. Este método apresenta um custo computacional relativamente baixo e uma boa tolerância a ruídos.

De uma forma geral, a determinação do fluxo óptico consiste em encontrar uma distribuição bidimensional das velocidades em uma sequência de imagens. Este cálculo resulta em um campo denso de vetores que normalmente é representado por um diagrama de vetores

Existem um grande número de trabalhos que buscam determinar o fluxo óptico na busca de melhor representação. Nós escolhemos o algoritmo proposto por Lucas-Kanade, o qual assume que valores das intensidades dos pixels em qualquer região da imagem, entre frames diferentes, não mudam, mas apenas se deslocam de uma posição para outra.

Assim, considerando que a intensidade de um pixel da imagem é dado por  $I(x,y,t)$ , adotaremos que a intensidade é função do tempo  $t$  assim como  $x$  e  $y$  também é. Para um pixel numa posição pouco distante, e num pequeno intervalo de tempo, a intensidade será

$$I(x + dx, y + dy, t + dt) = I(x, y, t) + \frac{\partial I}{\partial x}dx + \frac{\partial I}{\partial y}dy + \frac{\partial I}{\partial t}dt \quad \text{---}$$

Agora supondo que um objeto na imagem está na posição  $(x, y)$  no tempo  $t$ , e que depois de um determinado tempo  $dt$  esse objeto se mova a distância  $(dx, dy)$  na imagem. E, além disso, supondo que a intensidade da parte do objeto é justamente a mesma na imagem antes e depois do movimento, teremos que:

$$I(x + dx, y + dy, t + dt) = I(x, y, t), \quad (8)$$

sendo portanto

$$\frac{\partial I}{\partial x}dx + \frac{\partial I}{\partial y}dy + \frac{\partial I}{\partial t}dt = 0. \quad (9)$$

Dividindo tudo por  $dt$ , obtém-se:

$$\frac{dx}{dt} = u \quad e \quad \frac{dy}{dt} = v, \quad (10)$$

que são as velocidades do objeto nas direções  $x$  e  $y$  respectivamente. Assim, no limite de quando  $dt$  tende a zero, teremos:

$$\frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v = -\frac{\partial I}{\partial t}, \quad (11)$$

que é conhecida como Equação de Abertura do Fluxo Óptico. Esta equação possui infinitas soluções, já que se trata de um sistema de uma única equação e duas incógnitas.

Os métodos baseados em gradientes calculam a velocidade da imagem através das derivadas espaço-temporais de sua intensidade. O domínio da imagem é considerado contínuo, portanto, diferenciável, no espaço e no tempo.

O método proposto por Lucas-Kanade [7] é uma das abordagens que utilizam gradientes usando como variação a multi-restrição, o qual assume que a mudança de fluxo óptico é aproximadamente linear, assim sendo uma solução de suavização para a estimação do fluxo óptico pode ser obtida a partir de uma aproximação linear da restrição usada numa vizinhança  $N \times N$  do ponto sob observação. Mas detalhes sobre este método podem ser vistos nos trabalhos de Gonçalves [16] e Yokoyama [17].

### C. Detecção de Movimento Saliente

Este algoritmo realiza a detecção do movimento saliente em ambiente complexos combinando informações da diferença de imagens (*frames*) no tempo e o campo de fluxo óptico filtrado no tempo. Comparado a métodos que utiliza a subtração de *background*, este método não necessita de uma rotina de aprendizagem do modelo de *background*, o qual conta com grandes quantidades de imagens sem objetos em movimento. Este método se destaca na aplicação em vídeos onde haja mudanças rápidas na imagem como, por exemplo, iluminação.

A detecção de objetos em movimento em uma sequência de imagens é uma técnica útil e prática que permite que seja agregada características cinéticas as características espaciais de um determinado objeto em movimento. Pode ser útil na redução de tempo de processamento quando utilizada sobre uma sequência de imagens para descobrir quais partes podem ser consideradas fundo (objetos imóveis na cena) e quais partes da imagem podem ser consideradas de interesse. Esta abordagem pode ser implementada de maneira intuitiva através da subtração de uma imagem de uma cena obtida no instantes  $I(t)$  com uma imagem em um instante anterior  $I(t - 1)$ , que pode ser escrita como:

$$S(t) = I(t) - I(t - 1);$$

onde  $S(t)$  é uma matriz cujas posições  $(x, y)$  apresentam a diferença entre cada posição  $(x, y)$  da imagem  $I(t)$  e sua respectiva  $(x, y)$  em  $I(t - 1)$ . Necessariamente é realizado uma operação de binarização da imagem  $S(t)$  através de um limiar  $T$ , ou seja:

$$S(t) = \begin{cases} 1 & |I(t) - I(t-1)| \geq T \\ 0 & |I(t) - I(t-1)| < T \end{cases}$$

Considerando que dentro do intervalo de tempo  $t - 1$  a  $t$  não ocorra alterações no ambiente como, por exemplo, iluminação e deslocamento do fundo, as duas imagens serão equivalentes e a diferença dentre elas gerarem uma imagem  $S(t)$ , cujos pixels terão o valor zero ou, caso ocorra alguma alteração na imagem  $I(t)$ , responderá a valores iguais a um em  $S(t)$ .

Mas está ideia falha ao se trabalhar com rastreamento em vídeo onde possa haver mudanças no fundo. Por este motivo a abordagem de detecção de movimento saliente leva em consideração um parâmetro que carrega informações do pixel durante um intervalo de tempo (ou mais precisamente, o número de *frames*), o qual chamou de Imagem de Informação Acumulativa,  $I_{acum}$ . Este parâmetro é usado para obter a Imagem de Diferença Temporal  $I_{dif}(x, y, t)$  a qual fornece informações para as novas observações, conforme as equações seguintes [15]:

$$I_{dif}(x, y, t + 1) = \begin{cases} 1, & \text{se } (I_{acum}(x, y, t + 1) > T_d) \\ 0, & \text{caso contrario.} \end{cases} \quad (12)$$

$T_d$  é um parâmetro de limiar estatístico. E

$$I_{acum}(x, y, t + 1) = (1 - W_{acum})I_{acum}(x, y, t) + W_{acum}|I(x, y, t + 1) - I(x, y, t)|,$$

sendo  $W_{acum}$  o parâmetro de Acumulação o qual descreve a faixa temporal para acumulação de imagens diferentes.  $I_{acum}(x, y, t - 1)$  é inicializada como uma imagem vazia.

Após a obtenção destes dados o fluxo óptico é obtido e então um filtro temporal é aplicado nas regiões onde houveram mudanças, resultados obtidos pela diferença temporal de imagens. Assim podem-se detectar os pixels que se moveram continuamente na mesma direção. Com a soma destas informações, podemos rastrear os pixels do contorno original do objeto desde o primeiro *frame* até um determinado *frame n*.

### III. METODOLOGIA E RESULTADO

Para trabalhar com rastreamento de pessoas, um conjunto de condições de contorno foi considerado e a partir disso, é que houve um estudo de vários métodos para rastreamento e verificado quais realmente poderiam ser aplicados. Nesse trabalho foram levadas em consideração algumas condições iniciais para a realização do rastreamento de pessoas, como por exemplo:

- Pessoas são consideradas objetos não-rígidos, ou seja, deformáveis;

- O vídeo foi obtido por uma única câmera fixa;
- As pessoas são rastreadas através de contornos (máscaras) do corpo obtidas por meio de um algoritmo aplicado no *frame* inicial do vídeo. Importante comentar que o primeiro *frame* de um filme muitas vezes pode não apresentar todas as personagens do roteiro que serão rastreadas. Diante disso, é considerado como *frame* inicial para uma determinada máscara, o qual surgir determinada personagem pela primeira vez;
- Os contornos rastreados podem contornar todo o corpo de uma pessoa ou partes dele;
- Não foi considerada mudanças de luminosidade que possam ocorrer no vídeo;
- O vídeo apresenta cenas internas em um cenário fixo;
- Utilização de vídeos sem a presença de cenas onde aparece somente o *background*.

Algumas vezes ao estudar ou aplicar determinados métodos de rastreamento, deparou-se com problemas os quais acabavam barrando as intenções de utilizar os mesmos no projeto. Portanto houve um trabalho de busca e estudo de cada método, além do tempo proporcional para cada implementação e análise dos resultados que foi necessário realizarem para decidir se seguia ou não com o método. O vídeo utilizado para o projeto foi filmado antes de se iniciar o estudo dos algoritmos, e por isso, algumas características deste vídeo acabaram limitando a aplicação de alguns métodos como, por exemplo, métodos que utilizam um treinamento de *background* das cenas a serem rastreadas sem a presença de objetos que se movem.

A seguir são apresentados os resultados dos testes e aplicações dos diferentes métodos para rastreamento de máscaras (objetos) deformáveis. Os testes foram realizados em imagens (*frames*) e vídeos listados a seguir:

- Vídeo1: “VP00.mpeg” que foi filmado especificamente para o projeto;
- Vídeo2: “problem-raw.mpg” vídeo somente para teste.

Com relação aos resultados referentes à máscara, a Figura 3 apresenta os resultados obtidos na construção de uma máscara na cabeça do assaltante (vídeo1). Conforme o método proposto, a partir de um contorno poligonal obtido por pontos marcados na imagem, o método da snakes é aplicado e conforme o número de interações chega-se ao contorno final ou máscara final. O método snakes utilizando o GFV foi considerado satisfatório. Obtiveram-se bons resultados quando testado para outras partes do corpo das pessoas do vídeo1.

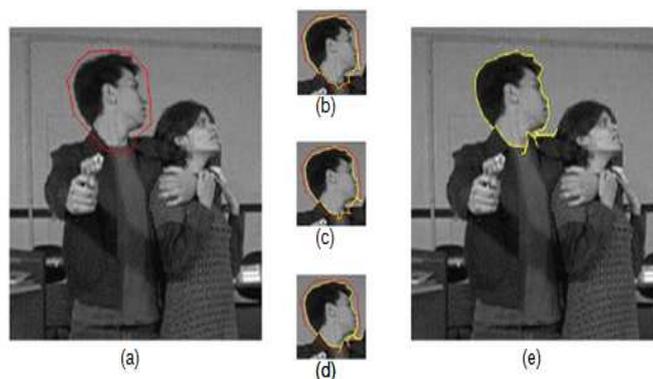


Fig. 3. Contorno poligonal inicial em vermelho obtido pela interpolação bilinear em (a); em (b), (c) e (d) o contorno (em amarelo) depois de 5, 10 e 30 iterações, respectivamente; e em (e) o contorno final em amarelo obtido pelo método *snake* utilizando o Fluxo de Vetores Gradientes (GVF).

Dado a máscara final, o próximo passo foi rastrear esta máscara durante o seguimento do filme. Foi trabalhada a abordagem utilizando Fluxo Óptico. Nessa abordagem foram aplicados os métodos de obtenção do fluxo óptico, por Lucas-Kanade [14], juntamente com a abordagem de *snakes* utilizada para a construção das máscaras. Dada uma máscara no *frame* referência, calcula-se o fluxo óptico em relação ao próximo *frame*. Os valores das velocidades  $V_x$  e  $V_y$  obtidos são utilizados para movimentar a máscara atual em cada ponto  $(x, y)$  do contorno, obedecendo a seguinte grade de posições:

$V_x < 0$	$V_x = 0$	$V_x > 0$
$V_y < 0$	$V_y < 0$	$V_y < 0$
$V_x < 0$	$V_x = 0$	$V_x > 0$
$V_y = 0$	$V_y = 0$	$V_y = 0$
$V_x < 0$	$V_x = 0$	$V_x > 0$
$V_y > 0$	$V_y > 0$	$V_y > 0$

Assim, dado um ponto do contorno cuja posição é  $(x, y)$ , sua próxima posição seria, por exemplo,  $(x + 1, y + 1)$  caso suas velocidades fossem  $V_x > 0$  e  $V_y > 0$  conforme o campo de fluxo óptico obtido. O *pixel* de posição  $(x, y)$  presente na lista de pontos “contorno atual” é atualizado em uma nova lista chamada novo contorno. A partir destes novos pontos, é realizada uma interpolação e necessariamente é utilizado o método *snakes* com GVF para obter o novo (próximo) contorno do objeto (máscara) de interesse para o próximo *frame*.

Esta abordagem foi aplicada para três tipos de sequências de vídeos. O primeiro teste foi realizado em uma sequência do vídeo2 desde seu primeiro *frame*. Conforme os *frames* apresentados na Figura 4, o resultado obtido nos primeiros 40 *frames*, pode-se comprovar que esta aplicação não foi satisfatória. Isto se deve ao desvio e à atração do contorno por uma borda do fundo (*background*), ou seja, inicialmente esta borda está oclusa pelo objeto de interesse. Uma vez o objeto em movimento, a borda aparece e conforme a abordagem do método de Fluxo Óptico, esta borda “ganha” velocidade atraindo o contorno para si. Este resultado pode ser conferido a partir do *frame* 10 (Figura 4b). Outro problema encontrado foi observado a partir do *frame* 37 (Figura 4e) quando o objeto permanece parado por um tempo e o contorno começa a se confundir

devido às velocidades dos pontos da borda do objeto (máscara) apresentar velocidades baixas ou aproximadamente zero.

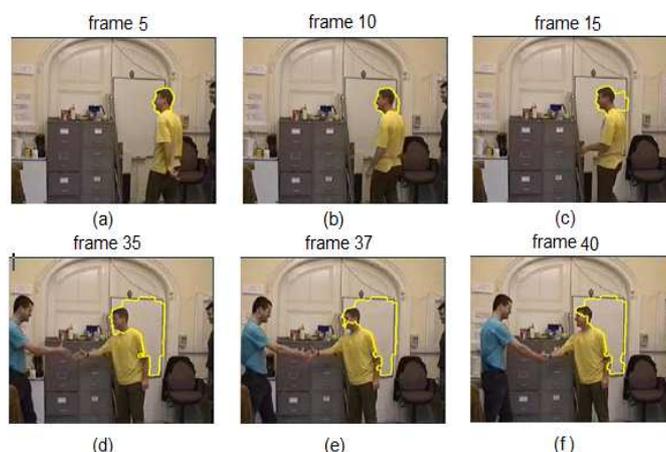


Fig. 4. Resultado do primeiro teste para os *frames* n° 5, 10, 15, 35, 37 e 40 da sequência do vídeo2.

O segundo teste foi realizado em uma sequência do vídeo2, só que agora se escolheu iniciar o rastreamento a partir do *frame* 11 (Figuras 5a), a fim de evitar o problema apontado no teste anterior, da atração do contorno pela borda do fundo. O rastreamento se comporta bem até o *frame* 37 (Figura 5e) quando o objeto permanece imóvel por um tempo. O contorno do objeto começa a ter distorções e é atraído para outras bordas internas do objeto.

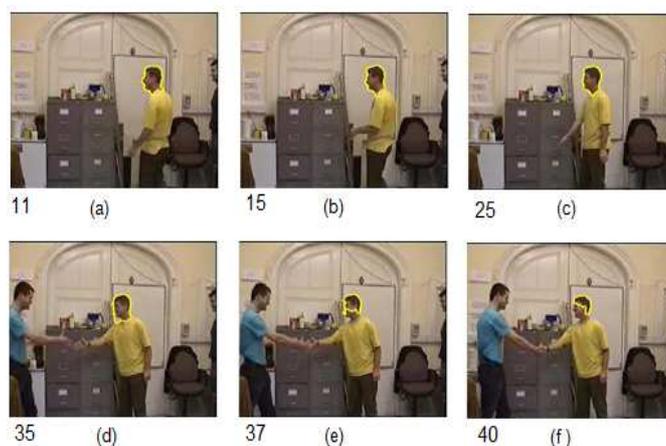


Fig. 5. Resultado do segundo teste para os *frames* n° 11, 15, 25, 35, 37 e 40 da sequência do vídeo2.

No terceiro teste utilizou-se uma sequência do vídeo1, no intervalo de *frames* 1565 a 1924, os quais foram renomeados como *frames* 1 a 360. Verificou-se um desvio da máscara já no início do rastreamento entre os *frames* 1 a 5 conforme as Figuras 6a e 6b, talvez devido á borda "fraca" de uma parte do objeto. Depois disso o contorno se recupera e volta a ter distorções no *frame* 130 (Figuras 6c e 6d). Entre os *frames* 170 e 190, é observado um intervalo sem movimento do objeto e assim o problema de distorção do contorno volta a ocorrer, conforme é mostrado nas Figuras 6(e) e (f). Já no *frame* 235 (Figura 6g) percebe-se que o contorno atrai bordas internas do objeto, mas

apesar de alguns desvios como no *frame* 295 (Figura 6h), o contorno termina deslocado no final do rastreamento no *frame* 360 (Figura 6i).

Conforme se pode concluir a partir dos resultados apresentados nos três conjuntos de *frames*, a aplicação desta abordagem não foi satisfatória. Problemas com áreas de oclusões, movimento do objeto temporariamente imóvel e bordas com pouco contrastes do objeto são alguns motivos pelo qual o rastreamento de objetos deformáveis falha.

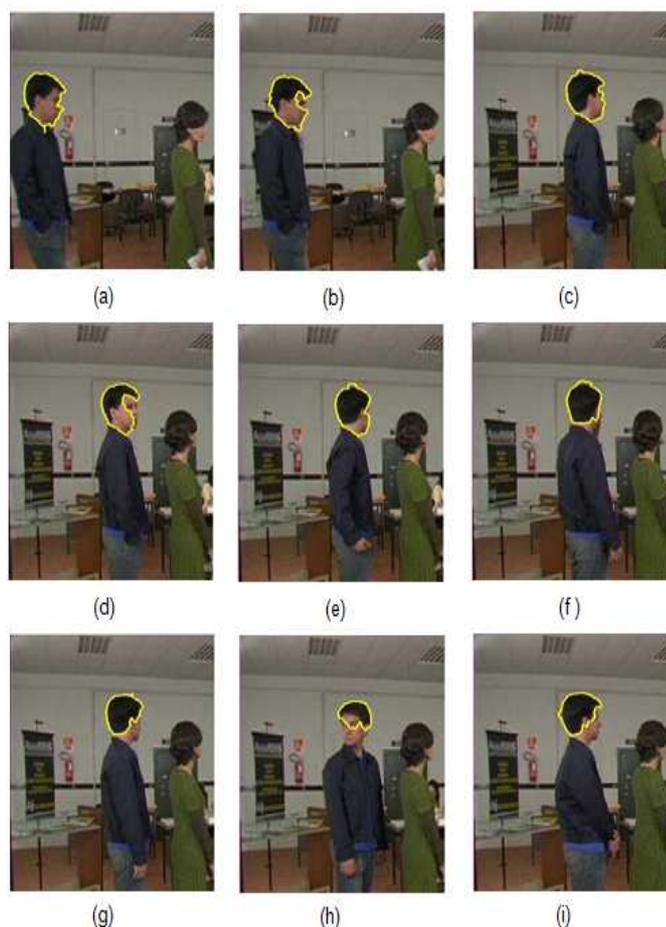


Fig. 6. Resultado do terceiro teste para os *frames* nº 1, 5, 75, 130, 170, 190, 235, 295 e 360 da sequência do vídeo1.

Diante disso foi utilizada a técnica de detecção de movimento saliente. Foram realizados diferentes testes com parâmetros variados, com exceção aos parâmetros  $T_d = 15$  e  $W_{acum} = 0.5$ , os quais se resolveu adotá-los conforme o artigo de Ying-Li [15]. Esta abordagem foi aplicada para quatro tipos de variedades do método.

No primeiro teste foi calculada a diferença temporal e em seguida o fluxo óptico entre *frames*. O campo de fluxo óptico é filtrado com o resultado da diferença temporal dos *frames*. Para obter o contorno final para o *frame* atual é aplicado à rotina de obtenção da máscara utilizando como borda de atração o resultado da diferença temporal atual do *frame*. Os resultados são apresentados na Figura 7. A partir da máscara do primeiro

*frame* (Figura 7a) observa-se que o rastreamento não é atraído por bordas fortes do fundo conforme mostrado na Figura 7d, ou seja, a atração ocorre somente quando o objeto está próximo da borda do fundo (Figura 7c), após isso ele percebe que a borda do fundo não apresenta movimento saliente e o contorno volta a se interessar somente pelo objeto em movimento.

Um dos pontos fracos desta variação da abordagem foi o leve atraso, em alguns momentos, com que o contorno persegue o objeto, conforme pode ser verificado na Figura 7b e 7d. Isso pode estar relacionado com a velocidade com que o objeto se move no vídeo. Outra desvantagem foi detectada quando o objeto apresentou um movimento lento ou de quase imobilidade, resultando em uma perda ou desvio da máscara rastreada (Figuras 7e e 7f).

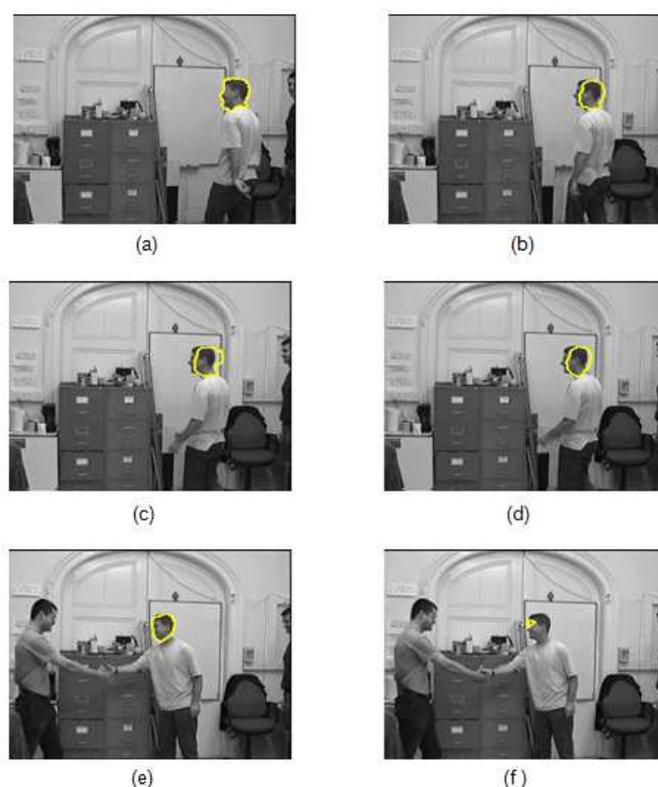


Fig. 7. Resultado do primeiro teste para os *frames* nº 1, 9, 11, 12, 38 e 39 da sequência do vídeo2.

No segundo teste houve apenas uma mudança em relação ao primeiro teste, calculou-se a diferença temporal e em seguida o fluxo óptico entre *frames*. O campo de fluxo óptico foi filtrado com o resultado da diferença temporal dos *frames*. Para obter o contorno final para o *frame* atual foi aplicado à rotina de obtenção da máscara utilizando como borda de atração o resultado da diferença temporal, só que neste teste utilizaram-se informações do próximo *frame*.

Os resultados são apresentados na Figura 8. A partir da máscara do primeiro *frame* (Figura 8a) observa-se que o rastreamento não é atraído por bordas fortes do fundo conforme mostrado na Figura 8b - 8d. Este resultado é melhor quando comparado ao

teste anterior, pois a borda do fundo não apresenta influência nem mesmo quando esta perto do objeto. Outro destaque está na diminuição do atraso da máscara durante o rastreamento. Pode-se considerar que não existe atraso. O ponto fraco ainda surge para quando existe movimento lento ou de quase imobilidade do objeto, resultando em uma perda ou desvio da máscara rastreada assim como pode ser verificado nas Figuras 8e - 8f.

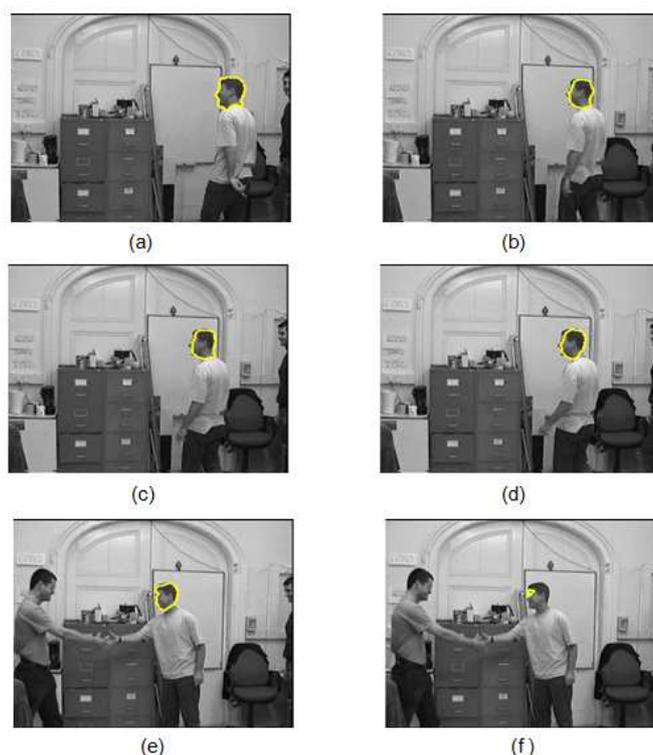


Fig. 8. Resultado do segundo teste para os *frames* nº 1, 9, 11, 12, 38 e 39 da sequência do vídeo2.

O terceiro teste apresenta uma mudança em relação aos anteriores na obtenção da máscara final. A partir da diferença temporal entre *frames*, obtém-se o fluxo óptico e o mapa de bordas usando o filtro Laplaciano da Gaussiana. O campo de fluxo óptico é filtrado com o resultado da diferença temporal dos *frames*. Para obter o contorno final para o *frame* atual é aplicado à rotina de obtenção da máscara utilizando como borda de atração, o mapa de bordas do próximo *frame*.

Os resultados são apresentados na Figura 9 a seguir. A partir da máscara do primeiro *frame* (Figura 9a) observa-se que o rastreamento não é atraído, ou quase nada, por bordas fortes do fundo conforme mostrado na Figura 9c - 9d. Este resultado pode ser considerado o melhor quando comparado com os dois testes anteriores, pois a borda do fundo apresenta uma influência pequena quando está próxima do objeto. É possível considerar que não existe atraso da máscara durante o rastreamento.

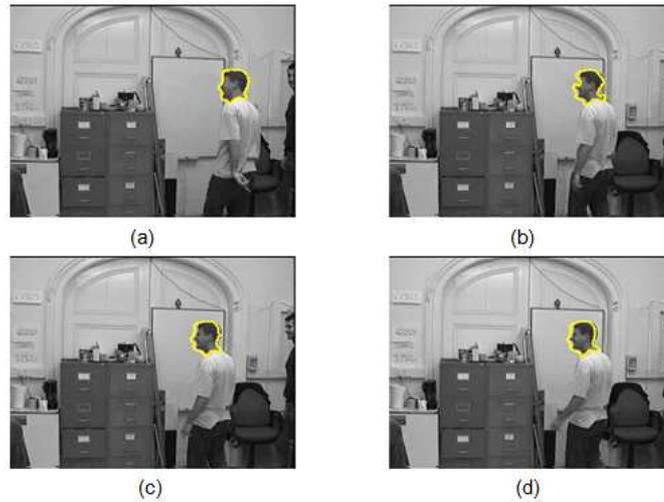


Fig. 9. Resultado do terceiro teste para os *frames* nº 1, 9, 10, 11 da sequência do vídeo2.

O quarto teste apresenta apenas uma mudança em relação ao teste anterior. Após o cálculo da diferença temporal entre *frames*, é obtido o fluxo óptico e, o mapa de bordas usando o filtro Laplaciano da Gaussiana. O campo de fluxo óptico é filtrado com o resultado da diferença temporal dos *frames*. Para obter o contorno final para o *frame* atual é aplicado a rotina de obtenção da máscara utilizando como borda de atração, o mapa de bordas do *frame* atual. Os resultados são apresentados na Figura 10. A partir da máscara do primeiro *frame* (Figura 10a) observa-se que o rastreamento é atraído, pela borda forte do fundo conforme mostrado na Figura 10b e 10c. Este resultado pode ser considerado o pior quando comparado com os três testes anteriores. Como houve desvio do contorno por causa de bordas fortes, não temos como comparar sobre o fato de haver ou não atraso das máscaras. Outra característica a ser descartada neste teste é sobre o desvio do contorno para quando existe movimento lento ou de quase imobilidade do objeto, resultando em um desvio da máscara rastreada assim como verificado na Figura 10d.



Fig. 10. Resultado do quarto teste para os *frames* nº 1, 9, 10, 36 da sequência do vídeo2.

No quinto teste, a mudança foi desfeita e aplicada à versão anterior (algoritmo do teste 3). O teste foi realizado em uma sequência de *frames* do vídeo2 e o seu resultado pode ser observado na Figura 11. Na Figura 12 é mostrado em detalhes do processamento da imagem destacando o contorno da cabeça da pessoa para demonstrar a eficiência do algoritmo. Por fim, no sexto teste foi aplicado o algoritmo nos *frames* do vídeo1. O algoritmo se mostrou adequado e respondeu corretamente ao acompanhamento da máscara ao longo da sequência (Figura 13).

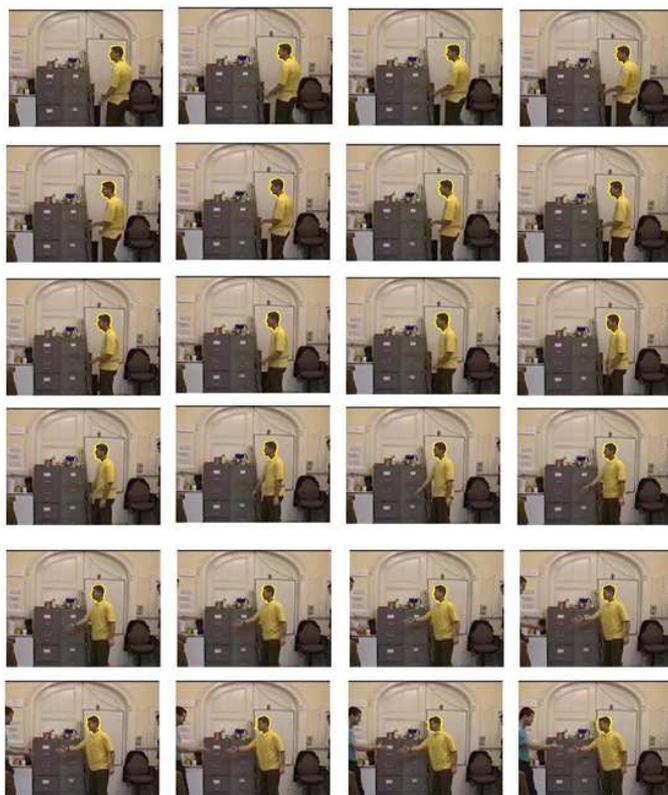


Fig. 11. Sequência de *frames* do quinto teste para sequência do vídeo2.



Fig. 12. Detalhe do resultado do sexto teste para sequência do vídeo2.

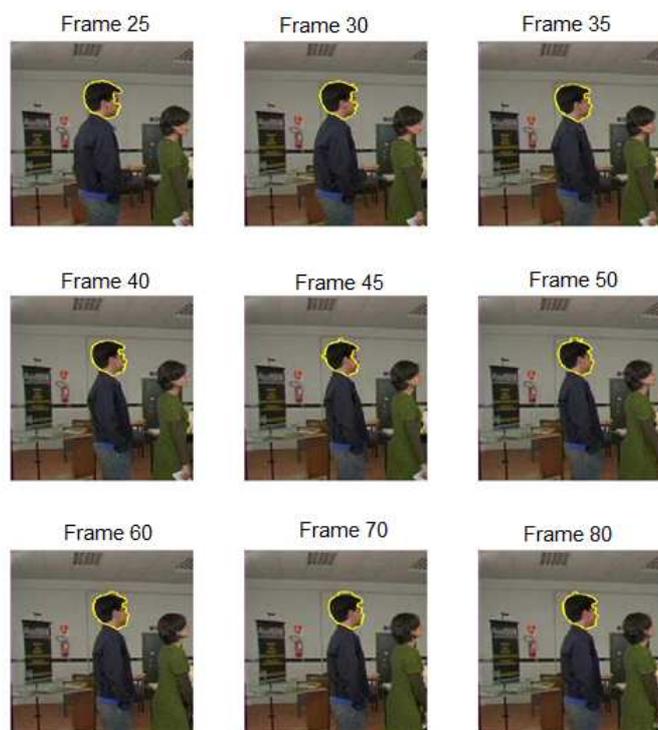


Fig. 13. Resultado do sétimo teste para sequência do vídeo1.

#### IV. CONSIDERAÇÕES FINAIS

A intenção foi utilizar um método que satisfaça todas as condições de contorno encontradas no vídeo1, como por exemplo, ambiente interno, iluminação, câmera fixa, entre outros. Em virtude da variedade dos métodos e características do vídeo, a ideia foi implementar diferentes algoritmos e abordagens de rastreamento para pessoas, e analisá-las para diferentes condições de ambientes e disposições dos objetos rastreados, a fim de encontrar a metodologia que melhor satisfizesse o sistema de rastreamento para o simulador.

Conforme o objetivo inicial de rastrear pessoas por meio de máscaras (partes do corpo) a partir de um vídeo, capturado por uma única câmera fixa na gravação de uma cena interna. A mudança de máscaras poligonais para máscaras deformáveis apresentou um resultado satisfatório, conforme pode ser visto nos resultados apresentados na seção 4. A execução do algoritmo para a construção da máscara não é considerado rápido para ser aplicado em tempo real do rastreamento do vídeo, mas isso não é prejudicial já que o relevante neste trabalho é obter máscaras que representem bem o objeto a ser rastreado. Posteriormente, o aumento do processador e da memória RAM do computador do simulador gerou um resultado satisfatório para o funcionamento correto.

Devido às informações das máscaras rastreadas precisarem ser divididas em: pessoas (vítimas ou assaltantes), partes do corpo (vitais ou não vitais) ou objetos relevantes (armas) decidiu-se optar por um rastreamento realizado a partir de uma máscara deformável que utiliza a técnica *snakes* (contornos ativos). Esta escolha foi realizada em razão da importância de cada máscara conter informações de relevância para a continuação do treinamento, pois quanto mais preciso fosse o contorno, mais adequado seria a tomada de decisão baseada no conjunto de imagens desenvolvida

especificamente para o projeto. Isto é, a técnica se mostrou positiva para este conjunto em específico de cenas.

Um estudo sobre diferentes abordagens de rastreamento de objetos deformáveis (não-rígidos) foram estudados e analisados. Foram investigados rastreamentos que utilizassem abordagem como, *Mean-Shift* [1], Contorno Ativos Dinâmicos, Estimação de Movimento e Rastreamento usando Regiões Ativas de Geodésicas, entre outros. Dentre eles foram implementados para teste, métodos que utilizavam Fluxo Óptico, Detecção de Movimento Saliente em vídeos, abordagens sobre *Level-Set*. Esses métodos foram escolhidos devido à facilidade de implementação e os quais apresentavam referências mais adequadas ao projeto. A ideia foi buscar métodos não necessariamente que funcionassem em tempo real de execução, mas que apresentassem rotinas rápidas computacionalmente, já que um vídeo simples pode ter milhares de *frames* a serem rastreados e trabalhados.

A escolha dos métodos também levou em consideração a necessidade de trabalhar a partir de determinadas condições de contorno. Infelizmente, alguns dos métodos testados, falharam para algumas destas condições, que só foram percebidas após a implementação e os testes realizados. Todas falharam em algum tipo de característica do rastreamento, principalmente no desvio e perda da máscara do objeto rastreado.

Existem alguns detalhes que não foram trabalhados neste projeto, mas que merecem ser levado em consideração e possivelmente serem melhorados como, por exemplo, a realização de um estudo e melhoramento do mapa de bordas utilizado no método da *snakes* com GVF. Poderiam ser testados outros tipos de filtros para capturar bordas da imagem. Quanto mais bordas detectadas melhor será o resultado e mais perfeitas serão as máscaras dos objetos de interesse. Além disso, ampliar os testes realizando uma maior mudança de parâmetros e variáveis nas rotinas de cálculo do fluxo óptico e no método que utiliza detecção de movimento saliente e, por fim, realizar um estudo de métodos estatísticos de rastreamento de objetos os quais fossem somados aos já trabalhados.

## AGRADECIMENTOS

Agradecimento ao CNPq por meio do seu programa RHAÉ (Programa de Formação de Recursos Humanos em Áreas Estratégicas) e DT (Produtividade em Desenvolvimento Tecnológico e Extensão Inovadora).

## REFERÊNCIAS

- [1] D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using mean shift. In IEEE Conf. on Comp. Vis. and Pat. Rec., volume 2, pages 142–149, SC, USA, 2000.
- [2] J.H. Duncan and T.C. Chou. On the detection of motion and the computation of optical flow. PAMI, 14(3):346–352, March 1992.
- [3] R. Jain, W.N. Martin, and J.K. Aggarwal. Segmentation through the detection of change due to motion. CGIP, 11(1):13–34, September 1979.
- [4] R.C. Jain and H.H. Nagel. On the analysis of accumulative difference pictures from image sequences of real world scenes. PAMI, 1(2):206–213, April 1979.

- [5] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *IJCV*, 1(4):321–331, jan 1988.
- [6] S. C. D. Pinto, E. B. Santos, and A. V. Netto. Video tracking for an interactive training system. In *Proceedings of XIX SIBGRAPI - Brazilian Symposium on Computer Graphics and Image Processing*. IEEE Computer Society, 8–11 Oct. 2006.
- [7] N. Ray and S. T. Acton. Active contours for cell tracking. In *SSIAI '02: Proceedings of the Fifth IEEE Southwest Symposium on Image Analysis and Interpretation*, pages 274–278, Washington, USA, 2002.
- [8] James Albert Sethian. *Level set methods: Evolving interfaces in geometry, fluid mechanics, computer vision, and materials science*. Number 3 in Cambridge monographs on applied and computational mathematics. Cambridge University Press, Cambridge, U.K., 1996. 218 pages.
- [9] Y. Shi and W.C. Karl. Real-time tracking using level sets. In *CVPR 2005: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 34–41, San Diego, USA, 2005.
- [10] Yonggang Shi and William Clement Karl. Real-time tracking using level sets. In *CVPR (2)*, pages 34–41. IEEE Computer Society, 2005.
- [11] C. Xu and J. Prince. Snakes, shapes, and gradient vector flow. *IEEE Trans. Image Proc.*, 7(3):359–369, Mar. 1998.
- [12] M. Yokoyama and T. Poggio. A contour-based moving object detection and tracking. pages 271–276, 2005.
- [13] Nilanjan Ray and Scott T. Acton. Active contours for cell tracking. In *SSIAI '02: Proceedings of the Fifth IEEE Southwest Symposium on Image Analysis and Interpretation*, page 274, Washington, DC, USA, 2002. IEEE Computer Society.
- [14] B.D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *IJCAI81*, pages 674–679, 1981.
- [15] Ying-Li Tian and Arun Hampapur. Robust salient motion detection with complex background for real-time video surveillance. In *WACV-MOTION '05: Proceedings of the IEEE Workshop on Motion and Video Computing (WACV/MOTION'05) - Volume 2*, pages 30–35, Washington, DC, USA, 2005. IEEE Computer Society.
- [16] Paulo Henrique Gonçalves. *Estimação do fluxo multidirecional de pedestres em ambientes abertos e não restritos, pela análise de sequências de imagens digitais*. Master's thesis, USP, São Carlos, apr 2004.
- [15] M. Yokoyama and T. Poggio. A contour-based moving object detection and tracking. In *2nd Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pages 271– 276, 2005.
- [16] Y. Mae, Y. Shirai, J. Miura, and Y. Kuno. Object tracking in cluttered background based on optical flow and edges. In *In Proc. the 13th Int. Conf. on Pattern Recognition*, volume 1), pages = 196-200, address = Vienna, Austria, 1996.