Monitoramento em Tempo Real baseado em Fluxo Ótico

Luiz Carlos Maia Junior[1] **, Anna Helena Reali Costa**[1]

[1]Laboratório de Técnicas Inteligentes
Escola Politécnica da Universidade de São Paulo
Av. Prof. Luciano Gualberto, 158, tv. 3 – 05508-900 São Paulo, SP

luiz.maia@poli.usp.br, anna.reali@poli.usp.br

Abstract. *This paper describes a real time monitoring system that uses a region matching based algorithm for extracting velocity vectors fields in consecutive images to detect a movement direction, if it exists. As this algorithm has a high computational cost, other techniques were added, aiming at balancing quality and performance. A human-computer interface is used to define a mask and an image of the environement background. A background subtraction reveals the region of interest inside the area defined by the mask, this region is then processed. The system will be applied for underground train platform monitoring, generating alarms from invasion events of the train circulation area.*

Resumo. *Este artigo descreve um sistema de monitoramento em tempo real que utiliza um algoritmo baseado em correspondência entre regiões de imagens consecutivas para extração de campos de vetores de velocidade para detectar a direção de uma movimentação, se esta existir. Como esse algoritmo tem alto custo computacional, outras técnicas foram agregadas, com o objetivo de balancear qualidade e desempenho. Uma interface homem-computador é usada para definir uma máscara e uma imagem do fundo do ambiente. Uma subtração de fundo revela a região de interesse para cada quadro dentro da região definida pela máscara, esta região é então processada. O sistema será aplicado no monitoramento de plataformas de trens subterrâneos, gerando alarmes a partir de eventos de invasão da área de circulação dos trens.*

1. Introdução

Este artigo descreve um sistema de monitoramento em tempo real, que utiliza algoritmos de visão computacional, mais especificamente de extração de campos de vetores de velocidade, para detectar movimento através de uma seqüência de vídeo capturada por câmeras de vigilância. Este sistema apresenta uma solução alternativa para o monitoramento de áreas restritas ou consideradas de risco. Seu objetivo é detectar movimento nessas áreas e gerar alarmes, decorrentes de movimentações fora dos padrões normais.

Uma combinação de técnicas gera o algoritmo usado para extrair o campo de vetores. É essencial que haja um compromisso entre seu desempenho e sua precisão; por isso, durante a descrição do algoritmo proposto, serão notadas simplificações que visam esse balanceamento. A extração de campo de vetores de velocidade é o que faz com que o algoritmo tenha alto custo computacional; esta é, portanto, a principal área a sofrer de algumas adaptações para reduzir este custo. As adaptações foram voltadas à redução do custo computacional e ao acréscimo do número de câmeras monitoradas por um só computador.

O sistema será aplicado no monitoramento de plataformas de trens subterrâneos e tem como objetivo principal detectar movimento em áreas consideradas de risco ou restritas, e gerar alarmes. Estes alarmes podem ser decorrentes da movimentação ter ocorrido em direções, horários ou regiões nas quais a mesma não deveria ocorrer. O padrão que configura a geração de alarmes é definido para cada câmera a ser monitorada antes que seja iniciado o algoritmo que realizará essa função.

Este trabalho está organizado da seguinte forma: na seção 2 é descrito o sistema de monitoramento proposto. Na seção 3 é feita uma apresentação simplificada sobre algoritmos de análise de movimento e técnicas a eles associadas para aumento de qualidade e desempenho. Na seção 3.3 o algoritmo proposto é detalhado. Na seção 4 são apresentados alguns resultados preliminares. E, finalmente, a seção 5 apresenta conclusões.

2. O Sistema de Monitoramento Proposto

Metrôs, em geral, trabalham com intervalos reduzidos para poder atender à demanda de passageiros. O sistema proposto nesse trabalho se destina ao suporte do tratamento das situações de risco, tentando amenizar os efeitos de uma ocorrência indesejada.

Atualmente, o operador tem à sua disposição uma comuniçação por rádio ou telefone com os responsáveis pelas plataformas e um sistema que alterna as câmeras automaticamente, em seqüências, períodos ou uma combinação programada dos dois, e disponibiliza essas imagens em monitores próximos ao operador. A detecção da situação de risco depende de duas coisas: ou o sistema que alterna as câmeras está no exato momento da ocorrência mostrando a câmera correta e o operador está observando o vídeo, ou um responsável por plataforma percebe a situação (ou é avisado) e avisa o operador em seguida. Pode ser que situações, em casos extremos, passem despercebidas por mais de 30 segundos. O sistema, da forma que opera hoje, é mostrado na Figura 1 (a). O principal problema de operar desta maneira é a demora que pode ocorrer para que a situação de risco seja percebida, já que tanto os operadores do Circuito Fechado de TV (CFTV) quanto os responsáveis pelas plataformas têm outras tarefas a desempenhar.

O sistema proposto, mostrado na Figura 1 (b), visa manter um monitoramento ininterrupto das plataformas de trens e é composto pelo computador que processa o sinal de vídeo, pelo servidor e pela IHC (Interface Homem Computador) do operador. O computador que processa o vídeo envia alarmes e eventos ao servidor que os guarda e repassa

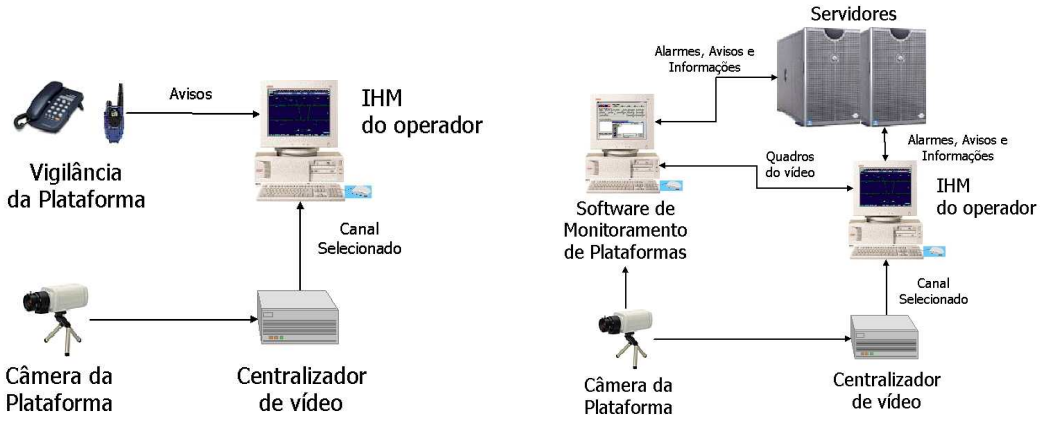


Figura 1: Visão geral do sistema de monitoramento: (a) Sistema atual de monitoramento de plataformas metroviárias. (b) Sistema proposto para monitoramento.

para a IHC do operador, que também recebe, do computador que processa o vídeo, quadros do momento do evento. Esse vídeo é guardado se o operador confirmar a situação de risco. É papel dos servidores serem *watchdogs* do *software* de monitoramento e, se detectarem a ausência de resposta do programa, devem gerar um alarme para a IHC. Os alarmes de situação de risco não virão diretamente do *software* de monitoramento pois devem passar pelo servidor para que sejam gravados na base de dados. O sistema pode ser configurado pelo operador para que em determinados horários não monitore determinadas câmeras para evitar a geração de alarmes em horários de manutenção ou limpeza da via. É importante ressaltar que decisões finais de proteção, como uma ação direta na parada da circulação dos trens, não serão tomadas pelo sistema, mas sim passadas para o operador que será o julgador final.

O sistema basicamente trabalha em um de quatro estados possíveis (ver Figura 2). O primeiro deles, e mais simples também, é quando não há nenhuma movimentação sendo detectada. Neste estado o sistema trabalha com folga de processamento pois não irá tentar extrair o campo de vetores de velocidade se nenhuma região de interesse tiver sido detectada. O segundo estado ocorre quando houver uma invasão na área de circulação de trens. Nesta situação o sistema poderá tanto voltar para o estado inicial quanto ir para o quarto estado. O terceiro estado é quando acontece a chegada do trem e durante toda a permanência do mesmo na plataforma. Também há neste estado a possibilidade tanto de voltar para o estado inicial quanto de ir para o quarto estado. E finalmente o quarto estado é quando existe a movimentação do trem e ao mesmo tempo uma invasão da área de circulação está ocorrendo. Os estados nos quais devem ser gerados alarmes são o segundo e o quarto, e o terceiro, apesar de apresentar movimentação, deve ser diferenciado dos demais através da análise tanto da direção do movimento quanto da área que está se movendo.

De todas as transições de estado do sistema, duas delas não devem ocorrer: do segundo para o quarto estado e do quarto para o segundo estado. A ocorrência de uma dessas transições pode sinalizar que a geração do alarme não ocorreu a tempo do operador tomar uma ação corretiva.

Pelo fato do sistema estar envolvido na detecção de situações de risco para o usuário, é esperado que o tempo de resposta do mesmo seja muito pequeno, menor que um segundo. Após a geração do alarme, a observação do mesmo pelo operador é aguardada para que alguma ação corretiva seja tomada. A confirmação do alarme pode ser apenas

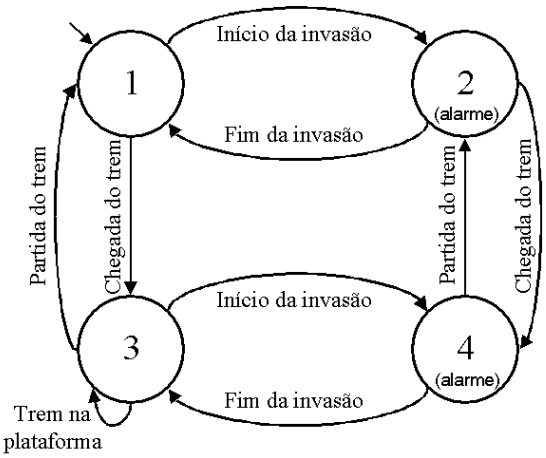


Figura 2: Estados e transições do sistema.

local, situação na qual o próprio operador decide o que fazer e a quem avisar, ou pode ser distribuída; nesse caso o alarme é repassado para o responsável pela área, o que pode acelerar o processo de correção se ambos estiverem treinados para enfrentar situações como essas.

Atualmente o sistema está sendo preparado para utilizar as câmeras de plataforma das estações de metrô, o que dificulta um pouco o trabalho, já que o posicionamento das câmeras não objetiva monitorar a área de circulação de trens e sim parte dela e da plataforma. Isso fez com que fosse preciso ser criada uma máscara que exclui da imagem a área da plataforma; por esse motivo, para cada câmera é necessária uma máscara correspondente. Essa máscara é uma imagem do tipo *Portable Greyscale* (PGM) porque, além de indicar a área a ser permanentemente excluída do processamento, ela possui o fundo estático da área que será processada (as câmeras são monocromáticas). Esse fundo será subtraído dos quadros capturados para obtenção da região de interesse para extração do campo de vetores de velocidade.

No caso estudado, a iluminação é constante devido à plataforma ser subterrânea; por isso, foi adotado um modelo de fundo fixo que serve para determinar a região de interesse. Seixas e Costa [2003] utilizam um algoritmo adaptativo para modelar a imagem de fundo que será subtraída e então obter a região de interesse; isto é necessário quando a iluminação não for constante.

Algumas considerações são feitas para eliminar ruídos da subtração de fundo. A primeira é definir um limiar de subtração que pode ser ajustado para eliminar pequenas variações na aquisição das seqüências de vídeo. Depois disso, operadores morfológicos [Gonzalez and Woods, 1992] são utilizados visando a eliminação de pequenas áreas isoladas (operador *open*) e, depois, para conectar pequenas áreas que estiverem próximas (operador *close*).

As imagens capturadas têm uma resolução de 320 por 240 *pixels* em 256 níveis da escala de cinzas. Essa resolução foi adotada por ser uma resolução intermediária e adequada para os interesses da aplicação.

O algoritmo utilizado no sistema proposto extrai o campo de vetores de velocidade através do algoritmo baseado em correspondência de regiões - BCR (*block matching* [Camus, 1997]), que será melhor detalhado na seção 3.1, e, a seguir, utiliza técnicas de testes de consistência do campo de vetores de velocidade obtido e multiresolução, técnicas essas baseadas no trabalho de Proesman et al. [1994] e detalhadas na seção 3.2.

Para que se obtenha um sistema eficiente, será dada preferência ao uso de técnicas de baixo custo computacional. No entanto, o ganho de desempenho não deve resultar em perda de precisão além do limite no qual a imprecisão da direção impossibilita "prever" o que seria a continuidade do movimento.

Em Barron et al. [1994] é feita uma análise quantitativa de diversos tipos de algoritmos para a análise de movimento. Laplante e Stoyenko [1997] descrevem as melhores técnicas para serem aplicadas em tempo real. As técnicas nestes referidos trabalhos inspiraram a presente proposta.

3. Análise de Movimento

O movimento pode ser analisado por meio da aplicação da função de Fluxo Ótico [Horn, 1986]. A função de Fluxo Ótico extrai de uma seqüência de imagens a velocidade e a direção do movimento dos *pixels* desta imagem. Fluxo Ótico é um termo largamente usado para uma grande variedade de ocasiões [Nalwa, 1994]. Neste artigo dois conceitos diferentes serão utilizados: Fluxo Ótico e Campo de Movimento.

De acordo com Horn [1986, pg 278]: "O movimento aparente de padrões de brilho observados quando uma câmera está se movendo relativamente aos objetos sendo filmados é chamado Fluxo Ótico". Segundo Nalwa [1994, pg 251]: "Campo de Movimento é um campo de vetores, definidos sobre a imagem, na qual cada vetor associado a um ponto da imagem representa o movimento na imagem do ponto da cena que se projeta naquele ponto". É um conceito puramente geométrico, atribuir um vetor de velocidade para cada ponto da imagem [Horn, 1986].

Dos conceitos acima conclui-se que Fluxo Ótico e Campo de Movimento não diferem em seus princípios; idealmente o resultado obtido da aplicação destes conceitos deveria ser igual [Horn, 1986].

O problema principal na análise do movimento é a estimação das componentes tridimensionais do movimento dos objetos sendo observados. Isto é de grande relevância em muitos problemas como reconstrução 3D ([Spies et al., 2001]), rastreamento de objetos ([Fuentes and Velastin, 2001], [Neumann and You, 1999] e [DeCarlo and Metaxas, 2000]), navegação de robôs ([Stephan et al., 2000] e [Kröse et al., 1999]) e rastreamento em tempo real ([Benoit and Ferrie, 1996]). A informação disponível em um sistema baseado em análise de movimento é relacionada com a projeção da velocidade 3D real no plano da imagem.

Embora os valores dos *pixels* não sejam estáveis no decorrer do tempo, considerando-se a grande quantidade de mudanças que podem ocorrer na imagem ou na geometria da imagem, eles podem ser considerados estáveis em mudanças suficientemente pequenas. A equação de restrição de gradiente considera que a intensidade de um ponto da imagem, enquanto ele se move, permanece constante. Assim, $f(x + dx, y + dy, t + dt) = f(x, y, t)$, na qual $f(x, y, t)$ é a intensidade da imagem. Uma expansão em séries de Taylor de $f(x, y, t)$ leva a $f(x + dx, y + dy, t + dt) = f(x, y, t) + \frac{\partial f}{\partial x}dx + \frac{\partial f}{\partial y}dy + \frac{\partial f}{\partial t}dt + O(x^2)$.

Ignorando os termos de ordem alta e igualando $f(x+dx, y+dy, t+dt) = f(x, y, t)$ resulta em $-\frac{\partial f}{\partial t} = \frac{\partial f}{\partial x}\frac{dx}{dt} + \frac{\partial f}{\partial y}\frac{dy}{dt}$ (3), a qual provê uma restrição na velocidade da imagem $(u, v) = (dx/dt, dy/dt)$ como uma função das derivadas locais $(Ex, Ey, Et) = (df/dx, df/dy, df/dt)$ da função de intensidade da imagem f. A equação 3 é conhecida como Equação de Restrição de Fluxo Ótico.

Abordagens baseadas em conceitos geométricos, que extraem o Campo de Mo-

vimento, utilizam-se da identificação de um conjunto de características esparsas e facilmente identificáveis dos objetos em movimento. Pelo rastreamento destas características, uma correspondência entre os quadros é buscada para estimar o movimento das características no plano da imagem. Essas características facilmente identificáveis podem ser linhas, formas, padrões, curvaturas, etc. Estas técnicas são muito úteis para rastrear objetos em movimento no espaço tridimensional, porque as características rastreadas podem ser partes relevantes do objeto. Mas o rastreamento destas características requer uma fase de pré-processamento (como filtragem, extração de bordas) nas imagens, com os correspondentes custos computacionais. Além disso, essas técnicas devem ser independentes de rotação e escala.

Três principais dificuldades afetam o problema de estimação do campo de vetores, seja ela feita através de conceitos geométricos ou utilizando a equação de restrição:

- Descontinuidades no campo que são originados pela presença de ruído no brilho da imagem;
- A presença de oclusões entre diferentes objetos em movimento (que usualmente têm diferentes velocidades) e entre os objetos em movimento e o fundo estático;
- O problema de *aperture* que é relacionado com a impossibilidade de recuperar a direção do movimento se o objeto é observado através de uma abertura que é menor que o tamanho do objeto, e as referências ao objeto sob observação (como textura) não são suficientes para perceber a componente transversal do movimento do objeto.

A forma mais comum de se representar o campo de vetores de velocidade extraído é o *needle map*. O *needle map* é uma figura com a representação dos vetores de todos os pontos da imagem para os quais foi calculado a função. Exemplos de *needle maps* são mostrados na seção 4.

3.1. Movimento Baseado em Correspondência de Regiões

A extração de vetores de velocidade baseada em correspondência de regiões determina o movimento correto de uma janela de *pixels* simulando o movimento da janela para cada posição $[x, y]$ e considerando qual a maior similaridade entre os valores das janelas nos quadros no instante *t* e *t+1* [Camus, 1997]. Se ϕ representa uma função que retorna um valor proporcional à similaridade das duas janelas em dois quadros diferentes (dada, por exemplo, pela como a Soma das Diferenças Absolutas dos valores de intensidade dos *pixels* - SAD), então a correspondência $M(x, y, s, w)$ da janela no ponto $[x, y]$ e todas aquelas nos deslocamentos (s, w) são calculadas. Dentre todos os deslocamentos possíveis, aquele que minimizar o critério da função ϕ atribuirá sua direção e módulo ao vetor de velocidade do *pixel* $[x, y]$:

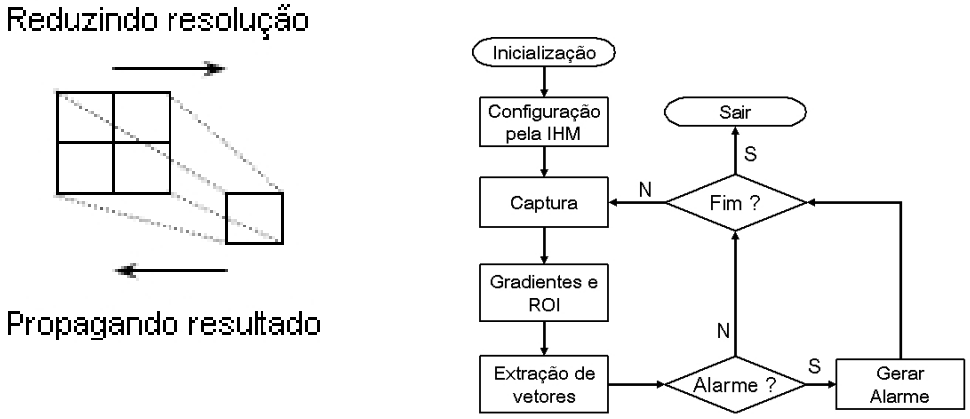
$$\vee(u, w) : M(x, y, s, w) = min(\sum \phi(E_1(i, j) - E_2(i + s, j + w))), (i, j) \in Pj, (s, w) \in Pb$$

Na qual Pj é a janela definida e Pb é a região de busca.

Este é um algoritmo bastante robusto já que não requer que os valores de intensidade dos *pixels* entre os quadros tenham valores próximos. Por exemplo, se houver uma mudança de iluminação certamente o valor de ϕ mudaria para todos os valores de (u, w) mas isso não afetaria o resultado da melhor combinação.

3.2. Consistência e Multiresolução

Um teste de consistência foi definido para refinar os dados obtidos pelo algoritmo Baseado em Correspondência de regiões (BCR), e a mutiresolução amplia a área de busca



(a) (b)

Figura 3: (a) Exemplificação da multiresolução em 1 nível. (b) Funcionamento geral do algoritmo para uma câmera.

do mesmo. Tendo-se os quadros a e b de uma seqüência de vídeo, pode-se imaginar que o campo de vetores de velocidade obtido considerando-se a seqüência direta (a depois b) e a seqüência reversa (b depois a) seria igual, exceto, que os vetores teriam direções opostas. Mas o que acontece realmente é que nas vizinhanças das descontinuidades, esse esquema dual tende à mostrar inconsistências. Usando os vetores de diferença [Proesman et al., 1994]

$$C_a = v_a(x, y) + v_b(x - u_a\Delta t, y - v_a\Delta t)$$

e

$$C_b = v_b(x, y) + v_a(x - u_b\Delta t, y - v_b\Delta t)$$

pode-se definir as medidas de inconsistência C_a e C_b que podem ser realimentadas para o esquema dual.

Outra técnica utilizada é a multiresolução, proposta por Proesman et al.[1994] para permitir a ampliação da máxima velocidade detectável pelo algoritmo de extração de vetores de velocidade. Por exemplo, ao se utilizar uma imagem de entrada com tamanho 320x240 e somente um nível de resolução extra, logo que a imagem de entrada é recebida o algoritmo a reduz à metade da sua resolução vertical e horizontal, ficando então, no exemplo com tamanho 160x120. Nesta resolução mais baixa é feita então a primeira extração dos vetores de velocidade. Esse resultado é repassado para a extração com a resolução mais alta, sendo que cada *pixel* na resolução inferior torna-se quatro na resolução superior. A extração na resolução superior começa então assistida pelo resultado obtido anteriormente, o que também produz um melhor refinamento do resultado. A Figura 3 (a) exemplifica a redução da resolução (direita para a esquerda) na qual é feita a média do valor da intensidade dos quatro *pixels* vizinhos que serão reduzidos a um pixel, e no sentido contrário a propagação de resultados (esquerda para a direita) na qual o vetor de velocidade de [x,y] é copiado para os quatro *pixels* correspondentes, nesta cópia o módulo da velocidade é multiplicado por dois já que com uma maior resolução as distâncias em *pixels* dobram.

3.3. Algoritmo Proposto

Foi escolhido um algoritmo base que se encaixa dentro dos que têm potencial para sistemas de tempo real segundo estudo comparativo realizado por Laplante e Stoyenko [1997].

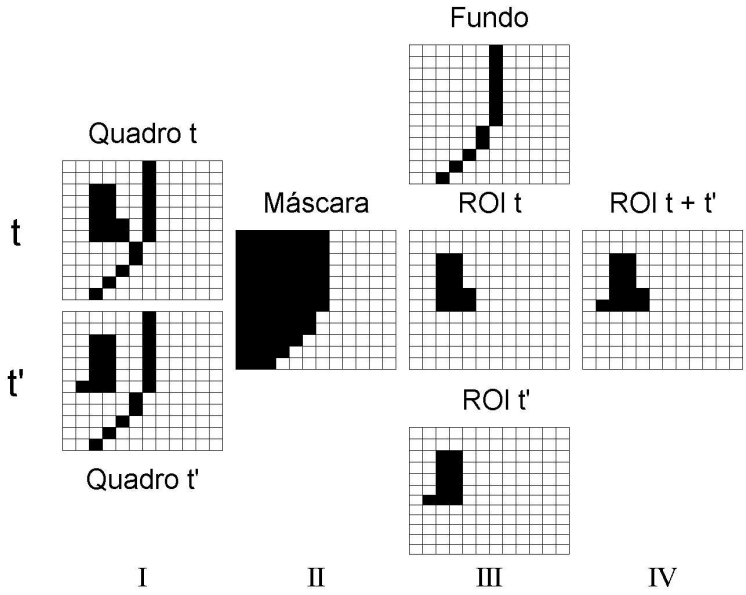


Figura 4: Passos para a obtenção da região de interesse

A implementação utiliza o algoritmo BCR para extração do campo de vetores de velocidade e o teste de consistência nas descontinuidades e a multiresolução (descritos no item 3.2), para refinamento do resultado e ampliação da área de busca, respectivamente. Outro motivo para que seja usado tanto o teste de consistência nas discontinuidades quanto a multiresolução é que pares de imagens isolados degradam o resultado ([Nalwa, 1994, pg 252]).

Para o algoritmo BCR utiliza-se janelas de tamanho 3x3 para dar eficiência ao algoritmo. Entretanto, a máxima velocidade detectável utilizando-se uma janela 3x3 é de 2 *pixels* entre imagens. Se for utilizado um subnível de resolução, a máxima velocidade detectável dobra para 4 *pixels* entre imagens, juntamente, é claro, com um maior custo computacional a ser considerado. Deve-se ressaltar que o algoritmo BCR consegue detectar velocidades *subpixel* mas para isso precisa de mais do que duas imagens. Como o domínio da aplicação não necessita de tal precisão, optou-se por utilizar apenas duas imagens para cada extração do campo de vetores e, portanto, a mínima velocidade detectável é de 1 *pixel* na alta resolução.

Na figura 3 (b) é mostrada uma visão geral do do algoritmo proposto. Após a configuração do fundo estático do ambiente e da máscara que define a área que inicialmente será processada, o algoritmo entra em um ciclo no qual é feita a extração dos gradientes no tempo e no espaço e da região de interesse (ROI) em dois quadros consecutivos. É importante observar que é o teste de consistência feito nas descontinuidades que precisa dos gradientes no tempo e espaço e não o algoritmo BCR. O próximo passo é a extração dos vetores de velocidade dentro da ROI e a análise deste resultado para que um alarme seja ou não gerado.

A definição da ROI visa diminuir a quantidade de pixels a ser processada, agilizando o sistema de monitoramento. Para a definição da ROI, os seguintes passos são executados:

1. Isolamento da área de risco através do uso de uma máscara;
2. Subtração do fundo que determina a região de interesse a ser processada;
3. Exclusão de *pixels* isolados da região de interesse através do uso de operadores morfológicos *open* e *close* ([Gonzalez and Woods, 1992]).

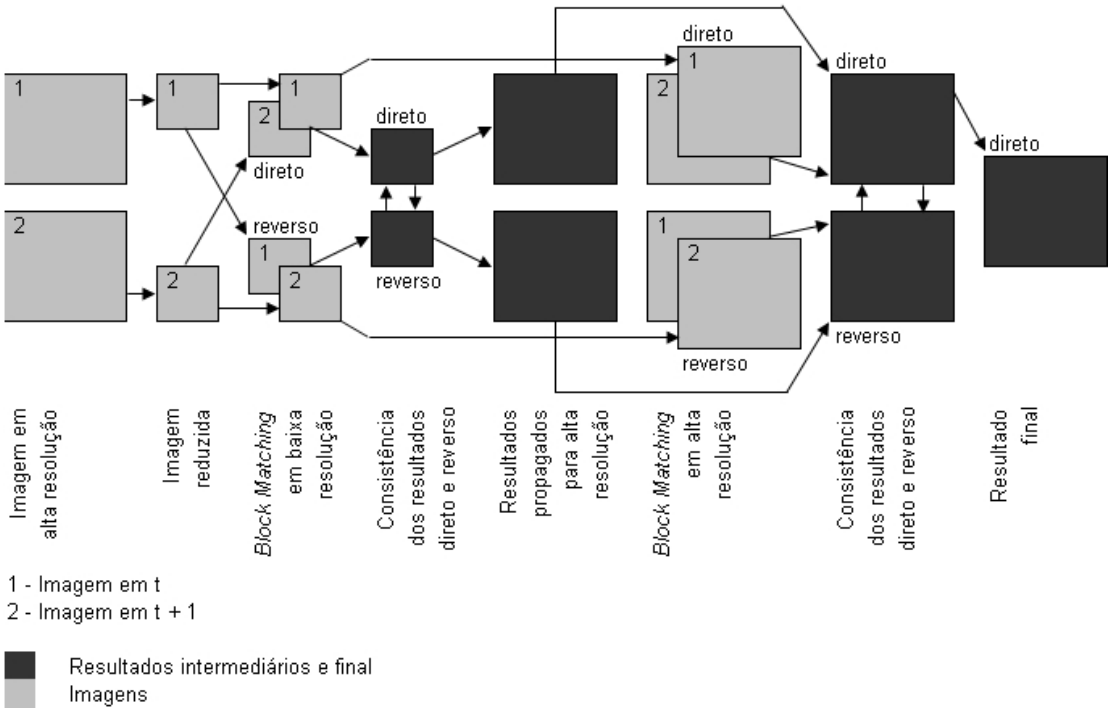


Figura 5: Algoritmo de extração de vetores de velocidade

A Figura 4 ilustra, com imagens binárias, as etapas necessárias para a obtenção da ROI: a etapa I representa a aquisição dos quadros; a etapa II, através da operação lógica *AND*, elimina a região fora da máscara, definida *a priori* pelo operador; na etapa III é subtraído o fundo da imagem para que se obtenha os objetos de interesse; e a etapa IV, através da operação lógica *OR*, obtém a área da região total de interesse que será utilizada no cálculo do campo de vetores de velocidade.

A figura 5 exemplifica o funcionamento do algoritmo em etapas após a obtenção da ROI. Inicialmente tem-se duas imagens de dois quadros consecutivos de uma sequência de vídeo, uma em t e outra em $t+1$. Para cada uma das imagens as etapas são as mesmas tendo-se que os dados do resultado (campo de vetores de velocidade) depende das duas. Primeiro é feita uma redução da resolução, resultando em imagens com $1/4$ da dimensão original. Em seguida é aplicado o algoritmo BCR direto e reverso para se obter dois resultados que, a princípio, deveriam ser, para todos os vetores de todos os pixels, iguais em módulo mas com direções opostas. O resultado direto é então consistido utilizando-se o resultado reverso conforme descrito na seção 3.2. Após esse teste de consistência o resultado é então copiado para a matriz de vetores de velocidade da resolução mais alta, sendo que o módulo dos vetores é multiplicado por dois, já que as distâncias dobram nesta resolução. Esse resultado é então utilizado para o teste de consistência dos resultados direto e reverso da resolução mais alta antes de ser feita a consistência entre os resultados direto e reverso. O resultado final é então o resultado direto consistido com o reverso na alta resolução.

4. Resultados Experimentais

Os resultados obtidos (com imagens fornecidas pelo metrô de São Paulo) são mostrados comparando os dados adquiridos com o uso do algoritmo BCR com e sem o uso dos testes de consistências nas descontinuidades e multiresolução. Nos dois casos a amostragem espacial utilizada é de 320x240 *pixels* com 256 níveis de cinza, amostragem temporal de

(a) (b)

Figura 6: (a)Fundo a ser subtraído e área delimitada pela máscara. (b)Exemplo de uma região de interesse obtida com limiar de subtração de fundo igual a 20.

200 ms entre capturas. O limiar de subtração de fundo utilizado para os dois casos tem valor igual a 20 (valores de níveis de cinza abaixo de 20 são eliminados).

Para os testes foi utilizado um microcomputador Pentium III 933 MHz com 128 MB de memória RAM. A seqüência de imagens utilizada é uma seqüência sintética de um homem se movendo sobre uma sequência de vídeo real de uma plataforma de trens subterrâneos.

A Figura 6 (a) mostra um exemplo de uma figura que tem a cena fundo já dimensionado apenas dentro da área da máscara. Esse fundo é subtraído das imagens que são capturadas, e a área que nunca deve ser processada (lado direito), é inteiramente branca (área ignorada). Esse é um exemplo de configuração na qual o lado direito da imagem, que tem a plataforma, nunca será processado, e sim a área na qual pode ocorrer movimentação de risco (lado esquerdo da imagem). Na figura 6 (b) é mostrado um exemplo de uma ROI obtida com limiar de subtração de fundo igual a 20.

4.1. Caso 1 - Resultados BCR

A seguir são apresentados os resultados do algoritmo de *block matching* sem a utilização das consistências e da multiresolução (caso 1).

Os resultados dos *needle maps* mostrados na Figura 7 apresentam coerência na direção do movimento detectada em toda a área da ROI, sendo que maioria dos vetores de velocidade têm a mesma direção, e a detecção de movimento próximo às bordas não é prejudicada.

A Figura 7 apresenta uma sequência de seis *needle maps* calculados um em seguida do outro através dos dados de entrada. Para este teste foram utilizados apenas dois quadros para o cálculo do campo de vetores, fato esse que prejudica a precisão da direção dos vetores obtidos, mas que melhora o desempenho. Para testes com poucos quadros por cálculo de campo de vetores é fácil observar que muitos vetores apresentam variações de 45 graus entre eles, isso por conta da falta de refinamento na informação da direção, informação esta que pode ser melhorada através da utilização de mais quadros (com prejuízo de desempenho).

O desempenho do algoritmo BCR, atuando isoladamente e com o uso dos valores indicados dos parâmetros, é acima do necessário pelo domínio da aplicação. Porém, não se pode esquecer que o sistema deverá ser capaz de diferenciar o movimento de uma pessoa ou objeto de um trem, mesmo quando os dois estiverem movimentando-se simultaneamente. Outro fato que justifica uma maior preocupação com o refinamento das

Figura 7: Needle Map de Camus: Método = NCC

direções obtidas nos resultados é que não se pode gerar alarmes a todo momento por falta de confiança na precisão do algoritmo.

		Processando todo o quadro	Processando somente a ROI
Função	Número de imagens por extração	ms/extração e extrações/segundo	ms/extração e extrações/segundo
SAD	2	690 e 1,45	85 e 11,76
SAD	3	1360 e 0,73	165 e 6,06
SAD	4	2025 e 0,44	200 e 5,00

Tabela 1: Desempenho do algoritmo de BCR

Na Tabela 1 são apresentados alguns resultados para o algoritmo BCR utilizando o método SAD como função ϕ.

Sum of Absolute Differences

$$SAD(\partial x, \partial y) = \sum_{(x,y)} |(E(x,y,t)) - E(x+\partial x, y+\partial y, t+\partial t)|$$

A primeira coluna mostra o método utilizado, a segunda coluna apresenta o número de imagens da seqüência que foram utilizadas para extrair os vetores de velocidade, e a terceira e quarta colunas mostram o desempenho em milisegundos gastos para extrair os vetores e seu inverso, indicando que seria quantos grupos de imagens por segundo poderiam ser processadas, respectivavente sem e com a ROI. Pode-se observar que, a cada imagem adicionada para o refinamento, aproximadamente o tempo gasto para extração do campo de vetores de um par de imagens é adicionado ao tempo total de extração. Quanto maior o número de quadros que são utilizados em uma estimação melhor é o resultado do campo de vetores de velocidade, mas observa-se uma queda no desempenho.

Para que o sistema torne-se mais confiável foram adicionadas outras características a esse algoritmo que são mostradas a seguir.

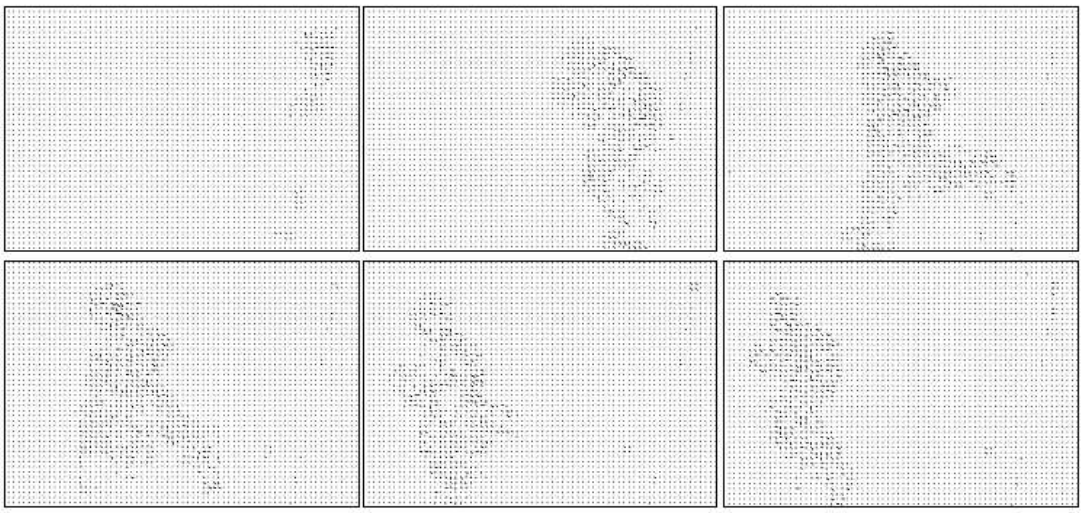


Figura 8: *Needle Map* do algoritmo de *block matching* multiresolução: Método = SAD com '1' subnível de resolução

4.2. Caso 2 - BCR multiresolução

A seguir são apresentados os resultados do algoritmo de BCR com a utilização das consistências e da multiresolução.

Os resultados dos *needle maps* para o caso 2 são mostrados na Figura 8 apresentam vetores com menor módulo do que os apresentados nos resultados do caso 1, mas esses vetores têm maior consistência em seus resultados e apresentam maior precisão na direção detectada do movimento. Essa redução no módulo dos vetores se deve ao fato que estão sendo utilizados apenas números inteiros no algoritmo de extração de vetores de velocidade, o que, depois de algumas iterações, têm esse efeito.

A Figura 8 apresenta uma sequência de seis *needle maps* calculados um em seguida do outro. A redução no módulo dos vetores não afeta o desempenho do sistema, que precisa apenas que a direção seja corretamente estimada para que se possa gerar o alarme com confiabilidade. Nesse caso o algoritmo utiliza técnicas que, tanto corrigem a estimação nas proximidades das descontinuidades, quanto ampliam a área de busca do algoritmo testado anteriormente, o que aumenta essa confiabilidade.

A Tabela 2 mostra na primeira coluna o método utilizado como função phi, a segunda coluna apresenta o número de imagens da seqüência que foram utilizadas para extrair os vetores de velocidade, e a terceira coluna mostra o desempenho em quantos milisegundos gastos para extrair os vetores e o inverso disso que seria quantos grupos de imagens por segundo poderiam ser processadas, utilizando a ROI.

Pode-se notar na Tabela 2 que o desempenho para uma mesma quantidade de imagens por extração é um pouco inferior ao desempenho do mesmo algoritmo sem nenhum teste de consistência ou melhoria na área buscada (multiresolução). Essa queda no desempenho não apresenta grande problema para o propósito do algoritmo já que se está tendo um resultado muito mais refinado, no que diz respeito à direção dos vetores de velocidade. Levando-se ainda em consideração que o computador utilizado para os testes não possui *hardware* especializado, e nem se trata de um computador que agrega as mais recentes tecnologias, pode-se esperar um ganho considerável no desempenho do algoritmo somente com a troca do hardware.

		Processando a ROI
Função	Número de imagens por extração	ms/extração e extrações/segundo
SAD	2	220 e 4,54
SAD	3	260 e 3,84
SAD	4	290 e 3,44

Tabela 2: Desempenho do algoritmo de *block matching* multiresolução

5. Conclusão

É esperado que a utilização das simplificações em conjunto com os testes de consistência e a multiresolução possa prover ao sistema a velocidade e a qualidade necessárias para atuar como detector de riscos, e que assim o sistema possa melhorar os serviços prestados em estações de trens.

Atualmente, o algoritmo proposto, com otimizações de desempenho e testes de consistências de descontinuidades, consegue extrair o campo de vetores de velocidade de aproximadamente quatro (4) pares de quadros por segundo, em um sistema monoprocessado. É esperado que o desempenho do algoritmo final fique entre oito (8) e dez (10) pares de quadros por segundo. É importante lembrar que os testes ainda estão sendo feitos em sistemas monoprocessados e que uma implementação *multithread* ainda será testada, sendo que o *hardware* utilizado influencia, e muito, desempenho final.

A redução da área de processamento reduz drasticamente o tempo gasto para processar as imagens capturadas por uma câmera de monitoramento, em alguns casos reduzindo este processamento a valores próximos de zero, no que diz respeito à extração do campo de vetores de velocidade, uma vez que a área a ser processada pode ser anulada pela subtração de fundo em um determinado instante, o que possibilita que mais câmeras de monitoramento sejam usadas e menos tempo seja gasto entre as capturas.

O algoritmo ainda deve ser testado em situações nas quais diferentes objetos se movimentam, cruzando suas trajetórias e causando oclusão parcial ou total de um (ou mais) deles. Além disso, uma análise mais rigorosa do resultado também será realizada.

O sistema final deverá se adaptar à rotina do operador de CFTV e não o contrário. Deve ser suficientemente prático, para facilitar sua configuração, deve evitar a necessidade de recalibragem, e deve ser testado ao máximo para evitar a geração de alarmes sem que um evento de risco tenha ocorrido.

Referências

Barron, J. L., Fleet, D. J., and Beauchemin, S. S. (1994). Performance of optical flow techniques. *IJCV*, pages 43–77.

Benoit, S. M. and Ferrie, F. P. (1996). Monocular optical flow for real time vision systems. Technical report, McGill University, Montréal, Québec, Canada.

Camus, T. (1997). Real-time quantized optical flow. *Journal of Real-Time Imaging*, 3:71–86.

DeCarlo, D. and Metaxas, D. (2000). Optical flow constraints on deformable models with applicatins to face tracking. *International Journal of Computer Vision*, pages 99–127.

Fuentes, L. M. and Velastin, S. A. (2001). People tracking in surveillance applications. In *2nd IEEE Int. Workshop on PETS*, Kauai, Hawaii, USA. IEEE Computer Society Press.

Gonzalez, R. C. and Woods, R. E. (1992). *Digital Image Processing*. Addison Wesley Publishing Company, Boston, MA.

Horn, B. K. P. (1986). *Robot Vision*. Massachusetts Institute of Technology, Cambridge, Cambridge, MA, USA.

Kröse, B. J. A., Dev, A., and Groen, F. C. A. (1999). Heading direction of a mobile robot from the optical flow. Technical report, University of Amsterdam.

Laplante, P. A. and Stoyenko, A. D. (1997). *Real Time Imaging: Theory, Techniques, and Applications*. IEEE Computer Society Press, Los Alamitos, CA.

Nalwa, V. S. (1994). *A Guided Tour of Computer Vision*. Addison-Wesley Longman, Boston, MA, USA.

Neumann, U. and You, S. (1999). Integration of region tracking and optical flow for image motion estimation. Technical report, University of Southern California.

Proesman, M., Gool, L. V., Pawels, E., and Oosterlinck, A. (1994). Determination of optical flow and its discontinuities using non-linear diffusion. In *3rd European Conference on Computer Vision, ECCV'94*, volume 2, pages 295–304.

Seixas, M. O. and Costa, A. H. R. (2003). Estimativa de um modelo adaptativo de cena de fundo para segmentação em tempo real de objetos de interesse. In ENIA, editor, *Anais do XXIII Congresso da Sociedade Brasileira de Computação, Volume VII*, pages 31–40.

Spies, H., Jähne, B., and Barron, J. L. (2001). Dense range flow from depth and intensity data. Technical report, Interdisciplinary Center for Scientific Computing, HeidelBerg, Germany.

Stephan, V., Winkler, T., and Gross, H.-M. (2000). Fast and robust prediction of optical flow field sequences for visuomotor anticipation. In *IJCNN'2000*, pages 436–441, Como, Italy.