

# Algoritmos de Subtração Básica de Fundo

**Leonardo Ronald Perin Rauta**  
leonardorauta@univali.br  
UNIVALI

**Anita Maria da Rocha Fernandes**  
anita.fernandes@univali.br  
UNIVALI

**Resumo:** Algoritmos de subtração de background são de suma importância para o processamento de vídeo, pois com o resultado desses algoritmos é possível desenvolver diversas aplicações nas mais diversas áreas do conhecimento. Este artigo apresenta um overview sobre alguns dos algoritmos de separação de background disponíveis na literatura, suas características, seu funcionamento e também sua complexidade. Os algoritmos apresentados são: Subtração básica de fundo estimado pela média, Subtração básica de fundo estimado pela mediana, Subtração básica de fundo estimado pela média adaptativa e Subtração básica de fundo estimado pela mediana de um intervalo. Os algoritmos apresentados tratam-se de algoritmos de subtração básica, dois adaptativos e dois não adaptativos. Testes feitos com esses algoritmos apresentam que o algoritmo com menor tempo de processamento foi o algoritmo da média adaptativa e o algoritmo que apresenta menor quantidade de falsas detecções é o algoritmo de subtração básica de fundo estimado pela mediana de um intervalo. Isso fica comprovado através da análise da complexidade dos algoritmos implementados.

**Palavras Chave:** Segmentação - Background - Processamento - Imagens - Subtração de fundo

## 1. INTRODUÇÃO

A utilização de sistemas computacionais que atuem de forma semelhante à visão humana vem crescendo largamente devido ao número de aplicações que esse tipo de sistemas pode oferecer. Uma das primeiras etapas do processamento de imagens para sistemas de visão computacional é a separação de *background* e *foreground*, ou seja, separar o que faz parte do fundo da cena, do que não faz parte do fundo da cena. Por este motivo é uma etapa muito importante no desenvolvimento da aplicação (GAGVANI, 2008).

A escolha de algoritmos de separação de *background* não é uma escolha simples, pois os diferentes algoritmos existentes para esta finalidade possuem comportamentos bastante distintos.

Neste artigo são apresentados alguns conceitos básicos sobre alguns algoritmos de subtração básica de fundo (*background*). Os algoritmos a serem tratados neste artigo são os algoritmos em que a imagem de *background* é estática, como os algoritmos de subtração básica de fundo pela média e subtração básica de fundo pela mediana, e alguns algoritmos adaptativos, como o algoritmo de subtração básica de fundo estimado pela média adaptativa e subtração básica de fundo estimado pela mediana de um intervalo. Também serão apresentadas as principais características de cada algoritmo, como as equações para cálculo do *background* e também a complexidade desses algoritmos. Como todos os algoritmos apresentados possuem um vetor para armazenar os  $n$  frames de profundidade, a complexidade apresentada nesse artigo é em relação à utilização desse vetor de profundidade.

## 2. SUBTRAÇÃO DE FUNDO

Algoritmos de subtração de fundo também são conhecidos como algoritmos de segmentação de *background* ou algoritmos de subtração de *background*. Essa técnica consiste em separar o que faz parte do fundo do que não faz parte do fundo, separar o *background* do *foreground*.

A subtração de fundo é um dos métodos mais comuns de segmentação principalmente em situações em que o fundo é relativamente estático. A subtração básica de *background* consiste basicamente em selecionar ou então compor um *frame* que servirá como fundo da imagem, o qual não pode conter objetos em movimento ou qualquer outro tipo de objeto que não pertença ao fundo. Em seguida, é calculada a diferença *pixel a pixel* entre o *frame* atual e o *frame* de referência. Caso esta diferença seja menor que um limiar definido, esse *pixel* pertence ao fundo, caso contrário, pertence ao objeto detectado (FERREIRA, 2007).

Para obter o *background* de uma imagem existem duas soluções relativamente simples e bastante utilizadas: o fundo ser apenas de uma cor, *Chroma Key* – largamente utilizado em efeitos especiais de filmes e novelas; ou tirar uma foto do fundo sem nenhum objeto e usar esta foto para subtrair o fundo. No entanto, nenhuma dessas opções pode ser utilizada em ambientes não controlados, pois a imagem de fundo não se adapta, é uma imagem de fundo estática (LARA, 2006).

Existem inúmeras abordagens, as quais se diferem quanto ao modelo de *background* e o processo utilizado para atualizar o modelo de fundo. Existem pesquisas sobre a adaptação de modelos de *background* para redução nas mudanças dinâmicas na cena para segmentação de movimentos (WANG *et al.*, 2003).

Uma modelagem simples para compor o *background* é utilizar a média ou a mediana, uma aproximação do *background* que é similar à cena estática corrente. Outra modelagem simples é estimar o fundo através da média adaptativa ou então pela mediana de um intervalo, o que faz com que o *background* se adapte às mudanças ocorridas no cenário.



Muitos algoritmos de subtração de fundo não tratam as imagens com suas cores originais, elas são transformadas em tons de cinza. Uma das técnicas para transformar uma imagem em tons de cinza é a das médias ponderadas para os diferentes espectros de cores, conforme a Equação 1.

**Equação 1:** Equação da transformação para tons de cinza  
$$\text{Gray} = -0.299 * R + 0.587 * G + 0.114 * B$$

## 2.1. SUBTRAÇÃO BÁSICA DE FUNDO ESTIMADO PELA MÉDIA

Técnica apresentada por Lo e Velastin (2001), a qual consiste em utilizar os  $n_i$  primeiros frames para estimar o fundo a partir da média. Com isso foi obtido o *frame* para ser utilizado como referência para subtração do fundo. A Equação 2 apresenta como é feito esse cálculo da média, em que  $n$  é o número de quadros da sequência  $1 \leq n_i \leq n$ .

**Equação 2:** Equação do cálculo da média

$$B(x) = \frac{1}{n} \sum_{i=1}^{n_i} f_i(x)$$

Para identificar um alvo que não pertença ao fundo é determinado um limiar  $\tau$  e então é feito o cálculo da diferença simétrica do quadro atual e do quadro calculado como *background*. A Equação 3 apresenta como é feito este cálculo, em que  $A_i(x)$  é o alvo a ser identificado,  $f_i(x)$  o frame atual e  $B(x)$  o frame calculado como fundo. Essa diferença é feita *pixel a pixel* nos frames.

**Equação 3:** Equação para cálculo do alvo

$$A_i(x) = |f_i(x) - B(x)| > \tau$$

Este modo de estimar o fundo é muito rápido, no entanto o seu desempenho depende muito de estar em um ambiente controlado sem alterações discrepantes no ambiente, como iluminação por exemplo. Além de ser um algoritmo que o seu desempenho seja dependente do limiar  $\tau$  escolhido e o modelo do *background* seja estático, é um algoritmo utilizado para determinar qual será o frame a ser escolhido como *background*, geralmente são os frames iniciais da captura.

Esta técnica de estimação de fundo consiste em duas etapas importantes, a etapa de adaptação do *background* e a etapa do cálculo do limiar. Na etapa de adaptação são capturados os  $n_i$  frames para compor o *background*. Já a outra etapa consiste apenas no cálculo do alvo.

A Figura 1 apresenta um exemplo do funcionamento desse algoritmo no frame 250, levando em consideração um limiar  $\tau = 10$  e uma média de 20 frames em um ambiente não controlado. O valor do limiar e da quantidade de frames foram escolhidos empiricamente.



**Figura 1:** Subtração básica de fundo estimado pela média,  $\tau = 10$  e média de 50 frames. a) Imagem original; b) Background composto em tons de cinza; c) Imagem subtraída: Imagem Original – Background Composto

Durante o processamento desse algoritmo é possível observar que o *background* permaneceu estático. Além disso, na Figura 1, é possível observar que essa técnica foi capaz de detectar todos os pontos em que existia movimento na cena, circulosados em vermelho. No entanto, foram detectados muitos falsos-positivos, ou pontos com ruídos, representados pelos pontos brancos na imagem “c)” e que não estão circulosados em vermelho.

Nos testes feitos em laboratório, com um ambiente controlado quanto à movimentação e iluminação, foi constatado que em um ambiente controlado a média pode ser utilizada com um valor próximo de 10, pois o ambiente não sofre muita alteração. Já para ambientes não controlados, como o da Figura 1, a média utilizada deve ser mais alta, pois o ambiente sofre variações, o que acaba prejudicando o *background*.

Durante a etapa de adaptação, a complexidade desse algoritmo é de  $3n+2$ , pois é necessário calcular qual o novo valor de média. Já para o cálculo do alvo, essa complexidade passa a ser 1.

Uma técnica utilizada para aumentar o tempo de captura das amostras e compor um *background* mais estável, é utilizar um *gap*, ou seja, a cada  $g$  frames, um é selecionado para fazer parte dos  $n$  frames que serão utilizados para compor o *background*. Quando se utiliza um *gap* a complexidade do algoritmo muda, pois o algoritmo agora possui três etapas, a de adaptação, o *gap* e o cálculo do alvo. Nesta etapa de *gap* a complexidade passa a ser 1, pois os *frames* são apenas descartados.

## 2.2. SUBTRAÇÃO BÁSICA DE FUNDO ESTIMADO PELA MEDIANA

Para estimar o fundo, Rossin e Ellis (1995) e Cucchiara, *et al.* (2001) utilizam a mediana. Esta abordagem consiste em calcular a mediana dos valores dos *pixels* nos  $n_i$  primeiros quadros da sequência  $1 \leq n_i \leq n$ , onde  $n$  é o número de quadros (Equação 4). A moda também pode ser utilizada como estatística para estimar o fundo.

**Equação 4:** Equação do cálculo da mediana

$$B(x) = \text{mediana}_{\{1 \leq i \leq n_i\}} f_i(x)$$

Do mesmo modo que a subtração básica de fundo estimado pela média, para identificar um alvo  $A_i(x)$ , é necessário definir um limiar  $\tau$  e é utilizada a Equação 3 para calcular o alvo. Esse algoritmo também possui duas etapas básicas de funcionamento, compor o *background* através da mediana, e calcular o alvo.

A Figura 2 apresenta um exemplo do funcionamento desse algoritmo, uma imagem capturada durante o cálculo do frame número 250, utilizando um  $\tau = 10$  e  $n = 50$ . Onde as marcações em vermelho representam os locais onde existia movimento e que foram detectadas pelo algoritmo. A imagem “a)” representa a imagem original, a imagem “b)” representa o



*background* gerado pelo algoritmo, e a imagem “c)”) representa a subtração entre a imagem original e a imagem de *background*.



**Figura 2:** Subtração básica de fundo estimado pela mediana,  $\tau = 10$  e número de frames = 50

Na Figura 2 foi possível observar que o *background* gerado foi muito semelhante ao *background* gerado pela técnica da média. O comportamento das duas técnicas são muito semelhantes em relação às detecções. No entanto, se diferem na complexidade de cada técnica.

Durante a etapa de adaptação, a complexidade desse algoritmo é de  $3n + 2 + x$ , em que  $x$  consiste no big(O) do algoritmo de ordenação que foi utilizado. Na etapa do cálculo do alvo a complexidade passa a ser 1, pois não é mais necessário utilizar o vetor de profundidade, ou seja, o vetor que comporta os  $n$  frames para calcular a mediana.

Do mesmo modo que a subtração básica de fundo estimado pela média, o número de quadros depende de quão controlado o ambiente se encontra. Por ser um algoritmo em que existe a necessidade de reordenar o vetor de frames recebidos a cada novo frame, é um algoritmo que necessita um poder de processamento maior que a subtração básica de fundo pela média, por este motivo é ideal utilizar um *gap* na captura dos frames.

Esta abordagem de estimação de fundo é mais robusta em relação a valores discrepantes, comparada com o algoritmo de subtração básica de fundo estimado pela média. No entanto as abordagens até então apresentadas não são adaptativas, isso significa que possuem um melhor desempenho se utilizadas em ambientes controlados.

### 2.3. SUBTRAÇÃO BÁSICA DE FUNDO PELA MÉDIA ADAPTATIVA

Com o objetivo de resolver alguns dos problemas dos algoritmos não adaptativos, McFarlane e Schofield (1995) e Horprasert; Haritaoglu e Wren (1998) desenvolveram um formato adaptativo para o modelo de fundo. Neste modelo adaptativo usa-se uma taxa de aprendizado  $\alpha$  que indica quanto o frame atual  $i$  influencia no modelo do fundo atual. O modelo de fundo  $B_i$  é inicializado com o primeiro quadro e os próximos quadros atualizam o modelo conforme a Equação 5, em que  $\alpha \in (0,1)$  e do mesmo modo que os algoritmos não adaptativos, o alvo é calculado conforme a Equação 3.

**Equação 5:** Cálculo do background pela média adaptativa

$$B_i(x) = \begin{cases} f_{i-1}(x) & \text{se } i = 2 \\ (1 - \alpha)B_{i-1}(x) + \alpha f_{i-1}(x) & \text{se } i > 2 \end{cases}$$

Mesmo sendo um algoritmo robusto em relação a alterações de iluminação e alterações no cenário, ele apresenta um problema: encontrar uma taxa de aprendizado  $\alpha$  que funcione bem com a sequência. Caso  $\alpha$  seja muito perto de zero, o fundo se adapta muito lentamente às alterações no cenário, ou seja, o cenário pode ser modificado e o modelo do fundo pode não corresponder a esse cenário, causando problemas como detectar falsos positivos durante longos períodos. Outro caso de mau funcionamento do algoritmo se dá quando  $\alpha$  é muito próximo de 1, em que o fundo adapta-se rapidamente às alterações no cenário e essa

velocidade na adaptação pode fazer com que partes dos alvos sejam perdidas por serem rapidamente consideradas como integrantes do fundo (FERREIRA, 2007).

Diferentemente dos algoritmos não adaptativos, essa técnica não deve utilizar um *gap*, primeiramente por ser um algoritmo que não necessita grande capacidade de processamento e também pela ideia principal do algoritmo, o *background* se adaptar a cada novo frame recebido.

A Figura 3 apresenta o funcionamento do algoritmo no frame de captura número 250, utilizando uma taxa de aprendizado  $\alpha = 0,2$ , escolhida empiricamente. Nessa figura as marcações em vermelho apresentam os locais onde existiam movimentações na cena e as marcações em amarelo apresentam os locais em que o algoritmo não considerou como estando em movimento. É possível observar que o algoritmo não detectou todos os locais onde existiam movimentos na cena, Isso ocorreu devido à taxa de aprendizagem utilizada no algoritmo. No entanto, o algoritmo não teve falsas detecções. As não detecções também podem ser explicadas pelo pouco movimento apresentado entre em um frame e outro.



**Figura 3:** Subtração básica de fundo estimado pela média adaptativa,  $\tau = 10$  e  $\alpha = 0,2$

Para esse algoritmo foram feitos testes variando o  $\alpha$  e o *threshold*. Tabela 1 apresenta a quantidade de ciclos de *clocks* gastos em cada teste feito sobre os diferentes *alphas* e diferentes *threshold*. Nessa tabela a coluna ‘Média’, ‘Mínimo’ e ‘Máximo’ correspondem à quantidade de ciclos gasta apenas no processamento de 1 *frame*. Já a coluna ‘Desvio Padrão’ corresponde ao desvio padrão médio de todos os *frames* analisados e também é apresentada em quantidade de ciclos de *clock*. Foram utilizados 1001 *frames* nas tomadas de tempo e os dados apresentados na tabela são as médias de trinta tomadas de tempo.

**Tabela 1:** Tabela de tempos gastos no processamento do algoritmo adaptativo de subtração básica de fundo estimado pela média adaptativa

$\alpha$	Threshold	Média	Mínimo	Máximo
0,3	10	18.770	10.000	40.000
0,5	10	18.770	10.000	40.000
0,8	10	18.690	10.000	40.000

Com base na Tabela 1 foi possível observar que o algoritmo sofre variação na quantidade de ciclos de *clocks* utilizadas no seu processamento quando o *threshold* é variado. A quantidade de ciclos cai, mas não é tão significativa essa diferença, pois o nível de processamento do computador utilizado chega na casa dos 3GHz, isso significa que 100 ciclos de *clock* seriam processados em 30 micro segundos. Um valor insignificante para o algoritmo, pois o tempo máximo gasto para processar um único *frame* gastou 40.000 ciclos, aproximadamente 130 mili segundos.

O desvio padrão apresentado na tabela apresenta que o algoritmo é bastante estável, pois a cada amostra recebida, o tempo de processamento varia em torno de 168 ciclos de



*clock*, aproximadamente 30 micro segundos de variação a cada *frame*. Um desvio padrão baixo significa que o algoritmo é bastante homogêneo quanto ao tempo de processamento. Neste artigo está sendo considerado um algoritmo estável quando o desvio padrão está abaixo de 250 ciclos de *clock*.

Em relação à complexidade do algoritmo, como ele não possui um vetor de profundidade, ou seja, a cada novo frame recebido ele recalcula o *background*, a complexidade é de 1. Pois variando a quantidade de frames recebidos o algoritmo terá a mesma velocidade de processamento. Isso ocorre pois a complexidade neste artigo está relacionada apenas ao vetor auxiliar que armazena os  $n$  frames para cálculo, não está relacionando o tamanho da imagem utilizada.

#### 2.4. SUBTRAÇÃO BÁSICA DE FUNDO PELA MÉDIA DE UM INTERVALO

Diferente do algoritmo de subtração de fundo pela média adaptativa, este algoritmo não necessita de uma taxa de aprendizado para adaptar-se às alterações no cenário e na iluminação.

Seja o inteiro  $z \in (1, n)$ , em que  $n$  é o número de quadros da sequência, calcula-se a mediana de cada *pixel*  $x$  dos  $z$  últimos quadros processados da sequência de imagens  $V$  conforme a Equação 6. O alvo  $A_i$  é calculado conforme a Equação 3.

**Equação 6:** Cálculo do background pela mediana de um intervalo

$$B_i(x) = \begin{cases} \text{mediana}_{1 \leq j \leq i} f(x) & \text{se } i - z < 1 \\ \text{mediana}_{i-z \leq j \leq i} f(x) & \text{se } i - z \geq 1 \end{cases}$$

Diferentemente do algoritmo adaptativo de subtração básica de fundo estimado pela média adaptativa que possuía uma taxa de aprendizagem para ser parametrizada, já na técnica do *background* estimado pela mediana, o que deve ser parametrizado é a quantidade de frames  $z$  que o algoritmo utilizará.

Além de parametrizar a quantidade de frames, outro parâmetro que deve ser levado em consideração neste algoritmo é o *gap* que será utilizado. Neste algoritmo o *gap* pode ser importante por se tratar de um algoritmo adaptativo, o que faz com que o *background* não se adapte tão rapidamente, pois um objeto pode permanecer em repouso por um curto espaço do tempo e esse passar a ser parte do fundo, dependendo da aplicação não é importante que isso aconteça.

A Figura 4 apresenta o comportamento desse algoritmo no frame de captura número 250, utilizando-se  $z = 20$ , sem *gap*, esses valores foram escolhidos empiricamente. As marcações representam as regiões onde existiam movimento e o resultado da detecção. É possível observar que todos os pontos onde existiam movimento foram detectados.

O algoritmo adaptativo de subtração básica de fundo estimado pela mediana de um intervalo precisa reordenar todos os *pixels* de  $n$  frames e pegar sempre o *pixel* central, mediana. Algoritmos de ordenação que sejam rápidos farão a diferença na aplicação, pois dependendo do algoritmo de ordenação escolhido o processo pode demorar muito mais que o desejado. Por este motivo foram avaliados alguns algoritmos clássicos de ordenação apenas quanto ao seu desempenho, avaliando seu tempo de processamento. Entre os algoritmos avaliados temos os algoritmos *Bubble Sort*, *Heap Sort*, *Insertion Sort*, *Merge Sort*, *Quick Sort*, *Selection Sort* e *Shell Sort*.



**Figura 4:** Subtração básica de fundo estimado pela mediana de um intervalo,  $\tau = 10$ , número de frames = 50 e gap de 10 frames

A Tabela 2 apresenta a quantidade de ciclos de processador gastos para o processamento de cada algoritmo, nessas tomadas de tempo não foi aplicado nenhum *gap* para avaliar a média de tempos de 501 *frames*, foram testados os algoritmos trinta vezes cada implementação com o mesmo intervalo  $n$ . Os dados presentes na tabela consistem na média das trinta tomadas de tempo.

**Tabela 2:** Tabela de tempos gastos no processamento e complexidade dos diferentes algoritmos de ordenação

Algoritmo	Tempo médio	Menor tempo	Maior tempo	Desvio padrão	Complexidade big(O)
<i>Bubble Sort</i>	738.900	590.000	1.040.000	13.331	$n^2$
<i>Heap Sort</i>	542.390	230.000	820.000	30.990	$n \log n$
<i>Insertion Sort</i>	358.850	125.000	540.000	45.938	$n^2$
<i>Merge Sort</i>	715.400	570.000	1.050.000	12.620	$n \log n$
<i>Quick Sort</i>	488.400	360.000	740.000	19.870	$n^2$
<i>Selection Sort</i>	463.510	380.000	660.000	33.461	$n^2$
<i>Shell Sort</i>	259.970	200.000	430.000	11.942	$n \log^2 n$

Com base na Tabela 2 é possível observar que o algoritmo de ordenação que teve um melhor desempenho em relação ao tempo de processamento foi o *Shell Sort* e também o algoritmo mais estável, pois apresenta também o menor desvio padrão médio de todos os algoritmos. Para ter uma ideia geral dos tempos gastos em segundos, 100.000 ciclos de *clock* representam 33 mili segundos. Como esse tempo é para o processamento de um único *frame*, 33 mili segundos por *frame* processado acabam acarretando um grande atraso para o processamento. No vídeo analisado a frequência do vídeo era de 30 *frames* por segundo, cada *frame* tem que ser processado em no máximo 33 mili segundos para não acabar atrasando o vídeo e possivelmente prejudicando a tomada de decisões em tempo real caso o algoritmo seja implementado em uma aplicação de tempo real.

A complexidade desse algoritmo é muito semelhante a complexidade do algoritmo de subtração básica de fundo estimado pela mediana, porém este algoritmo é adaptativo. A complexidade do algoritmo é  $3n + 2 + x$ , em que  $x$  consiste no big(O) do algoritmo de ordenação que foi utilizado.

### 3. CONCLUSÃO

Nesse trabalho foram apresentados alguns algoritmos de subtração básica de fundo, suas características e seu funcionamento. Foi possível observar que algoritmos não adaptativos não possuem um bom desempenho em ambiente não controlados, por isso geralmente são utilizados algoritmos adaptativos de subtração de *background*.



Através dos testes feitos com cada algoritmo foi possível observar que os algoritmos que estimam o *background* através de intervalos conseguem fazer com que o fundo seja o mais próximo possível do real, mas isso só se for utilizado um intervalo de *frames* muito grande, caso o intervalo seja pequeno, faz com que o fundo se adapte mais rapidamente, ficando uma espécie de rastro do objeto, ocasionando muitas vezes uma falsa detecção. Já o algoritmo da média adaptativa faz com que o *background* se adapte a cada novo *frame* capturado, sendo assim o fundo é sempre recomposto e muitas vezes o objeto em movimento acaba fazendo parte do fundo.

Quanto à quantidade de falsas detecções, os algoritmos que utilizam estimação através de um intervalo se mostraram mais estáveis, detectando mais objetos em movimento que o algoritmo da média adaptativa, esse se dá pelo fato de qual o método que compõem o fundo. A sensibilidade do algoritmo quanto à variação da iluminação é outro fator que depende como o algoritmo compõem o *background*, sendo assim, os algoritmos que utilizam estimação do fundo através de intervalos, se mostraram mais robustos quanto a variações de iluminação no ambiente.

Em todos os algoritmos implementados a escolha do *threshold* era muito importante para o funcionamento do algoritmo, pois ele determina o quanto o algoritmo é sensível às diferenças entre os *frames*, atual e *background*.

Na questão do processamento, foi possível observar através das complexidades que os algoritmos que utilizam *gap* não necessitam processar durante esta etapa, a complexidade permanece 1, fazendo com que a etapa mais custosa do algoritmo seja feita apenas em alguns *frames* e não em todos os *frames*.

Para trabalhos futuros, esses algoritmos que foram desenvolvidos podem ser otimizados, fazendo com que o tempo de processamento a cada novo *frame* seja reduzido. Além da otimização, aplicar esses algoritmos em diferentes ambientes é importante para comprovar que algoritmos adaptativos possuem um melhor comportamento em ambientes não controlados. Fazer a aplicação desses algoritmos em diferentes áreas pode ser uma boa pesquisa a ser feita com base nesses algoritmos, pois será possível observar seu comportamento em diferentes ambientes.

#### 4. REFERÊNCIAS

- Cucchiara, R.; Grana, C.; Piccardi, M.; Prati, A.** "Detecting objects, shadows and ghost in video streams by exploiting color and motion information." 11th International Conference on Image Analysis and Processing, Palermo, Itália, p. 360-365, 2001.
- Ferreira, M. A. L.** "Detecção de movimento através de subtração de fundo para vigilância eletrônica automática". Proposta de Tese (Doutorado) Doutorado em Informática, PUC-Rio, Rio de Janeiro, RJ. 2007.
- Gagvani, N.** "Introduction to video analytics." [S.I]. 2008. Disponível em: "<http://www.eetimes.com/design/industrial-control/4013494/Introduction-to-video-analytics>". Acesso em 14 jan. 2011.
- Horprasert, T.; Haritaoglu, I; Wren, C.; Harwood, D; Davis, L.; Pentland, A.** "Real-time 3D motion capture", Workshop on Perceptual User Interfaces. [S.I]. p. 87-90, 1998.
- Lara, A. C.** "Segmentação de Movimento para Aplicações de Vigilância Eletrônica." Monografia (Mestrado) Mestrado em Ciências da Computação, USP, São Paulo, SP, 2006.
- Lo, B.P.L. Velastin, S.A.** "Automatic congestion detection system for underground platforms". Proceedings of 2001 International Symposium on Intelligent Multimedia, Video and Speech Processing. Hong Kong, Japão. p. 158-161, 2001.
- McFarlane, N. J. B.; Schofield, C. P.** "Segmentation and tracking of piglets in images." Machine Vision and Applications, vol. 8, no. 3. p. 187-193, 1995.



**IX SEGeT 2012**

**SIMPÓSIO DE EXCELÊNCIA EM  
GESTÃO E TECNOLOGIA**

Tema: Gestão, Inovação e Tecnologia para a Sustentabilidade

**Rosin, P.; Ellis, T.** “Image difference threshold strategies and shadow detection.” Proc. British Machine Vision Conference, 1995.

**Wang, L.; Hu, W.; Tan, T.** “Recent developments in human motion analysis.” Pattern Recognition, 36 (3), 2003.