

# Segmentação de Alvos Móveis em Vídeo com Fundo Não-Estático Utilizando um Modelo de *Codebook* Modificado

Ronald Annoni Junior e Carlos Henrique Quartucci Forster

Instituto Tecnológico de Aeronáutica, Pça Mal. Eduardo Gomes, 50, 12228-900, São José dos Campos, SP

**Resumo** — A segmentação em tempo real de alvos móveis sobre planos de fundo não-estáticos, a partir de imagens produzidas por câmeras estáticas em áreas externas, sujeitas a variações de iluminação e movimentos espúrios durante longos períodos de observação requer métodos de modelagem e subtração de planos de fundo que sejam rápidos, com estrutura de armazenamento enxuta e sejam compatíveis com distribuições multimodais. Neste artigo é feita uma revisão dos principais métodos de modelagem e subtração de planos de fundo e é feita uma proposta de modificação do método de *codebook*, que atende aos requisitos expostos, proporcionando melhoras em sua estrutura de armazenamento e em seu desempenho computacional.

**Palavras-chaves** — inteligência de máquinas e robótica, segmentação de imagens, vídeo.

## I. INTRODUÇÃO

A extração de informações de interesse contidas em uma sequência de imagens de vídeo é uma operação fundamental para aplicações de visão computacional.

Sistemas inteligentes de vigilância eletrônica por vídeo, por exemplo, podem ser capazes de, automaticamente, detectar movimento [1], rastrear pessoas e veículos [2], monitorar tráfego e até mesmo monitorar atividades, alertando o operador do sistema caso ocorram eventos considerados não usuais ou suspeitos [3].

Todas essas funcionalidades requerem, de alguma maneira, que os objetos de interesse sejam destacados das imagens para que as suas características e os seus comportamentos possam ser analisados e assim a informação possa ser produzida.

Uma das formas mais comuns de se realizar essa tarefa, no caso de vídeos gerados por câmeras estáticas, é por meio da modelagem do plano de fundo (*background*) das imagens.

Uma vez estabelecido o modelo do plano de fundo, pode-se obter os alvos móveis (primeiro plano ou *foreground*) pela subtração de cada quadro  $Q$  da sequência de vídeo pelo plano de fundo  $F$  modelado, utilizado como imagem de referência, e com a aplicação de um limiar  $L$  na forma de (1).

$$|Q(x, y, t) - F(x, y, t)| > L \quad (1)$$

Todavia, a obtenção deste plano de fundo de referência nem sempre é uma tarefa trivial, principalmente quando se trata de imagens produzidas em ambiente externo não controlado e com trânsito constante.

Nestes casos, em razão do trânsito constante, é muito difícil obter uma imagem única em que todos os *pixels* façam parte do plano de fundo.

Além disso variações de iluminação resultantes da mudança da posição do sol ao longo do dia, da mudança de um céu claro para um encoberto, ou até mesmo variações bruscas em parte da imagem provocadas pela passagem de uma nuvem ou por iluminação artificial tornam o processo bastante complicado.

Ainda, em muitos casos, é preciso descontar do modelo de plano de fundo movimentos espúrios decorrentes, por exemplo, de vegetação balançando ao vento ou de corpos de água em movimento.

Os esforços deste trabalho concentraram-se no objetivo de aprimorar os aspectos de velocidade de processamento e de eficiência no armazenamento dos dados.

Tais escolhas refletem o propósito de utilizar este processo de segmentação de vídeo não como um fim em si mesmo, mas como um módulo dentro de um processo maior de análise de trajetórias em tempo real, baseada em dados adquiridos de câmeras de segurança estáticas em áreas externas, de forma constante e durante longos períodos de observação.

Para atingirmos o objetivo estabelecido neste artigo, analisamos diversas técnicas de modelagem e subtração de plano de fundo encontradas na literatura e selecionamos para implementação o modelo de *codebook* [4], considerado como sendo o mais viável para o propósito relatado acima.

São propostas algumas modificações na idéia original do modelo a fim de torná-lo mais rápido e com estrutura de armazenamento ainda mais compacta.

Nas seções seguintes deste artigo faremos um breve resumo das técnicas estudadas, da proposta de modificação do modelo de *codebook* e dos resultados alcançados.

## II. TRABALHOS RELACIONADOS

Conforme exposto em [5] e mais recentemente em [6], existe uma vasta literatura a respeito das técnicas de modelagem e subtração ou remoção de plano de fundo, com diversas abordagens, algumas das quais apresentamos a seguir, de forma não exaustiva e bastante sucinta.

### *Modelos Unimodais*

**Diferenciação temporal** [6], que podemos considerar como a técnica mais simples e rápida, utiliza o quadro anterior da sequência de vídeo como modelo de plano de fundo para o quadro atual.

$$F(x, y, t) = Q(x, y, t-1) \quad (2)$$

$$|Q(x, y, t) - Q(x, y, t-1)| > L \quad (3)$$

Esta técnica, apesar de bastante rápida, na maioria das vezes acaba não sendo muito útil pois não detecta *pixels* internos de objetos com cores uniformes e objetos que param de se movimentar. Também tem muita dificuldade para detectar objetos que se movem lentamente ou que se deslocam na direção da câmera.

**Filtragem pela média** [6] estabelece como plano de fundo a média aritmética de um conjunto de quadros que podem ser selecionados de várias formas tais como os primeiros  $n$  quadros de uma sequência ou a média móvel com os  $n$  quadros mais recentes (5), consecutivos ou não, que também pode ser calculada de forma recorrente pura ou, privilegiando-se valores mais recentes, de forma recorrente exponencial.

$$F(x, y, t) = \frac{1}{n} \sum_{i=0}^{n-1} Q(x, y, t-i) \quad (4)$$

$$\left| Q(x, y, t) - \frac{1}{n} \sum_{i=0}^{n-1} Q(x, y, t-i) \right| > L \quad (5)$$

**Filtragem pela mediana** [6] parte do princípio de que o *pixel* contém o valor do plano de fundo durante a maior parte do tempo de observação  $n$ .

$$F(x, y, t) = \text{med} \{ Q(x, y, t-i) \}_{i=0}^{n-1} \quad (6)$$

$$|Q(x, y, t) - \text{med} \{ Q(x, y, t-i) \}_{i=0}^{n-1}| > L \quad (7)$$

**Filtragem preditiva linear** com filtro de *Wiener* [7] para prever o valor atual de um *pixel* utilizando uma combinação linear de  $p$  valores anteriores. Para um dado *pixel*, seu valor previsto é dado por:

$$s_t = - \sum_{k=1}^p a_k s_{t-k} \quad (8)$$

E o erro esperado da previsão é dado por:

$$E[e_t^2] = E[s_t^2] + \sum_{k=1}^p a_k E[s_t s_{t-k}] \quad (9)$$

Os coeficientes  $a_k$  são calculados a partir dos valores de covariância de  $s_n$ . Se a diferença entre o valor real do novo *pixel* e a sua previsão for maior que  $4\sqrt{E[e_t^2]}$ , então o *pixel* é considerado *foreground*.

Os modelos apresentados até aqui são relativamente simples de implementar e rápidos. Aqueles que utilizam vários quadros para a composição do *background* apresentam resultado bem melhor que a diferenciação temporal na

deteção de *foreground*. Porém, pela mesma razão, possuem requisitos de memória relativamente maiores. No caso da filtragem pela média este problema pode ser minimizado pelo cálculo recorrente. A maior fragilidade das técnicas apresentadas até aqui, no entanto, é a incapacidade de modelar *pixels* com distribuições multimodais.

### Modelos Multimodais

**Modelagem não-paramétrica** [8] trata o *pixel* como uma variável aleatória com uma função densidade de probabilidade associada que é estimada diretamente a partir dos dados sem qualquer suposição sobre a sua distribuição, evitando-se a escolha de um modelo específico de distribuição e o cálculo de seus parâmetros. Assim, dada uma sequência de valores  $X_1, X_2, \dots, X_n$  (de dimensão  $r$ ) para um determinado *pixel*, o estimador da probabilidade  $p$  de que este *pixel* assumo o valor  $x$  é dado por:

$$\hat{p}_h(x) = \frac{1}{nh^r} \sum_{i=0}^n K\left(\frac{x - X_i}{h}\right), \quad x \in \mathbb{R}^r \quad (10)$$

Onde  $K$  é uma *função de kernel* e  $h$  é o tamanho da aresta de um hipercubo de dimensão  $r$  sobre o qual a densidade de probabilidade é estimada e cujo valor  $h$  afeta a suavidade da estimação. Neste caso multivariado é conveniente utilizar o produto de *kernels* na forma:

$$K(x) = \prod_{j=1}^r K(x_j) \quad (11)$$

Então (10) passa a ser escrita como:

$$\hat{p}_h(x) = \frac{1}{nh^r} \sum_{i=0}^n \left\{ \prod_{j=1}^r K\left(\frac{x_j - X_{ij}}{h}\right) \right\}, \quad x \in \mathbb{R}^r \quad (12)$$

Se utilizarmos um *kernel* gaussiano:

$$K(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \quad (13)$$

A forma final de (12) então fica:

$$\hat{p}_h(x) = \frac{1}{nh^r} \sum_{i=0}^n \left\{ \prod_{j=1}^r \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2} \frac{(x_j - X_{ij})^2}{h^2}} \right\}, \quad x \in \mathbb{R}^r \quad (14)$$

Caso a probabilidade estimada para  $x$  seja menor que determinado limiar, então o *pixel* é considerado *foreground*.

Como este modelo se adapta a uma distribuição qualquer, ele é capaz de representar *pixels* que apresentam comportamento multimodal. No entanto, a desvantagem deste modelo reside em sua complexidade, em sua necessidade de armazenar vários quadros e no estabelecimento de um valor adequado para  $h$ . Valores abaixo ou acima do adequado produzirão estimações respectivamente muito irregulares ou

super suavizadas prejudicando o resultado final.

**Modelo gaussiano múltiplo ou mistura de gaussianas** [9], é uma das soluções mais populares utilizada para modelagem de planos de fundo não-estáticos, multimodais. Neste modelo cada *pixel* do plano de fundo é descrito por um número pré-definido  $K$  ( $K > 2$ ) de distribuições gaussianas

$N_{kt} = N(\mu_{kt}, \sigma_{kt}^2)$ ,  $k=1, \dots, K$ , que são atualizadas ao longo do tempo  $t$ . A cada gaussiana é associado um valor de peso  $w_{kt}$  normalizado (soma = 1) que também evolui com o tempo. Assim, a probabilidade de se observar um valor de *pixel*  $x_t$  é dada por:

$$P(x_t) = \sum_{k=1}^K w_{kt} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(x_t - \mu_{kt})^2}{\sigma_{kt}^2}\right) \quad (15)$$

Para se classificar um novo *pixel*, ele é comparado a cada gaussiana e considera-se que há uma correspondência quando o valor do *pixel* encontra-se dentro da faixa de  $\pm 2,5\sigma_{kt}$  da  $\mu_{kt}$ . De acordo com [9], caso haja uma correspondência com a gaussiana  $l$ , atualizam-se os pesos conforme (16):

$$w_{k,t} = (1 - \alpha)w_{k,t-1} + \alpha(M_{k,t}) \quad (16)$$

Onde a constante  $\alpha$  é a taxa de aprendizagem e  $M_{k,t}$  é igual a 1 para a gaussiana correspondente e zero para as demais. Após isso os pesos são renormalizados e os parâmetros da gaussiana correspondente são atualizados da seguinte forma:

$$\mu_{lt} = (1 - \rho)\mu_{l(t-1)} + \rho x_t \quad (17)$$

$$\sigma_{lt}^2 = (1 - \rho)\sigma_{l(t-1)}^2 + \rho(x_t - \mu_{lt})^2 \quad (18)$$

Onde:

$$\rho = \alpha P(x_t | \mu_l, \sigma_l^2) \quad (19)$$

Caso nenhuma correspondência seja encontrada, a gaussiana com o menor peso é substituída por uma nova com média  $x_t$  e a ela é atribuída um valor alto de variância e um valor baixo para o peso, e esta passa a ser a gaussiana correspondente. Neste ponto já se sabe qual gaussiana tem maior probabilidade de conter o novo *pixel*. Resta agora saber se esta gaussiana representa o *background* ou o *foreground*. Para isso as gaussianas são ordenadas pela razão  $w_{kt}/\sigma_{kt}$  de forma que valores altos para esta razão indiquem maior chance da gaussiana representar o *background*. A classificação então dependerá de uma constante  $T$  que representa, de forma geral, a proporção das observações que podem ser atribuídas ao *background*. Assim, as distribuições gaussianas  $k = 1, \dots, B$  ordenadas pela razão  $w_{kt}/\sigma_{kt}$  são consideradas *background* para:

$$B = \operatorname{argmin}_b \left( \sum_{k=1}^b w_{kt} > T \right) \quad (20)$$

Este modelo, diferentemente do modelo não-paramétrico, não requer o armazenamento de vários quadros. Requer apenas uma estrutura que suporte  $K$  gaussianas por *pixel*. No entanto, apesar de possibilitar a modelagem de planos de fundo não-estáticos, ele também possui a desvantagem de ser computacionalmente intensivo.

**Modelo baseado em agrupamentos** [10], inspirado no modelo gaussiano e pretendendo um melhor desempenho computacional, cada *pixel* da imagem é modelado como um conjunto composto por um número pré-determinado de agrupamentos (*clusters*), cada um definido por um centróide e um valor de peso. O processo de formação do modelo e de classificação de um novo *pixel* é feito por comparação do valor do novo *pixel* com o valor do centróide de cada agrupamento do conjunto ao qual ele está relacionado, a partir do agrupamento de maior peso. Assume-se que há uma correspondência quando a distância de Manhattan ( $L_1$ ) entre o valor do novo *pixel* e o centróide do agrupamento for menor que determinado limiar. Quando nenhuma correspondência é encontrada, o agrupamento de menor peso é substituído por um novo com o valor do novo *pixel* como centróide e um pequeno valor de peso. Quando há uma correspondência, os valores do centróide correspondente e de todos os pesos são atualizados. Então os pesos são renormalizados (soma = 1) e os agrupamentos reordenados por estes valores. A classificação do novo *pixel* como *background* ou *foreground* depende de um critério que leva em conta a soma dos pesos a partir do maior até o agrupamento correspondente ao novo *pixel*, de forma similar ao modelo gaussiano múltiplo.

**Modelo baseado em codebook** [4] é um tipo de modelo baseado em agrupamentos, que neste caso recebem o nome de *codewords*. Uma diferença importante é que não há, para cada *pixel*, um número fixo de *codewords*. Esta característica apenas já conduz a uma estrutura mais enxuta pois muitos *pixels* só irão precisar de uma única *codeword* para sua representação. A estrutura de cada *codeword* é formada por um vetor RGB  $v_i = (\bar{R}, \bar{G}, \bar{B})$  e uma sêxtupla auxiliar  $aux_i = \langle \check{I}_i, \hat{I}_i, f_i, \lambda_i, p_i, q_i \rangle$  onde os componentes de  $v_i$  são as médias aritméticas dos componentes RGB,  $\check{I}_i$  e  $\hat{I}_i$  são o menor e o maior valor de brilho de todos os *pixels* atribuídos à *codeword* calculados como  $I = \sqrt{R^2 + G^2 + B^2}$ ,  $f_i$  é o número de vezes que a *codeword* foi acessada,  $\lambda_i$  é o maior intervalo de tempo em que ela não foi acessada e  $p_i$  e  $q_i$  os tempos do primeiro e do último acesso. Dado um novo *pixel*  $x_t = (R, G, B)$ , os critérios para determinar se ele pertence a uma *codeword*  $m$  são:

$$1. I_{low} < I < I_{hi} \quad \text{onde} \quad I_{low} = \alpha \hat{I}_m \quad \text{e} \quad I_{hi} = \min \left\{ \beta \hat{I}_m, \frac{\check{I}_m}{\alpha} \right\} \quad \text{e}$$

$$2. \text{distorção de cor } \delta(x_t, v_m) < \epsilon \quad \text{dada por:}$$

$$\delta = \sqrt{\|x_t\|^2 - \frac{\langle x_t, v_m \rangle^2}{\|v_m\|^2}} \quad (21)$$

Onde  $\|\cdot\|$  é o módulo e  $\langle \cdot, \cdot \rangle$  o produto escalar. A capacidade do método *codebook* de representar planos de fundo não-estáticos aliada à simplicidade dos cálculos e à estrutura enxuta justificaram a escolha do modelo.

### III. MODIFICAÇÃO PROPOSTA

Conforme já apresentado na seção anterior, o método de *codebook* utiliza coordenadas RGB para a representação dos *pixels*. Entretanto esse espaço de cor não é o mais indicado para a construção do modelo de plano de fundo. Para comprovar essa hipótese, analisamos uma sequência de vídeo obtida do *data set* PETS2001 ([visualsurveillance.org](http://visualsurveillance.org)) na qual acompanhamos a evolução dos valores de três *pixels* da imagem localizados em pontos com características diferentes e em três espaços de cores diferentes: RGB (*Red, Green, Blue*), HSV (*Hue, Saturation, Value*) e YCrCb (luminância e crominância).

A Fig. 1 mostra a localização dos três *pixels* analisados. Um está em uma área descoberta sobre a grama, o segundo está localizado em uma área de sombra dos carros estacionados e o terceiro está localizado próximo a uma extremidade da árvore, de forma que o seu valor alterne entre a cor da árvore e a cor da rua situada atrás da árvore à medida em que o vento balança os galhos.

O vídeo analisado possui 5336 quadros (3min e 33s) e é considerado de difícil segmentação, com múltiplos alvos, variações significativas de iluminação e uma árvore em primeiro plano balançando ao vento.

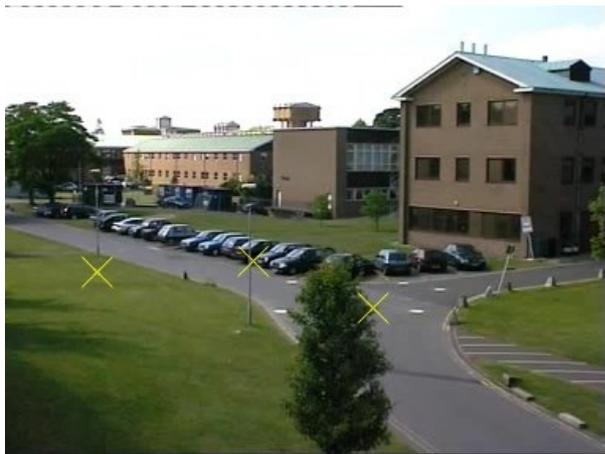


Fig. 1: Posições dos pixels analisados

Fig. 2, Fig. 3 e Fig. 4 contém os gráficos com os resultados das análises e estão organizados da seguinte forma. A coluna da esquerda contém os resultados do *pixel* localizado na área descoberta. Na coluna do meio encontram-se os resultados do *pixel* situado na sombra. E na coluna da direita estão os resultados do *pixel* posicionado na extremidade da árvore. Nas linhas, em sequência de cima para baixo, temos na Fig. 2 os valores de *R*, *G* e *B*; na Fig. 3 *H*, *S* e *V*; e na Fig. 4 *Y*, *Cr* e *Cb*. Nas abscissas dos gráficos temos a numeração dos quadros do vídeo, de 1 a 5336, e no eixo ordenado os valores dos componentes, de 0 a 255.

Pode-se observar, principalmente nos gráficos do *pixel* localizado na área descoberta, que ocorrem variações graduais significativas ao longo da sequência devido à passagem de nuvens que modificam a iluminação pelo sol.

Observam-se também variações bastante acentuadas e de curta duração resultantes da passagem dos alvos pelos *pixels*.

O modelo deve ser capaz de se adaptar às variações graduais do *background* e ao mesmo tempo identificar as variações provocadas pelos alvos. Para isso é necessário extrair informação de qualidade das imagens.

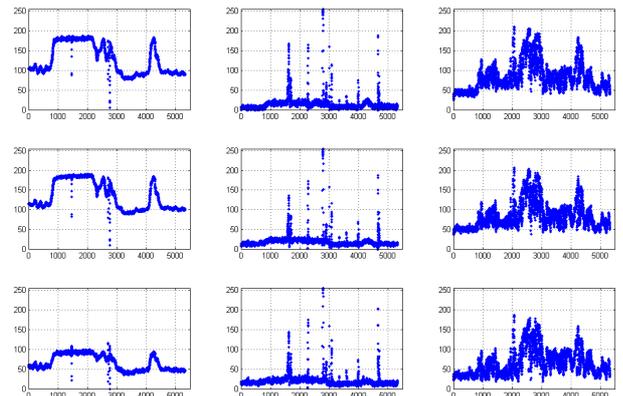


Fig. 2: Evolução dos pixels no espaço RGB

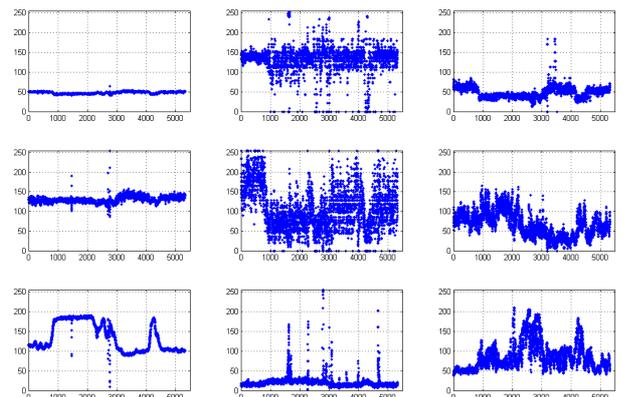


Fig. 3: Evolução dos pixels no espaço HSV

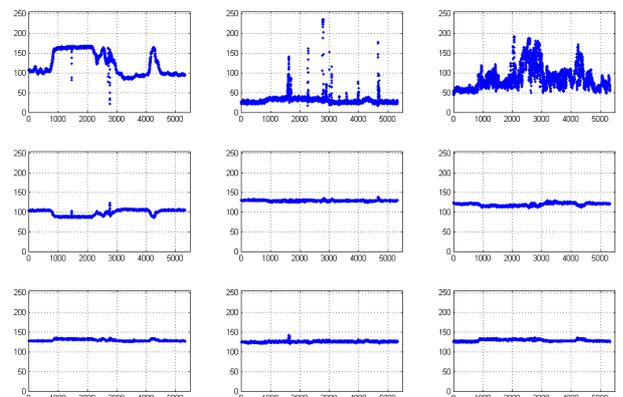


Fig. 4: Evolução dos pixels no espaço YCrCb

Um dos atributos de qualidade da informação é que ela seja distinta para evitar o desperdício de recursos na sua análise.

Na Fig. 2, o perfil gráfico semelhante para cada variável RGB indica que existe uma única variável da qual essas 3 variáveis dependem e, portanto, é considerado um espaço redundante.

A partir dessa constatação procuramos outras alternativas de espaço de cor que pudessem fornecer melhor informação da imagem. Em [11] há uma adaptação do modelo de *codebook* em que se utiliza o espaço de cor HSV. No entanto, a Fig. 3 mostra que a escolha deste espaço pode ser problemática. Vemos que nas áreas escuras (coluna do meio), o matiz (*H*) e a saturação (*S*) sofrem grande variação, inviabilizando a construção de um modelo nesses pontos. Uma saída interessante seria utilizar somente o componente do brilho (*V*), mas nesse caso a utilização da luminância *Y* do espaço de cor YCrCb seria mais interessante porque o *Y* é

uma composição dos valores RGB enquanto o  $V$  é igual ao componente RGB de maior valor. Além disso o YCrCb é um formato presente em muitos dispositivos de vídeo.

A conversão de RGB para YCrCb pode ser calculada da seguinte forma:

$$Y = K_{ry}R + K_{gy}G + K_{by}B \quad (22)$$

$$C_b = B - Y \quad (23)$$

$$C_r = R - Y \quad (24)$$

$$K_{ry} + K_{gy} + K_{by} = 1 \quad (25)$$

Onde  $K_{ry}=0,299$  e  $K_{by}=0,114$ , conforme a ITU-R Recommendation BT.601 ([www.itu.int](http://www.itu.int)).

Podemos observar pela Fig. 4 que no vídeo de teste a luminância praticamente representa toda a variação enquanto as crominâncias permanecem praticamente constantes. As crominâncias também não foram sensíveis aos alvos, portanto utilizaremos na proposta de modelo de *codebook* somente o componente  $Y$ .

Desta forma não faz mais sentido o cálculo de distorção de cor e também podemos substituir o armazenamento dos valores máximo e mínimo por um desvio padrão recorrente e eliminar outros parâmetros desnecessários.

Assim, em vez de termos as estruturas de *codewords* com 5 variáveis do tipo ponto flutuante  $R, G, B, \hat{I}, \hat{I}$  e 4 variáveis do tipo inteiro  $f, \lambda, p, q$ , teremos estruturas com apenas 2 variáveis do tipo ponto flutuante  $\bar{Y}, \sigma_Y$  e 2 variáveis do tipo inteiro  $f, q$ .

Os cálculos da luminância média e do desvio padrão recorrentes são feitos da seguinte forma, para obtermos um modelo adaptativo:

$$\bar{Y}_t = (1 - \alpha) \bar{Y}_{t-1} + \alpha Y \quad (26)$$

$$\sigma_{Y_t} = \sqrt{(1 - \alpha) \sigma_{Y_{t-1}}^2 + \alpha (Y - \bar{Y}_t)^2} \quad (27)$$

Onde  $\alpha$  é a taxa de aprendizado.

Finalmente, assumimos que um novo *pixel* com luminância  $Y$  pertence a uma determinada *codeword*  $c$  e, portanto, pertence ao *background* quando  $Y$  estiver dentro da faixa:

$$\bar{Y}_c \pm k \sigma_{Y_c} \quad (28)$$

Em análises qualitativas dos testes realizados utilizando diferentes sequências de vídeo, o método proposto mostrou, sistematicamente, desempenho ligeiramente superior na classificação correta dos *pixels*.

Na Fig. 5 apresentamos um quadro da sequência de vídeo para exemplificar a análise qualitativa. Com o método proposto os alvos estão com os contornos mais bem definidos e com os interiores mais bem preenchidos. Além disso observa-se uma imagem mais limpa, com menos falsos positivos que podem ser facilmente eliminados por um filtro morfológico.

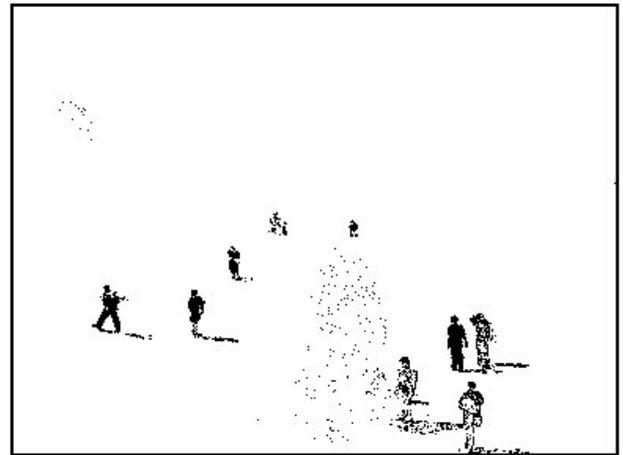


Fig. 5: De cima para baixo: imagem original, codebook original, codebook modificado

A estrutura mais simples reduziu ainda mais os requisitos de armazenamento e a simplificação do algoritmo proporcionou uma melhora no desempenho computacional.

Os algoritmos foram implementados com Microsoft Visual C++ Express Edition 2008, utilizando a biblioteca OpenCV. Em um sistema com processador AMD Turion 64 X2 Mobile Technology TL-58 com 1.90 Ghz e 2,00 GB de RAM, com sistema operacional Windows Vista Home Premium, o método *codebook* original operou a 14 quadros por segundo enquanto o método proposto atingiu 22 quadros por segundo.

#### IV. CONCLUSÕES

A modelagem de plano de fundo de sequências de vídeo é uma etapa importante em muitos processos de visão computacional e tem sido tema presente na literatura científica por muitos anos.

Neste trabalho foi realizada uma revisão dos métodos mais importantes para modelagem e subtração de planos de fundo expondo brevemente suas bases teóricas, características, seus pontos fortes e suas fragilidades.

Estabelecido um contexto de aplicação, com requisitos de obtenção de alvos móveis por câmera estática em área externa sujeita a variações de iluminação, planos de fundo multimodais, não-estáticos, e longos períodos de observação, optou-se pelo método de *codebook* como sendo o que melhor atende a essas necessidades.

Da análise da estrutura de dados apresentada pelo método de *codebook*, verificou-se que o espaço de cor RGB é intrinsecamente redundante e que a utilização do valor de luminância em substituição ao vetor RGB proporciona ganhos em relação aos requisitos de armazenamento e desempenho computacional.

#### REFERÊNCIAS

- [1] R. Cucchiara, C. Grana, M. Piccardi, A. Prati, "Detecting Moving Objects, Ghosts and Shadows in Video Streams", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 25, No. 10, October 2003, p. 1337-1342.
- [2] D. R. Magee, "Tracking Multiple Vehicles Using Foreground, Background and Motion Models", Image and Vision Computing, 22 (2004), p. 143-155.
- [3] C. Stauffer, W. E. L. Grimson, "Learning Patterns of Activity Using Real-Time Tracking", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 22, No. 8, 2000, p. 747-757.
- [4] K. Kim et al., "Real-time foreground-background segmentation using codebook model", Real-Time Imaging 11 (2005), p. 172-185.
- [5] M. Piccardi, "Background Subtraction Techniques: a Review", IEEE International Conference on Systems, Man and Cybernetics, 2004.
- [6] S. Y. Elhabian, K. M. El-Sayed, S. H. Ahmed, "Moving Object Detection in Spatial Domain Using Background Removal Techniques - State-of-Art", Recent Patents in Computer Science, Vol. 1, 2008, p. 32-54.
- [7] K. Toyama, et al., "Wallflower: principles and practice of background maintenance", International Conference on Computer Vision (ICCV'99), Vol. 1, 1999, p. 255-261.
- [8] A. Elgammal et al., "Background and foreground modeling using nonparametric kernel density estimation for visual surveillance", Proceedings of the IEEE, Vol. 90, No. 7, July 2002, p. 1151-1163.
- [9] C. Stauffer, W. E. L. Grimson, "Adaptive background mixture models for real-time tracking", Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 2, 1999, p. 246-252.
- [10] D. E. Butler, V. M. Bove Jr., and S. Sridharan, "Real-Time Adaptive Foreground/Background Segmentation", EURASIP Journal on Applied Signal Processing, vol. 2005, no. 14, p. 2292-2304, 2005.
- [11] A. Doshi, M. Trivedi, "Hybrid Cone-Cylinder Codebook Model for Foreground Detection with Shadow and Highlight Suppression", Proceedings of the IEEE International Conference on Video and Signal Based Surveillance (AVSS'06), 2006.