colloquium exactarum

DETECÇÃO DE MOVIMENTOS EM IMAGENS DE CÂMERAS DE VÍDEO

MOTION DETECTION IN VIDEO CAMERA IMAGES

Richard Hiroki Nishizima[1]; Mário Augusto Pazoti[1]; Francisco Assis da Silva[1]; Danillo Roberto Pereira[1]; Almir Olivette Artero[2]; Marco Antonio Piteri[2]

[1]Faculdade de Informática – FIPP, Universidade do Oeste Paulista – UNOESTE
e-mail: rhn_jp39@hotmail.com, {mario, chico, danilopereira}@unoeste.br
[2]Faculdade de Ciências e Tecnologia – FCT, Universidade Estadual Paulista – Unesp
e-mail: {almir, piteri}@fct.unesp.br

RESUMO – Neste trabalho é apresentado um sistema para classificação automática de movimentos em imagens gerados por câmeras de vídeo. Primeiramente é detectado o contorno da pessoa no vídeo fazendo o uso de métodos como o *saliency map*, filtros de binarização e detecção de bordas; em seguida, definem-se os *bounding box* dos contornos detectados, e a partir deles é feita a extração de características utilizando sobreposição de contornos e a divisão da imagem por regiões. Para a fase de treinamento e classificação dos contornos detectados, foi utilizado o classificador SVM. Os experimentos foram realizados com vídeos obtidos de uma base disponível na Web e vídeos elaborados pelos autores. Foram consideradas quatro classes de movimentos neste trabalho: Walk, Wave, Clap e Boxe. A taxa de acerto na detecção dos movimentos foi em média 85%.
Palavras-chave: Extração de Características; Detecção de Movimento; Saliency Map; SVM.

ABSTRACT – This paper presents an automatic classification of motion system in images generated by video camera. Initially is detected the contour of people in video using saliency map, binarization filters and edge detection filter methods; then, the bounding box of detected contours are defined and the features are extracted using an overlapping contours and division by regions. SVM classifier was applied for the training and classification steps. The experiments were performed with videos available in an action database on Web and videos generated by authors. Four motions classes were treated in this paper: Walk, Wave, Clap and Boxing. The hit rate obtained was on average 85%.
Keywords: Features Extraction; Action Recognition; Saliency Map; SVM.

Recebido em: 11/03/2015
Revisado em: 17/05/2015
Aprovado em: 13/06/2015

1 INTRODUÇÃO

Nos últimos anos, o número de sistemas de vídeo segurança tem aumentado consideravelmente em razão da preocupação das pessoas com a segurança de um local e também pelo baixo custo dos equipamentos de segurança. Diariamente é produzido um grande número de vídeos dessas câmeras de segurança ao redor do mundo, e na maioria das vezes, faz-se necessário ter uma pessoa para analisar os vídeos e identificar movimentos ou atitudes suspeitas nesses vídeos. Essa forma de análise pode trazer problemas em termos de tempo despendido, precisão e também de custo. Uma pessoa não pode prestar atenção em tudo que está ocorrendo o tempo todo, devido a distrações, fadiga ocasionada por longas jornadas de trabalhos, entre outros motivos. Portanto, é notável a necessidade de um sistema automatizado que faça a detecção e classificação do movimento em tempo real para que os responsáveis pela segurança de um local possam analisar tais movimentos e se for o caso, tomar as atitudes necessárias. (ROSHTKHARI; LEVINE, 2013).

Este trabalho vem contribuir com uma solução computacional que, a partir dos vídeos produzidos por câmeras de segurança, realize a detecção e classificação dos movimentos previamente definidos. O trabalho busca facilitar a análise e detecção de movimentos realizados pelos responsáveis pela segurança de um certo local, seja ele público ou privado.

O presente artigo está organizado da seguinte maneira. Na Seção 2 são apresentados os trabalhos relacionados à detecção de movimentos em vídeos e as metodologias utilizadas em cada um dos trabalhos. Na Seção 3 são apresentados os conceitos fundamentais que serviram de base para o desenvolvimento deste trabalho. Na Seção 4 é apresentada a metodologia utilizada no desenvolvimento da proposta. Na Seção 5 são apresentados os experimentos realizados e os resultados obtidos. Por fim, na Seção 6 são apresentadas as considerações finais e propostas de trabalhos futuros.

2 TRABALHOS RELACIONADOS

No trabalho de Maciel e Vieira (2012), sobre o reconhecimento de ações humanas em imagens de vídeo, são utilizados métodos para a extração e obtenção de descritores de uma imagem e também para a classificação da imagem. Para obter os descritores das imagens, o autor utiliza o método de extração de descritores baseado em histogramas de gradiente. Após essa etapa, eles são processados, gerando os vetores de tensores localmente agregados (VLAT – *Vector of Locally Aggregate Tensors*) para, posteriormente, realizar a classificação desses vetores por meio de ferramentas

apropriadas, como o SVM (*Support Vector Machine*). Segundo o autor, os resultados obtidos com a utilização desses métodos foram muito próximos aos resultados de outros trabalhos relacionados, como o de Klaser, Marszalek e Schmid (2008), Laptev et al. (2008) e o de Perez et al. (2012), alcançando uma taxa de reconhecimento de 89.9%.

No trabalho de Laptev et al. (2008), que trata o aprendizado das ações humanas a partir de imagens de vídeo, são utilizados métodos de reconhecimento de padrões para identificar as ações em imagens de vídeo. Para classificar o movimento, são calculados histogramas em volume espaço-temporal na vizinhança de pontos de interesse. Os volumes são divididos em cubos e para cada cubo calculam-se os histogramas de gradiente e de fluxo óptico. Posteriormente é realizada a normalização dos dados e, esses histogramas e fluxos normalizados são anexados em um descritor. A partir de um conjunto desses descritores é gerado um BOF (*Bag of Features*), que é utilizado posteriormente para a classificação do movimento. Para o reconhecimento e classificação do movimento, o autor utilizou o SVM com um *kernel* gaussiano multicanal, que é um método, que, a partir de um dado qualquer de entrada, faz a classificação desse dado com base em padrões de dados de classes utilizadas na fase de treinamento. Segundo o autor, utilizando o conjunto de dados de ações encontradas na base de vídeo *Action Database – Recognition of human actions* (LAPTEV; CAPUTO, 2005), que contém seis tipos de ações realizadas em diferentes cenários. A taxa de precisão para o reconhecimento dos movimentos, segundo o autor, foi de 91.8%.

3 CONCEITOS FUNDAMENTAIS

Nesta seção é apresentada a fundamentação teórica sobre os métodos utilizados para o desenvolvimento deste trabalho.

3.1 *Threshold* Otsu

O método proposto por Otsu (1979) baseia-se nas características de distribuição dos vários tons de cinza pertencentes à imagem. É um método não-paramétrico e não-supervisionado, em que se procura selecionar, de forma automática, os limites dos níveis de cinza em uma imagem, buscando melhor separar os elementos de interesse contidos na imagem.

Dada uma imagem, se os seus pixels forem representados por *N* níveis de cinza, então o número de pixel com nível *i* é descrito por d_i e o número total de pixels por $D = d_1 + d_2 + \cdots + d_N$.

De acordo com Otsu (1979) o histograma normalizado (Equação 1) é uma função de distribuição de probabilidade. A

Equação 2 representa a probabilidade de ocorrência, a Equação 3, o nível médio e a Equação 4, o desvio padrão por região ou classe segmentada, em que c representa a classe e T_c representa os limites dos níveis dessa classe.

$$p_i = \frac{d_i}{D};\ p_i \geq 0;\ \sum_{i=1}^{N} p_i = 1 \qquad (1)$$

$$q_c = \sum_{i=T_{c-1}+1}^{T_c} p_i \qquad (2)$$

$$\mu_c = \sum_{i=T_{c-1}+1}^{T_c} \frac{i*p_i}{q_c} \qquad (3)$$

$$\sigma_c^2 = \sum_{i=T_{c-1}+1}^{T_c} \frac{(i-\mu_c)^2*p_i}{q_c} \qquad (4)$$

O critério para a seleção dos limites dos níveis de cinza é baseado na minimização da expressão da variância, conforme Equação 5.

$$Min\ f(T_1, T_2, \ldots, T_{c-1}) = \sum_{i=1}^{c} q_i * \sigma_i^2 \qquad (5)$$

3.2 Filtros Passa-Baixa e Passa-Alta

Segundo Gonzalez e Woods (2000) filtragem espacial é utilizada para realizar o processamento de imagens utilizando máscaras espaciais, também denominadas de filtros espaciais.

Os filtros passa-baixa diminuem ou eliminam os componentes de alta frequência da imagem, por outro lado, os componentes com frequência baixa não são alterados, ou seja, o filtro permite a passagem apenas das frequências baixas do domínio. Os componentes de alta frequência no domínio representam as bordas e outros detalhamentos finos em uma imagem, portanto o efeito resultante dos filtros passa-baixa é o borramento da imagem.

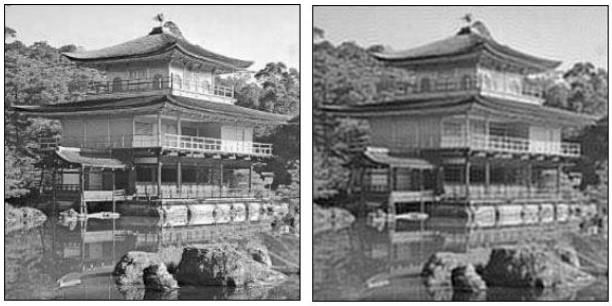

Figura 1. (a) Imagem Original; (b) resultados da aplicação de filtro passa-baixa Butterworth com frequência de corte de 64 pixels.
Fonte: (MARQUES FILHO; VIEIRA NETO, 1999).

Por outro lado, os filtros denominados passa-alta diminuem ou eliminam os componentes de baixa frequência na imagem. Como os componentes de baixa frequência são as frequências responsáveis pelas características que variam lentamente em uma imagem, como contraste total e intensidade média, o efeito resultante da aplicação do filtro passa alta em uma imagem é a redução dessas características da imagem, o que corresponde a um aparente realce das bordas e outros detalhes finos da imagem. Na Figura 2 é mostrado um exemplo de aplicação do filtro passa-alta em uma imagem a fim de realçar as bordas.

3.3 Integral da Imagem

De acordo com Nascimento et al. (2014), a integral da imagem representa uma matriz de área somada de uma imagem. Dada uma imagem *M*, a sua imagem integral

correspondente *I*(*x,y*) contém a soma das intensidades dos pixel de *M* acumulados, sendo que as somas são iniciadas em (*0,0*) e vão até o pixel (*x,y*), ou seja,

$$I(x,y) = \sum_{i=0}^{x} \sum_{j=0}^{y} M(i,j) \qquad (6)$$

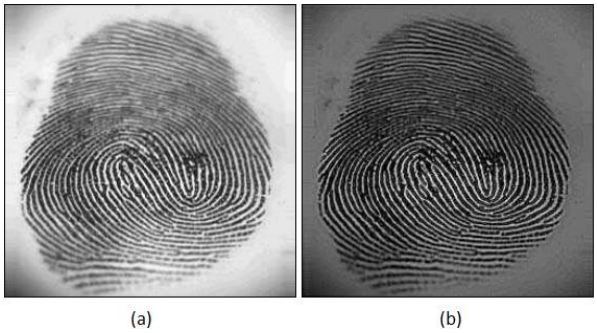

Figura 2. (a) Imagem Original; (b) imagem processada com filtro Butterworth passa-altas com ênfase em alta frequência.
Fonte: (MARQUES FILHO; VIEIRA NETO, 1999).

Em uma imagem integral é possível realizar o cálculo da área de qualquer região retangular da imagem de uma forma bem simples, como mostrado na Figura 3.

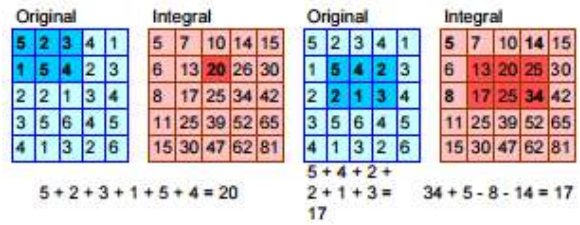

Figura 3. Exemplo da imagem integral e cálculo a partir da região em destaque.
Fonte: (NASCIMENTO et al., 2014).

3.4 Saliency Map

O S*aliency Map*, desenvolvido por Montabone e Soto (2008), aplica filtros e cálculos na imagem em várias escalas e resulta em um mapa de intensidade, destacando assim regiões de maior intensidade e interesse da imagem. A Figura 4 apresenta as etapas de execução do *Saliency Map*.

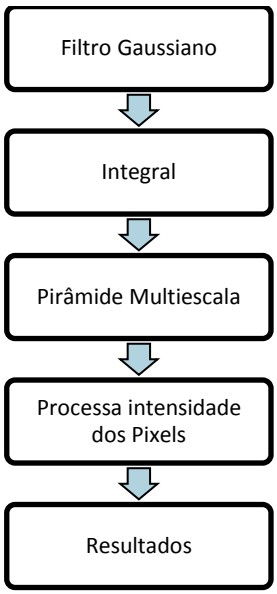

Figura 4. Etapas de processamento realizadas pelo *Saliency Map*.

Inicialmente, é aplicado um filtro gaussiano e é calculada a integral da imagem. Em seguida, é realizada a análise da pirâmide multiescala, em que são geradas várias subimagens em várias escalas, como mostrado na Figura 5.

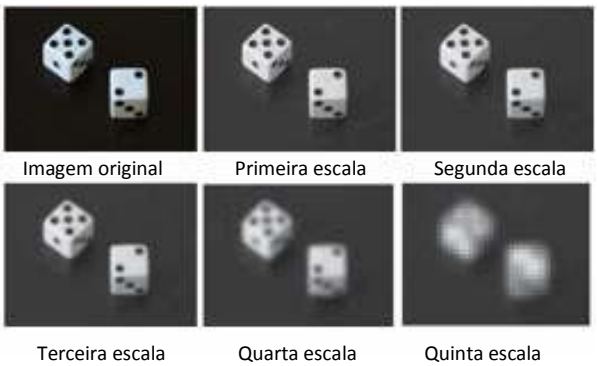

Figura 5. Imagem original e subimagens em várias escalas.
Fonte: (MONTABONE; SOTO, 2008).

Posteriormente é calculada a intensidade dos pixels em cada escala gerada, de acordo com as Equações 7 e 8

(MONTABONE; SOTO, 2008), mostradas a seguir:

$$Int_{On,\delta}(x,y) = Max\{center(x,y,\delta) - Surround(x,y,\delta,\sigma), 0\} \quad (7)$$

$$Int_{Off,\delta}(x,y) = Max\{Surround(x,y,\delta,\sigma) - center(x,y,\delta), 0\} \quad (8)$$

onde δ representa a escala e σ a vizinhança do pixel.

Após calcular Int_{On} e Int_{Off}, que representam respectivamente a intensidade do pixel atual e a intensidade da vizinhança ao redor do pixel, para cada escala da imagem, são calculadas as intensidades gerais do mapa, de acordo com as Equações 9 e 10 (MONTABONE; SOTO, 2008):

$$Int_{On} = \sum_{\delta} Int_{On} \quad (9)$$

$$Int_{Off} = \sum_{\delta} Int_{Off} \quad (10)$$

Por fim, são calculadas as intensidades, pixel a pixel, do mapa final por meio da Equação 11 (MONTABONE; SOTO, 2008):

$$Sal(x,y) = \frac{(Int_{On}(x,y) + Int_{Off}(x,y))}{MaxVal} \quad (11)$$

3.5 SVM (*Support Vector Machine*)

Segundo Oliveira Junior (2010), SVM (*Support Vector Machine*) é capaz de resolver problemas de classificação e regressão com o aprendizado adquirido na etapa de treinamento.

De acordo com Soares (2008), o funcionamento da SVM pode ser descrito da seguinte maneira: dadas *n* classes e o conjunto de pontos que pertencem a essas *n* classes, a SVM determina o hiperplano que separa os pontos de modo que o maior número de pontos de uma mesma classe fique de um lado, ao mesmo tempo em que, maximiza a distância de cada classe a esse hiperplano, como mostrado na Figura 6. A distância entre o hiperplano e uma classe é representada pela menor distância entre ambos e, é denominada margem de separação. O hiperplano é determinado por um subconjunto dos pontos das duas classes a serem classificadas, denominados de vetores de suporte.

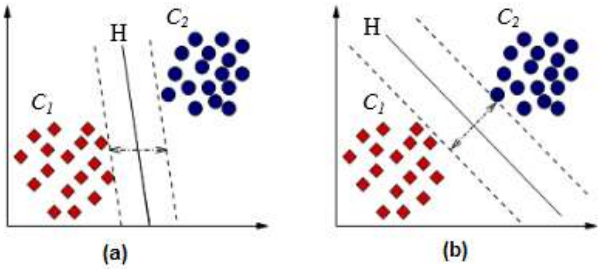

Figura 6. (a) Representação de hiperplano com margem de separação pequena e (b) Representação de hiperplano com margem de separação máxima.
Fonte: (SOARES, 2008).

O intuito é criar a partir desses métodos um classificador que funcione adequadamente para exemplos não conhecidos, ou seja, que não foram utilizados na fase de treinamento.

4 ALGORITMO PARA A DETECÇÃO DOS MOVIMENTOS

Foram aplicadas técnicas de visão computacional para a extração de características e classificação de movimentos

em vídeos gerados por câmeras com tempo de execução próximo do real. Na Figura 7, é mostrado o fluxograma que detalha as etapas do processo de desenvolvimento deste trabalho.

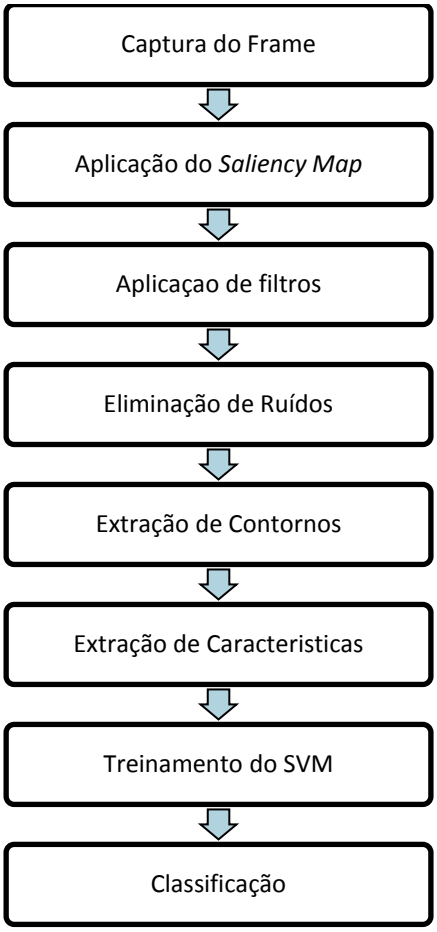

Figura 7. Representação das Etapas para o desenvolvimento do algoritmo.

Inicialmente é realizada a captura dos quadros do vídeo e, para cada quadro, aplicado a conversão para tons de cinza.

Após isso, cada quadro é submetido ao S*aliency Map*, conforme descrito na Subseção 3.4, o qual retorna um mapa de intensidade da imagem de entrada.

Na Figura 8 são mostrados os resultados gerados pelo *Saliency Map*.

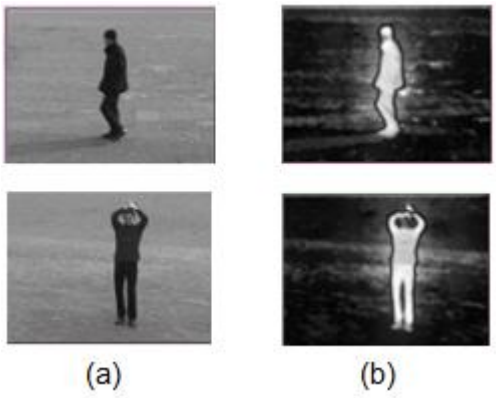

Figura 8. (a) Imagens de Entrada do *Saliency Map*. (b) Imagens de Saída, geradas pelo *Saliency Map*.

As imagens geradas pelo *Saliency Map* ainda não estão prontas para gerar as características necessárias para o treino do SVM, sendo necessário aplicar os seguintes filtros: *threshold* proposto por Otsu (1979) para binarizar a imagem; filtro gaussiano para suavizar a imagem; filtro de detecção de Bordas de Canny (1986) para, na sequência, usar a função *FindContours*(), da biblioteca OpenCV (2014) que utiliza o Algoritmo de detecção de borda *FollowContour* para realizar a detecção dos contornos na imagem. Na Figura 9 são mostrados alguns resultados gerados pela função *FindContours*().

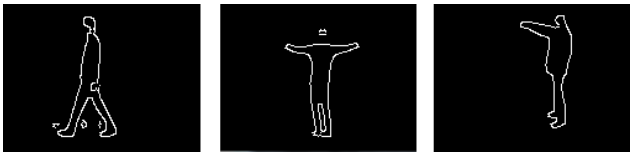

Figura 9. Contornos gerados a partir de três tipos de movimentos diferentes, denominado a seguir como *walk*, *wave* e *boxe*.

Os contornos extraídos são analisados de acordo com o seu tamanho, ou seja, são

selecionados contornos com um tamanho médio de uma pessoa e então é recortada da imagem apenas a área desses contornos (*bounding box*), como mostrada na Figura 10.

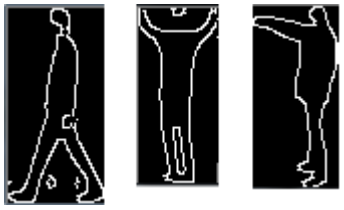

Figura 10. Áreas dos contornos extraídas da imagem.

Após a extração das áreas do contorno, é iniciada a etapa para a extração de características. Nessa etapa, primeiramente é armazenado um histórico de 10 contornos do movimento para que posteriormente, o movimento possa ser analisado de forma temporal e não somente o estado dele no quadro atual. Na Figura 11 é mostrada a sequência dos contornos referentes aos 10 últimos quadros, bem como a sobreposição de todos eles.

Para cada contorno armazenado é calculada a posição específica do seu centro de massa com o uso das Equações 12 e 13

$$CM_x = \frac{m_1x_1+m_2x_2+\cdots+m_nx_n}{m_1+m_2+\ldots+m_n} \quad (12)$$

$$CM_y = \frac{m_1y_1+m_2y_2+\cdots+m_ny_n}{m_1+m_2+\ldots+m_n} \quad (13)$$

onde x e y são as coordenadas da imagem e m é a massa, ou seja, a intensidade do pixel na coordenada (x, y).

Com o centro de massa de cada contorno calculado, todos os contornos são sobrepostos em uma única imagem de dimensão 120x120 pixels a partir do ponto de centro de massa com o uso da Equação 14 (QIAN et al., 2010).

$$E(x, y, t) = \bigcup_{i=t_0}^{t} D(x, y, i) \quad (14)$$

sendo t o número de contornos, D a intensidade do pixel na posição (x,y) do contorno i e E o resultado da sobreposição. Com isso é formado um esboço da execução do movimento, que é normalizado para que a imagem sobreposta fique com valores somente entre 0 e 1, gerando uma imagem em que regiões com várias repetições no movimento tenham intensidade maior.



Figura 11. Exemplo de histórico do movimento armazenado e sobreposição dos contornos, onde o primeiro quadro é o contorno atual, os outros os nove contornos anteriores e o ultimo a sobreposição pelo ponto de centro de massa.

Na Figura 12 é apresentado o resultado obtido nesta etapa.



Figura 12. Resultado obtido após a sobreposição e a normalização dos contornos.

Uma vez obtida a imagem dos contornos sobrepostos e normalizados, é iniciada a etapa de retaliação da imagem, em que a imagem é retalhada em várias subimagens como mostrado na Figura 13.

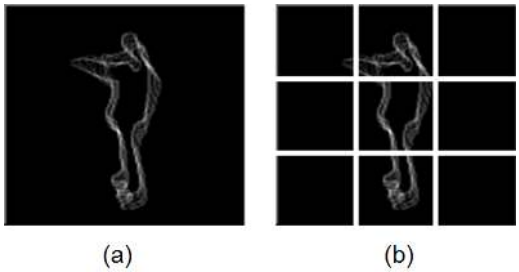

Figura 13. (a) Movimentos sobrepostos e normalizados. (b) Imagem retalhada.

Neste trabalho, foram geradas 121 subimagens, ou seja, a imagem sobreposta foi retalhada em 11x11 regiões. Cada subimagem contém um número de pixels do contorno original. Então, para cada subimagem é calculada a porcentagem de pixels do contorno pertencentes a ela, em relação à imagem original, como mostrado no exemplo da Figura 14. Essas porcentagens são as características a serem utilizadas no treinamento do classificador.

Antes de fazer a classificação, é necessário preparar o classificador para tal tarefa. Neste trabalho, o classificador utilizado foi o SVM.

Para o treinamento são utilizadas duas matrizes. A primeira matriz contém em cada linha os dados que serão utilizados para o treino e a segunda matriz contém em cada linha o rótulo da classe que será treinada, referente aos dados da mesma linha na primeira matriz. Para os parâmetros da SVM, foi utilizado um *kernel* linear, que separa as classes através de um hiperplano linear e o tipo da SVM foi o C_SVC (*C-Support Vector Classification*), que pode ser utilizado para a classificação de *N* classes, sendo *n*>2.

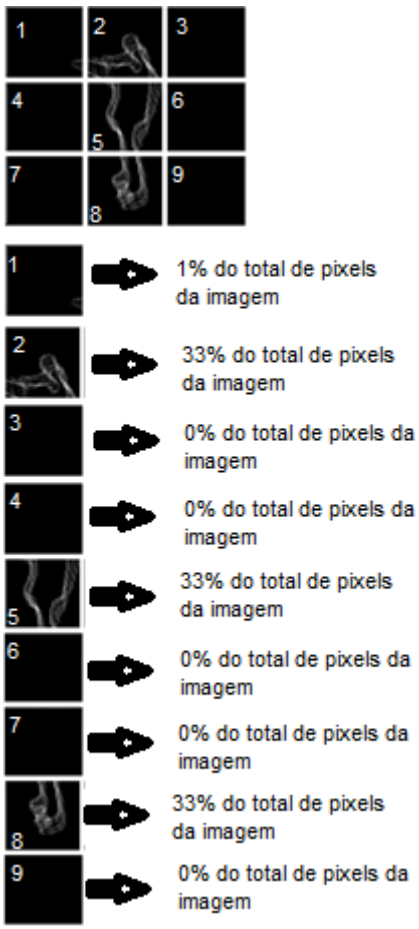

Figura 14. Exemplo das porcentagens calculadas por cada região.

Para o treinamento do SVM foram utilizadas as porcentagens das subimagens, geradas na Subseção 3.4, dispostas conforme

a Tabela 1, sendo que *N* é a quantidade de frames utilizados no treino, e *C* é a classe no qual os dados da linha pertencem.

Para a classificação é necessária somente a matriz de dados, os quais deverão estar dispostos da mesma forma como foi mostrado na Tabela 1.

Tabela 1. Layout dos dados de entrada para treino do SVM. (a) Matriz de Dados; (b) Matriz de Rótulos.

(a)						(b)
Frame 1	% subimagem 1	% subimagem 2	% subimagem 3	...	% subimagem 121	C
Frame ...	...	...	...	...	...	C
Frame N	% subimagem 1	% subimagem 2	% subimagem 3	...	% subimagem 121	C

5 RESULTADOS

Nesta seção são apresentados os resultados obtidos com os experimentos realizados com o algoritmo desenvolvido.

Os vídeos utilizados para o treinamento e em parte dos experimentos pertencem a uma base de dados denominada *Action Database – Recognition of human actions* (LAPTEV; CAPUTO, 2005), que contém vários tipos de ações realizadas em diferentes cenários. Uma segunda fase de experimentos foi realizada com um vídeo gerado pelos autores.

5.1 Experimento 1

Para a primeira fase de experimentos, foram utilizados os vídeos da base de dados (LAPTEV; CAPUTO, 2005), em que foram selecionados quatro tipos de movimentos: andando (classe walk), como mostrado na Figura 15; movimentando os braços (classe wave), como mostrado na Figura 16; batendo palmas (classe clap), como mostrado na Figura 17 e socos (classe boxe), como mostrado na Figura 18. Após realizar os experimentos com os vídeos da base de dados, os resultados obtidos foram organizados separadamente por classe.

Na Tabela 2 são apresentados os resultados do classificador quando realizados os testes com a classe walk (Figura 15). Pode-se observar que o classificador, mesmo treinado apenas com um vídeo (Figura 15(a)) identificou corretamente o movimento durante o teste dos demais vídeos desta classe contidos na base (Figura 15(b)(c)(d)).



Figura 15. (a) vídeo utilizado para treino da classe walk; (b) teste 1; (c) teste 2; (d) teste 3.

Tabela 2. Resultados obtidos para a Classe Walk.

	Teste 1	Teste 2	Teste 3
Walk	100%	100%	100%
Wave	0%	0%	0%
Clap	0%	0%	0%
Boxe	0%	0%	0%

Na Tabela 3 são apresentados os resultados do classificador quando realizados os testes com a classe wave (Figura 16). Pode-se observar que o classificador, treinado apenas com um vídeo (Figura 16(a)), identificou boa parte das cenas corretamente, mas houve confusão desse movimento com o movimento pertencente à classe clap, durante o teste dos demais vídeos desta classe contidos na base (Figura 16(b)(c)(d)).

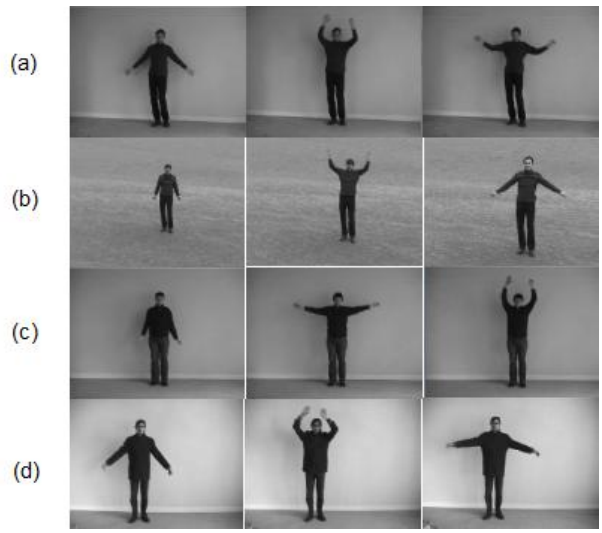

Figura 16: (a) vídeo utilizado para treino da classe wave. (b) teste 1. (c) teste 2. (d) teste 3.

Tabela 3. Resultados obtidos para a Classe Wave

	Teste 1	Teste 2	Teste 3
Walk	0%	0%	0%
Wave	95%	76%	100%
Clap	5%	24%	0%
Boxe	0%	0%	0%

Na Tabela 4 são apresentados os resultados do classificador quando realizados os testes com a classe clap (Figura 17). Pode-se notar que o classificador, treinado apenas com um vídeo (Figura 17(a)), não teve tanto êxito na classificação do movimento, confundindo nos testes 2 e 3 boa parte das cenas com a classe wave (Figura 17(c)(d)).

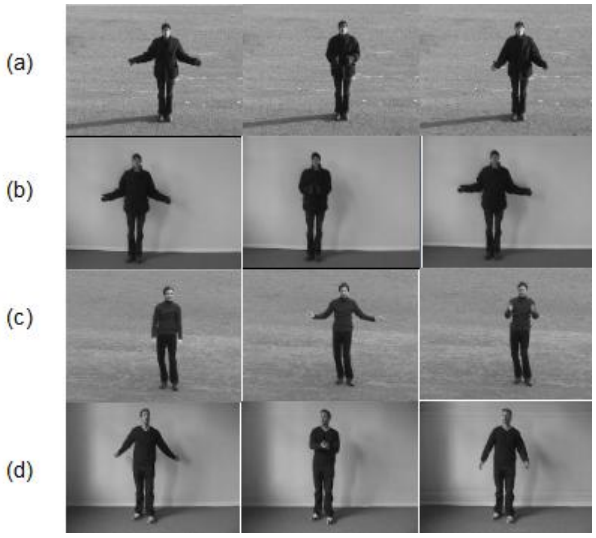

Figura 17. (a) vídeo utilizado para treino da classe clap. (b) teste 1. (c) teste 2. (d) teste 3.



Figura 18: (a) vídeo utilizado para treino da classe Boxe. (b) teste 1. (c) teste 2. (d) teste 3.

Tabela 4. Resultados para a Classe Clap.

	Teste 1	Teste 2	Teste 3
Walk	0%	0%	0%
Wave	0%	76%	70%
Clap	100%	24%	30%
Boxe	0%	0%	0%

Tabela 5. Resultados para a Classe Boxe.

	Teste 1	Teste 2	Teste 3
Walk	0%	0%	0%
Wave	0%	8%	0%
Clap	0%	0%	0%
Boxe	100%	92%	100%

Na Tabela 5 são apresentados os resultados do classificador quando realizados os testes com a classe boxe (Figura 18). Observa-se que o classificador, treinado apenas com um vídeo (Figura 18(a)) identificou boa parte das cenas corretamente, mas houve confusão desse movimento com o movimento pertencente à classe wave, em uma pequena parcela das cenas durante o teste 2 apenas (Figura 18(c)).

Posteriormente, foi calculada a média geral entre os quatro conjuntos de testes e a partir dessa média geral por classe foi gerada a matriz de confusão mostrada na Tabela 6, a fim de facilitar a visualização dos resultados obtidos.

Tabela 6. Matriz de Confusão dos resultados obtidos com os experimentos realizados com os vídeos da base de dados

	Walk	Wave	Clap	Boxe
Walk	100%	0%	0%	0%
Wave	0%	90%	49%	3%
Clap	0%	10%	51%	0%
Boxe	0%	0%	0%	97%

Na Tabela 6 é possível observar que para as classes walk e boxe, a taxa de acertos foi alta, com 100% para a classe walk e 97% para a classe boxe em relação às classificações corretas dos movimentos. Já para as classes wave e clap, a taxa de acerto foi razoável, com 90% para a classe wave e 51% para a classe clap de movimentos identificados corretamente. Isso ocorreu devido aos contornos gerados pelos movimentos dessas duas classes serem bem semelhantes durante a realização do movimento. Isso pode ser notado nos dois quadros extraídos de vídeos de ambas as classes, nas quais os contornos são bastante semelhantes, conforme mostrado na Figura 19.

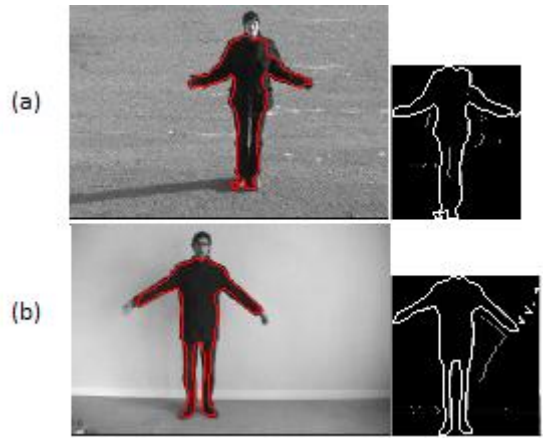

Figura 19. (a) Frame e contorno detectado para a classe de movimento Clap; (b) frame e contorno gerado para a classe de movimento Wave.

5.2 Experimento 2

Para a segunda fase de experimento, foi gerado um único vídeo contendo os movimentos das diferentes classes juntas, baseando-se nos movimentos e ambiente dos vídeos da base de dados utilizados no Experimento 1.

Os vídeos foram gravados com uma câmera de 6 megapixels e convertidos para uma resolução de 240x160 pixels.

Na Figura 20 são apresentados alguns quadros pertencentes às classes no vídeo que foi gerado para este segundo experimento.



Figura 20. Quadros das sequencias de movimentos; (a) Quadros da classe Walk; (b) Quadros da classe Wave; (c) Quadros da classe Boxe; (d) Quadros da classe Clap.

Com os experimentos realizados, foi calculada a quantidade de quadros pertencente a cada movimento e a quantidade de acertos e erros para cada classe. Desta forma, foi possível obter a porcentagem da média de acerto para cada movimento separadamente. Então, foi criada a matriz de confusão (Tabela 11) a partir das médias obtidas, para melhor visualização dos resultados.

Na Tabela 11 a taxa de acerto obtido para os movimentos foram de 78% para a classe walk, 62% para a classe wave, 71% para a classe clap e 57% para a classe boxe. Os resultados obtidos nesse segundo experimento decaíram em relação ao primeiro, porém, levando em consideração as diferenças de detalhes no ambiente, luminosidade e resolução do vídeo, os resultados atenderam as expectativas do que era esperado.

Tabela 11. Matriz de confusão dos resultados obtidos com os experimentos realizados com o vídeo gerado

	Walk	Wave	Clap	Boxe
Walk	78%	13%	20%	39%
Wave	14%	62%	9%	4%
Clap	3%	0%	71%	0%
Boxe	2%	25%	0%	57%

6 CONSIDERAÇÕES FINAIS

O setor de vídeo segurança nos últimos anos, vem crescendo notavelmente. E, devido a esse crescimento acelerado desse setor, o volume de vídeos gerados diariamente em todo o mundo é gigantesco. O que faz com que sistemas inteligentes para a detecção e classificação de movimentos sejam cada vez mais necessários, tornando assim a visão computacional um item indispensável para esses tipos de sistemas. Com um sistema para detectar e classificar movimentos automaticamente, o processo de análise dos vídeos gerados pelas câmeras não dependerá única e exclusivamente de uma pessoa.

Assim, o algoritmo desenvolvido neste trabalho utilizando técnicas de visão computacional como *saliency map*, extração de características por regiões e também classificadores como o SVM para a detecção e classificação do movimento, mostrou-se totalmente possível, e os resultados obtidos com as técnicas utilizadas neste trabalho comprovam essa possibilidade.

Em comparativo com os trabalhos de Maciel e Vieira (2012) e Laptev et al. (2008), que realizam a detecção e classificação de movimentos em vídeos, este trabalho realizou a classificação de 4 classes de movimentos em um tempo bem próximo ao tempo real e com uma taxa geral de acerto de 85%. Considerando os resultados das quatro classes utilizando os vídeos da base de dados, observa-se que eles ficam bem próximos aos resultados obtidos pelos trabalhos de Maciel e Vieira com 89% e Laptev et al. com 91%.

Após vários experimentos realizados, foi observado que, a extração do contorno da pessoa é uma etapa crucial para a extração de boas características para melhoria da classificação do movimento.

Com os resultados obtidos com o algoritmo desenvolvido, acredita-se que, após algumas melhorias, já seja possível a sua

utilização para o auxílio da segurança em ambientes fechados.

Para trabalhos futuros, foram listadas algumas mudanças e melhorias necessárias. Primeiramente, existe a necessidade de melhoria ou substituição das técnicas para detectar e extrair os contornos das pessoas. O contorno extraído precisa ser o mais próximo possível ao contorno original da pessoa, para que seja extraída do vídeo boas características, que, consequentemente melhorarão a classificação final do movimento.

Outros trabalhos importantes e necessários são a implementação de detecção de movimentos anormais para determinados ambientes, e também uma implementação para detecção e classificação dos movimentos de várias pessoas de uma só vez, ou seja, uma implementação para utilização em ambientes de grande fluxo.

REFERÊNCIAS

CANNY, J.A. **Computational approach to edge detection**. In: IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, n.6, p.679-698, 1986. http://dx.doi.org/10.1109/tpami.1986.4767851

GONZALEZ, R.C.; WOODS, R.E. **Digital image processing**. Edgard Blucher, 2000. 509p.

KLASER, A.; MARSZALEK, M.; SCHMID, C. A spatio-temporal descriptor based on 3d-gradients. In: BRITISH MACHINE VISION CONFERENCE. **Proceedings...** 2008. p.995–1004. http://dx.doi.org/10.5244/c.22.99

LAPTEV, I.; CAPUTO, B. **Recognition of human actions**: action database. 2005. Disponível em: <http://www.nada.kth.se/cvap/actions/>. Acesso em: mar 2015.

LAPTEV, I. et al. Learning realistic human actions from movies. In: IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION. **Proceedings...** Anchorage, AK.

MACIEL, L.M.S.; VIEIRA, M.B. **Reconhecimento de ações humanas utilizando histogramas de gradiente e vetores de tensores localmente agregados**. 2012.

MARQUES FILHO, O.; VIEIRANETO, H. **Processamento digital de imagens**. Rio de Janeiro: Brasport, 1999.

MONTABONE, S.; SOTO, A. Human detection using a mobile platform and novel features derived from a visual saliency mechanism. **Image and Vision Computing**, v.28, n.3, p.391-402, 2010. http://dx.doi.org/10.1016/j.imavis.2009.06.006

NASCIMENTO, A.C.P. et al. Uma metodologia para detecção de sinais de trânsito em vídeo. **Colloquium Exactarum**, v.6, n.3, p.26-44. 2015. http://dx.doi.org/10.5747/ce.2014.v06.n3.e086

OLIVEIRA JUNIOR, G.M. **Máquina de Vetores de Suporte:** estudo e análise de parâmetros para otimização de resultado. 2010. 41p. Monografia (Graduação em Ciência da Computação) – Centro de Informática, Universidade Federal de Pernambuco, Recife - PE.

OPENCV. **Open Source Computer Vision Library**. 2014. Disponível em: <http://opencv.org>. Acesso em: mar 2015.

OTSU, N. A threshold selection method from gray-level histograms. **Automatica**, v.11, n.285-296, p.23-27, 1975.

PEREZ, E.A. et al. Combining gradient histograms using orientation tensors for human action recognition. In: International Conference on Pattern Recognition. **Proceedings…** 2012. p. 3460-3463.

QIAN, H. et al. On Video-Based Human Action Classification By SVM Decision Tree. In: WORLD CONGRESS ON INTELLIGENT CONTROL AND AUTOMATION. **Proceedings…** 2010. p.385-390. http://dx.doi.org/10.1109/wcica.2010.5553829

ROSHTKHARI, M.J.; LEVINE, M.D. An on-line, real-time learning method for detecting anomalies in videos using spatio-temporal compositions. **Computer Vision and Image Understanding,** v.117, n.10, p.1436-1452, 2013. http://dx.doi.org/10.1016/j.cviu.2013.06.007

SOARES, H.B. **Análise e classificação de imagens de lesões da pele por atributos de cor, forma e textura utilizando máquina de vetor de suporte**. 2008. 154p. Tese (Doutorado) - Universidade Federal do Rio Grande do Norte, Natal – RN.