

RASTREAMENTO DE ALVOS EM VÍDEO VIA ALGORITMO SHUFFLED FROG LEAPING OPTIMIZATION

EDWARDS C. CASTRO, EVANDRO O. T. SALLES, PATRICK M. CIARELLI

*Programa de Pós-Graduação em Engenharia Elétrica, Universidade Federal do Espírito Santo
Av. Fernando Ferrari, 514, Prédio CT VI, sala 201, 29075-910, Vitória, ES, Brazil.*

E-mails: edwards_castro@yahoo.com.br, evandro@ele.ufes.br, pmc01@uol.com.br

Abstract— A new video target tracker is proposed for detection based on swarm intelligence strategy. To verify its performance, a trial design was planned with two other detectors and ten benchmark videos. Statistical tests of variance analysis were applied on the variables of Pascal. The results showed that the performance of the proposed tracker is as robust as the classic *Mean Shift*, but produces more homogeneous results.

Keywords— Tracking, Swarm Intelligence, Optimizations Algorithms, SFL, PSO, Mean Shift.

Resumo— É proposto um novo rastreador de alvos em vídeo por detecção baseado em estratégia de inteligência coletiva. Para verificar seu desempenho, um delineamento experimental foi planejado com outros dois rastreadores por detecção e dez vídeos *benchmark*. Testes estatísticos de análise da variância foram aplicados sobre a variável de desempenho de Pascal. Os resultados mostraram que o desempenho do rastreador proposto é tão robusto quanto ao clássico *Mean Shift*, porém, produz resultados mais homogêneos.

Palavras-chave— Tracking, Inteligência Coletiva, Algoritmos de Otimização, SFL, PSO, Mean Shift.

1 Introdução

Rastreamento em vídeo de alvos em movimento é uma área ativa de pesquisa em visão computacional e conta com uma ampla gama de aplicações em problemas do mundo real (Sardari e Moghaddan, 2017). Apesar dos avanços e das várias abordagens propostas, o rastreamento de alvos continua sendo uma tarefa bastante desafiadora, uma vez que inúmeros fatores, tais como: oclusão do objeto rastreado, deformação de objetos não rígidos, movimentos rápidos, variação na iluminação, baixo tempo de processamento e rotações afetam o desempenho dos rastreadores.

Os métodos de rastreamento podem ser categorizados, basicamente, em dois grupos (Sardari e Moghaddan, 2017): Tipo 1: Rastreamento por detecção que tentam identificar o alvo em cada *frame*, como exemplo, o *Mean Shift* (Comaniciu, Ramesh e Meer, 2003); e, Tipo 2: Rastreamento por métodos probabilísticos que tentam prever o estado não observado do alvo no *frame* corrente (a posição) baseado nos estados observados do alvo nos *frames* anteriores, tal como o Filtro de Kalman (Weng, Kuo e Tu, 2016) e o Filtro de Partículas (Gordon, Salmond e Smith, 1993).

As técnicas de otimização estocástica baseadas em populações têm sido empregadas juntamente com os algoritmos de ambos os grupos de rastreamento produzindo rastreadores com melhor desempenho. Por exemplo, elas são usadas no pré-enriquecimento das partículas, no algoritmo de Filtro de Partículas, empurrando-as para regiões do espaço de busca onde se encontram as modas da distribuição a posteriori. Dessa forma, diminui-se o efeito inevitável de degradação das partículas. Veja em Li et al. (2013), Walia e Kapoor (2014), Djogatovic, Stanojevic e Mladenovic

(2014), Gao et al. (2015) e Sardari e Moghaddan (2017).

Bae et al (2016) propuseram um rastreador do Tipo 1 empregando uma estratégia de busca por inteligência de partículas baseada no algoritmo de otimização por enxame de partículas, (*Particle Swarm Optimization*), PSO, de Kennedy e Eberhart (1995) em que um enxame de partículas é aleatoriamente distribuído num espaço de busca adaptativo buscando a partícula de melhor aptidão de acordo com a similaridade de alguns pontos específicos com o alvo modelo.

A proposta deste artigo é apresentar um rastreador de alvo em vídeo do Tipo 1. Em cada *frame*, o algoritmo *Shuffled Frog Leaping*, SFL, de Eusuff, Lansey e Pasha (2006) é executado. Inicialmente, um enxame de partículas é aleatoriamente distribuído num espaço de busca discreto e reduzido que se localiza a partir da posição do *pixel* central da caixa delimitadora que representa o alvo estimado no *frame* anterior. Após a convergência do algoritmo, a partícula de maior aptidão indica a presença do alvo no *frame* corrente. O objetivo é obter um rastreador robusto e generalista, no sentido de rastrear um alvo qualquer e o seguir, qualquer que seja o seu movimento, com um desempenho competitivo e superior em relação à robustez do rastreamento quando comparado aos rastreadores clássicos, tal como o *Mean Shift* de Comaniciu, Ramesh e Meer (2003).

Esse artigo está estruturado da seguinte forma: a seção 2 discute a metodologia adotada do algoritmo proposto e dos algoritmos competidores, as características estatísticas, o modelo de aparência dos alvos e as medidas de similaridade. O experimento e os resultados obtidos sobre um conjunto de vídeos *benchmark* serão apresentados na seção 3. Finalmente, a seção 4 aponta a conclusão deste experimento.

2 Métodos de Rastreamento

2.1 O Rastreamento de Alvo em Vídeos

O novo rastreador proposto, o rastreador de alvos baseado no algoritmo *shuffled frog leaping* (SFL), daqui em diante chamado de **TSFL** (T de *tracker*), é do primeiro tipo: rastreamento por detecção. Portanto, rastreadores do segundo tipo não serão abordados (uma ampla revisão sobre rastreadores do segundo tipo pode ser encontrada em Rao e Satyanarayana, 2013).

Resumidamente, os rastreadores por detecção, em geral, buscam pelo candidato a alvo no *frame* corrente do vídeo com a maior similaridade em relação ao alvo modelo. A medida de similaridade é calculada comparando as aparências dos alvos com base em características pré-determinadas. Para diferentes características, modelos de aparência e medidas de similaridade, o leitor pode consultar Rao e Satyanarayana (2013) e Sardari e Moghaddan (2017).

Três rastreadores do primeiro tipo são considerados aqui: o **TSFL**; o rastreador baseado no algoritmo PSO, o **TPSO**; e o rastreador baseado no algoritmo *Mean Shift*, o **TMS**. Todos compartilham o mesmo modelo de aparência e a mesma medida de similaridade.

A característica utilizada é o histograma padronizado dos canais de cor RGB (histograma padronizado é aquele cujas frequências somam 1) com h intervalos o qual é estimado pela regra de Struges (Martinez e Martinez, 2002, pag. 274), ou seja,

$$h = \frac{R_x}{\lceil 1 + 3.32 \log N \rceil}, \quad (1)$$

em que R_x é a amplitude amostral (é a diferença entre o máximo e o mínimo da amostra), N é o número total de *pixels* do *frame*, \log é o logaritmo na base 10 e $\lceil x \rceil$ é o menor número inteiro maior ou igual a x . O modelo de aparência dos alvos é a característica observada e a medida de similaridade é a distância quadrática de Bhattacharyya (Kailath, 1967). A escolha se justifica, pois, os histogramas de cor são características robustas na identificação dos alvos (Yang et al, 2011) e a distância Bhattacharyya é mais apropriada para o espaço métrico das probabilidades (Comaniciu, Ramesh e Meer, 2003).

2.2 O Algoritmo de Rastreamento Proposto

Nos últimos anos, algoritmos bioinspirados têm sido usados para resolver diversos tipos de problemas onde soluções convencionais são difíceis ou impossíveis de serem alcançadas. Há duas abordagens baseadas em populações: Os algoritmos evolutivos e os algoritmos de enxames. Os métodos baseados em inteligência de enxames têm chamado cada vez mais a atenção dos pesquisadores devido à robustez e fle-

xibilidade, principalmente, em ambientes dinâmicos (Serapião, 2009). O rastreador proposto, **TSFL**, estima o alvo no *frame* corrente espalhando um enxame de partículas sobre uma região da imagem. Essas partículas são candidatas a alvo e se movem nesta região da imagem em busca do local de maior similaridade com o alvo modelo.

Dado um vídeo, o **TSFL** busca identificar a posição do alvo no *frame* corrente reduzindo o espaço de busca. Esta região de interesse, ROI, reduzida da imagem é em torno da posição central da caixa delimitadora do alvo estimado no *frame* anterior. A ROI representa o espaço de busca no algoritmo de detecção em que é gerada aleatoriamente N partículas. Cada partícula representa uma janela $[x,y,s]$ em que (x,y) é um ponto da imagem correspondente ao centro da caixa delimitadora retangular que representa o alvo candidato e s é um fator de escala que controla o tamanho das caixas, ou seja, controla o comprimento e a altura das caixas delimitadoras (em número de *pixels*). As partículas, portanto, representam uma estimativa da presença do alvo no *frame* corrente.

Na seqüência, o algoritmo SFL é empregado para movimentar as partículas na direção da posição de menor distância Bhattacharyya em relação à aparência do alvo modelo.

O algoritmo SFL é uma meta heurística baseada em população do algoritmo memético. Meme é o nome dado ao mecanismo em que os indivíduos realizam troca de informações globais entre si, e representa um padrão de informação contagioso que infecta os indivíduos alterando os seus comportamentos e provocando a propagação desse padrão (Eusuff, Lansley e Pasha, 2006). O meme se assemelha ao gene no contexto de evolução genética, porém, ele constitui num mecanismo mais flexível, pois, pode ser transmitido rapidamente entre quaisquer dois indivíduos da população.

2.3 O Algoritmo SFL

No SFL, o espaço de busca é modelado como um pântano com pedras e sapos. As pedras estão localizadas em pontos discretos do espaço e representam as soluções. Os sapos são pontos no espaço e representam as possíveis soluções que estão sendo analisadas.

Inicialmente, os sapos são espalhados aleatoriamente no pântano e são agrupados em comunidades de sapos chamadas de memplexes. Os sapos realizam saltos mudando sua posição original com o objetivo de encontrar uma pedra. Essas mudanças são influenciadas pela posição do melhor sapo dentro dos memplexes e pela posição do melhor sapo do pântano. O melhor ou pior sapo é determinado pelo valor da função de aptidão que, neste trabalho, é dado pela distância quadrática de Bhattacharyya.

Seja x_g a posição do melhor sapo do pântano e sejam x_w a posição do pior sapo e x_b a posição do melhor sapo para cada memplex. Para alterar a posi-

ção do pior sapo impõe-se a ele um salto aleatório limitado l_w ($-l_{Max} \leq l_w \leq l_{Max}$), ou seja

$$x_{w,New} = x_w + l_w. \quad (2)$$

em que

$$l_w = U(x_b - x_w). \quad (3)$$

ou

$$l_w = U(x_g - x_w). \quad (4)$$

Nas equações 3 e 4, U é um número pseudo-aleatório gerado de uma distribuição Uniforme no intervalo unitário contínuo $[0,1]$.

O valor de l_w é calculado pela equação 3 se a aptidão de $x_{w,New}$ é menor que a aptidão de x_w caso contrário, l_w é calculada pela equação 4 (se a aptidão de $x_{w,New}$ se manter maior que a aptidão de x_w então $x_{w,New}$ será substituído por uma nova posição aleatória gerada dentro do espaço de busca).

Após a realização dos saltos em todos os memplexes, os sapos são novamente reunidos e redistribuídos aleatoriamente aos memplexes. Todo o processo se repete até que a condição de parada do algoritmo seja alcançada.

Neste trabalho foi utilizado 20 sapos e 5 memplexes (4 sapos por memplex). Os sapos são as partículas, ou seja, os sapos são as janelas $[x,y,s]$.

A condição de parada para todos os rastreadores é: rodar até 10 vezes o laço principal de otimização ou a distância quadrática Bhattacharyya ser menor que 0,005. Todos os números pseudo-aleatórios foram gerados pelo MatLab.

Para mais detalhes do algoritmo e o seu pseudocódigo, veja Eusuff, Lansley e Pasha (2006).

2.4 Os Algoritmos de Rastreamento Competidores

Para comparar o desempenho do rastreador proposto, dois outros rastreadores do Tipo 1 foram escolhidos: o **TPSO** e o **TMS**.

O **TPSO** é semelhante ao rastreador **TSFL** exceto no uso do algoritmo de busca empregado. Neste caso, o algoritmo usado é o PSO clássico (Kennedy e Eberhart, 1995). O **TPSO** é similar ao algoritmo proposto por Bae et al (2016).

O PSO imita o comportamento social observado por um enxame de pássaros (ou peixes). No PSO, o aprendizado das partículas se dá por duas formas: o cognitivo, baseado na experiência individual passada de cada partícula; e, o social, baseado no comportamento coletivo observado pelo enxame.

A principal diferença entre os algoritmos PSO e SFL é que o espaço de busca do PSO é contínuo e do SFL é discreto. Neste trabalho, uma busca local na vizinhança 8 de cada partícula foi introduzida para tornar possível o uso do PSO.

Neste trabalho, utilizou-se 25 partículas, o peso de inércia $w = 0,73$, e os pesos da parte cognitiva e da parte social foram, respectivamente, $\varphi_1 = 2,0$ e $\varphi_2 = 1,4$.

Veja o algoritmo PSO e outros algoritmos variantes do PSO em Bae et al (2016) e Serapião (2009).

O rastreador **TSM** é baseado no algoritmo *Mean Shift*. A sua escolha se deve ao fato de ser um algoritmo clássico já bastante testado e comparado com diversos outros rastreadores na literatura (Dou e Li, 2015).

O algoritmo *Mean Shift* é um método de rastreamento semi-automático e não se baseia em estratégia de enxames e nem em evolução estocástica.

O alvo modelo é representado pela sua função densidade da probabilidade (fdp) do espaço de cores q . Nos *frames* subsequentes, um alvo candidato é definido no local y_0 e é caracterizado pela fdp $p(y_0)$. Ambas as fdp são estimadas pelos dados. O coeficiente de Bhattacharyya, $\rho(y)$, é a função de similaridade utilizada e desempenha o papel de verossimilhança, portanto, o seu local de máximo na imagem indica a presença do alvo. Para encontrar o máximo dessa função é utilizado um procedimento de otimização baseado no gradiente. Portanto, para $\rho(y)$ se tornar uma função contínua em y , uma regularização é imposta mascarando os candidatos com um *Kernel* isotrópico, $K(\cdot)$ (convexo e monotonicamente decrescente), no domínio espacial. Para mais detalhes sobre as propriedades deste rastreador e o pseudocódigo detalhado, veja em Comaniciu, Ramesh e Meer (2003).

3 Experimentos e Resultados

Nessa seção, os resultados experimentais aplicados aos três rastreadores sobre dez vídeos *benchmark* obtidos de “*Visual Tracker Benchmark/Hanyang*” (Wu, Lim e Yang, 2013) são mostrados e discutidos. A Tabela 1 lista os vídeos e suas principais características: O número T de *frames* por vídeo e os principais desafios para o rastreamento, em que MB indica movimento borrado, FM indica movimento rápido do alvo, IPR indica rotação do alvo no plano da imagem, OPR indica rotação do alvo fora do plano da imagem, DEF indica deformação do objeto, BC indica confusão do alvo com o fundo da imagem, SV indica variação na escala e LR indica baixa resolução da imagem (Wu, Lim e Yang, 2013).

A Figura 1 mostra o primeiro *frame* de cada vídeo.

Tabela 1. Descrição dos vídeos *benchmarks*.

Vídeo	Nome	T	Desafio
1	<i>BlurBody</i>	334	SV, DEF, MB, FM, IPR
2	<i>BlurCar2</i>	585	SV, MB, FM
3	<i>BlurFace</i>	493	MB, FM, IPR
4	<i>Bolt2</i>	293	DEF, BC
5	<i>Boy</i>	602	SV, MB, FM, IPR, OPR
6	<i>Couple</i>	140	SV, DEF, FM, OPR, BC
7	<i>Dog</i>	127	SV, DEF, OPR
8	<i>MountainBike</i>	228	IPR, OPR, BC
9	<i>Surfer</i>	376	SV, FM, IPR, OPR, LR
10	<i>Twinnings</i>	472	SV, POR
Total	-	3650	-

Para medir a qualidade do desempenho dos rastreadores, as medidas *Recall* (R), *Precision* (Pr), F e Pascal (Pa) foram adotadas (Everingham et al., 2010). Todas mensuram o percentual de sobreposição das janelas que representam os alvos verdadeiros manualmente marcados (ξ) e os estimados pelos rastreadores (ε). Todas as medidas variam de 0%, para total falta de sobreposição, a 100%, para total sobreposição.



Figura 1. Amostra de imagens dos vídeos (os vídeos de 1 a 10 estão, respectivamente, da esquerda para direita, de cima para baixo).

Recall mede o quanto do ξ é coberto por ε e pode ser expresso por

$$R(\xi, \varepsilon) = \frac{|\xi \cap \varepsilon|}{|\varepsilon|} \times 100\%. \quad (5)$$

em que $|A|$ indica a cardinalidade do conjunto A .

De forma similar, *Precision* mede o quanto do ε é coberto por ξ e pode ser expresso por

$$Pr(\xi, \varepsilon) = \frac{|\xi \cap \varepsilon|}{|\xi|} \times 100\%. \quad (6)$$

Um bom rastreador possui ambos Pr e R próximos de 100%, portanto, a medida F , definida como a média harmônica entre Pr e R , mede a qualidade espacial da sobreposição das janelas e é calculada como:

$$F(\xi, \varepsilon) = \frac{2 Pr(\xi, \varepsilon) R(\xi, \varepsilon)}{Pr(\xi, \varepsilon) + R(\xi, \varepsilon)} \times 100\%. \quad (7)$$

A medida de Pascal determina a sobreposição das janelas da seguinte maneira:

$$Pa(\xi, \varepsilon) = \frac{|\xi \cap \varepsilon|}{|\xi \cup \varepsilon|} \times 100\%. \quad (8)$$

As medidas de Pascal e F são mais robustas que as medidas Pr e R uma vez que as medidas Pr e R podem assumir valores 100% em duas situações indesejadas: Quando um rastreador estima como alvo toda a imagem do *frame*, neste caso, $Pr = 100\%$, ou quando o rastreador estima como alvo um único *pixel* do alvo marcado, neste caso, $R = 100\%$. Em geral, se utiliza a medida de Pascal para considerar se um alvo foi detectado e isso será adotado neste trabalho.

O alvo é considerado detectado em um dado *frame* se a medida de Pascal superar um limiar predefinido, lm ($0\% \leq lm \leq 100\%$), em geral, $lm = 50\%$.

Todos os dez vídeos foram utilizados por cada um dos rastreadores para gerar os resultados.

Os resultados gerados neste experimento consistem no registro das seguintes variáveis: Tempo de Processamento (Tm) medidos em segundos, a medida de Pascal (Pa), *Precision*, (Pr), *Recall* (R) e a medida F (F).

O alvo modelo considerado em todos os casos é o objeto marcado no primeiro *frame* de cada vídeo.

Foi utilizado um processador Intel Pentium dual-core de 1.86 GHz e 2 GB DDR2. Os rastreadores **TSFL** e **TPSO** foram programados e executados no MATLAB R2014a; e, para o rastreador **TMS**, foi utilizado o *toolbox* elaborado por Dollar (2016).

Neste trabalho não será feita uma análise da variável Tm , pois, não é adequado comparar o desem-

penho entre os rastreadores uma vez que o **TMS** utiliza códigos otimizados e os outros não.

Como os rastreadores **TSFL** e **TPSO** são baseados em algoritmos aleatórios, cada vídeo foi executado cinco vezes. Os valores registrados para as variáveis são as medianas das replicações (a mediana foi escolhida nessa etapa do processo por ser mais robusta a *outliers* que a média, principalmente em pequenas amostras como neste caso com apenas cinco replicações).

As Tabelas 2, 3 e 4 apresentam os resultados do experimento realizado, respectivamente, para os rastreadores **TSFL**, **TPSO** e **TMS**.

Tabela 2. Medidas do desempenho do rastreador **TSFL**.

VÍDEO	$Pa^{(1)}$	Pr	R	F	$Pa^{(2)}$
1	55,2 [2]	76,7	65,5	70,0	66,5 [2]
2	51,4 [2]	68,4	68,1	68,2	57,8 [2]
3	69,2 [3]	82,0	80,3	80,9	90,9 [2]
4	43,5 [2]	57,2	50,2	53,4	54,3 [2]
5	32,0 [2]	43,7	49,4	45,9	18,4 [2]
6	41,4 [2]	62,2	47,7	53,2	45,0 [2]
7	31,8 [2]	33,9	79,9	46,0	15,8 [2,5]
8	47,7 [1]	73,4	52,6	60,7	54,0 [1,5]
9	28,1 [3]	57,5	30,8	38,9	20,2 [3]
10	40,8 [2]	47,1	67,7	53,6	33,7 [2]
Média	44,1	60,2	59,2	57,0	45,6
Mediana	42,5	59,8	59,1	53,5	49,5
Std (cv)	0,12 (28)	0,15 (26)	0,15 (27)	0,13 (22)	0,24 (53)
Rank médio	2,1	-	-	-	2,1

Nas Tabelas 2, 3, e 4, $Pa^{(1)}$ é o valor mediano da variável Pa e $Pa^{(2)}$ representa o percentual de *frames* detectados pela variável Pa . Os valores entre colchetes representam o *rank* das variáveis $Pa^{(1)}$ e $Pa^{(2)}$, e os valores entre parênteses representam o coeficiente de variação, *cv*, em valores percentuais (*Std* é o desvio-padrão da média).

Os valores observados nas Tabelas 2, 3 e 4 mostram, para todos os rastreadores, uma notável variação dos resultados: *cv* varia de 22% a 63% da escala da variável. Isso pode ser explicado pelo fato da amostra ser pequena (dez vídeos). Logo, é inadequado inferir qualquer informação dos dados sem o apoio dos testes estatísticos. Entretanto, de modo empírico, é possível observar um desempenho semelhante dos três rastreadores ao comparar as medianas (ou mesmo as médias).

O coeficiente de variação das variáveis de desempenho do **TMS** é cerca de 20% a 50% superior aos coeficientes de variação das respectivas variáveis dos outros rastreadores. Isso é uma evidência de menor homogeneidade dos resultados do **TMS** em relação aos demais rastreadores.

Tabela 3. Medidas do desempenho do rastreador **TPSO**.

VÍDEO	$Pa^{(1)}$	Pr	R	F	$Pa^{(2)}$
1	55,4 [1]	77,3	65,9	70,4	71,0 [1]
2	56,3 [1]	75,4	67,6	70,3	73,9 [1]
3	70,3 [2]	83,1	80,9	81,7	90,3 [3]
4	42,5 [3]	56,8	48,8	52,4	53,9 [3]
5	28,9 [3]	40,6	45,9	42,5	13,5 [3]
6	39,9 [3]	59,3	46,5	51,5	40,7 [3]
7	31,7 [3]	34,0	79,1	45,7	15,8 [2,5]
8	47,2 [2]	73,7	51,9	60,4	54,0 [1,5]
9	28,7 [2]	58,4	31,4	39,5	21,0 [2]
10	40,5 [1]	47,5	66,7	53,4	33,2 [3]
Média	44,1	60,6	58,5	56,8	46,7
Mediana	41,5	58,9	58,9	52,9	47,3
Std (cv)	0,13 (30)	0,17 (27)	0,16 (27)	0,14 (24)	0,26 (56)
Rank médio	2,1	-	-	-	2,3

Em relação às medidas de detecção, $Pa^{(2)}$ e F , todos os rastreadores são similares com um desempenho mediano (ou médio) menor para o **TMS**.

As Figuras 2 e 3 mostram as curvas de percentual de detecção com limiares de sobreposição variando de 0% a 100% (eixos x) para a medida de Pascal, $Pa^{(1)}$. Cada janela de gráficos corresponde a um vídeo e mostra as curvas dos três rastreadores considerados (as curvas em azul referem-se ao rastreador **TSFL**, verde ao **TPSO** e vermelho ao **TMS**).

Tabela 4. Medidas do desempenho do rastreador **TMS**.

VÍDEO	$Pa^{(1)}$	Pr	R	F	$Pa^{(2)}$
1	32,9 [3]	85,5	33,6	44,8	24,6 [3]
2	21,7 [3]	22,1	69,2	31,2	15,0 [3]
3	74,5 [1]	89,0	83,1	84,7	92,3 [1]
4	52,8 [1]	90,4	58,2	67,5	55,0 [1]
5	64,9 [1]	78,3	82,2	77,4	82,7 [1]
6	67,1 [1]	77,9	79,8	77,8	87,1 [1]
7	43,3 [1]	50,6	78,5	58,5	33,9 [1]
8	11,0 [3]	12,8	15,7	13,8	13,6 [3]
9	36,0 [1]	64,7	41,5	45,2	35,6 [1]
10	38,3 [3]	87,9	42,6	48,7	35,4 [1]
Média	44,2	65,9	58,5	55,0	47,5
Mediana	40,8	78,1	63,7	53,6	35,5
Std (cv)	0,21 (46)	0,28 (43)	0,24 (41)	0,23 (41)	0,30 (63)
Rank médio	1,8	-	-	-	1,6

É importante observar que tanto o **TSFL** quanto o **TPSO**, diante de uma deformação do objeto, o alvo não é perdido apesar de produzir baixo valor para medida de Pascal. Isso ocorre por causa do fator de escala, s , que controla o tamanho das janelas de detecção do objeto. Os diferentes fatores de escala as-

sociados a cada partícula aumentam (ou diminuem) a largura e a altura das janelas de forma proporcional à largura e à altura da janela referente ao alvo modelo. Logo, quando o alvo não rígido sofre uma deformação, a janela do alvo marcado é diferente da janela estimada produzindo um valor pequeno para a medida de Pascal. Isso ocorreu nos vídeos 6 e 7, principalmente no vídeo 7 onde as deformações do objeto são maiores.

Os gráficos das Figuras 2 e 3 reforçam a necessidade dos testes estatísticos.

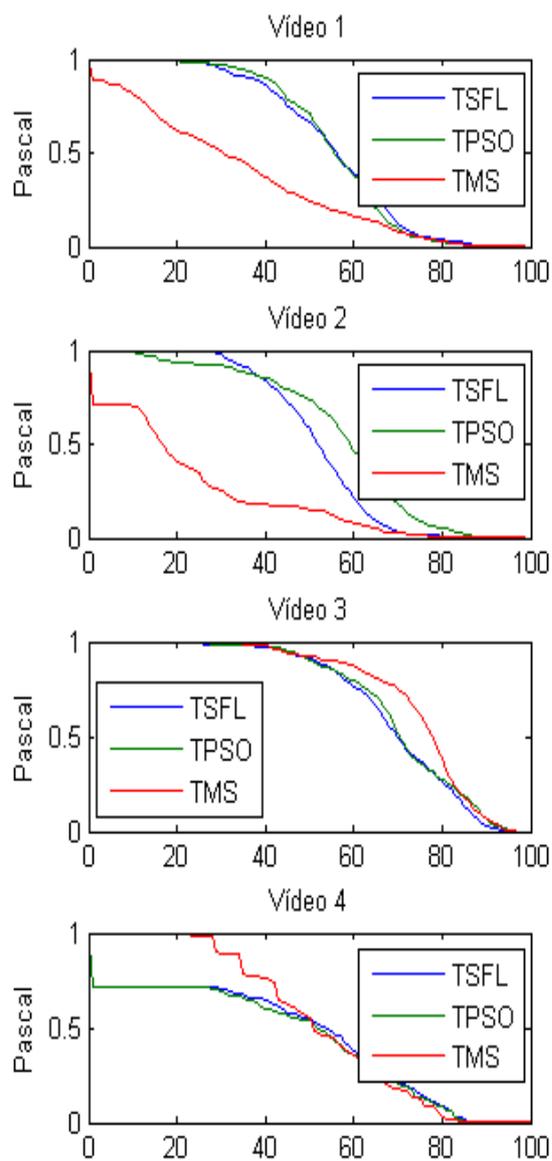


Figura 2. Curvas de detecção por Pascal.

Considerando, portanto, os rastreadores como tratamentos para os quais se pretende medir seus efeitos sobre as unidades experimentais e considerando os vídeos como unidades experimentais. Considerando, ainda, que os mesmos vídeos são utilizados para obter o desempenho de todos os rastreadores, então, pode-se considerar esse delineamento experimental

como um experimento em blocos ao acaso (Sheskin, 2011, p 1005-1118). Neste contexto, o procedimento estatístico correto de inferência é a Análise de Variância ANOVA *two-ways* não paramétrico uma vez que não há como avaliar ou garantir normalidade nem homocedasticidade dos dados.

O teste estatístico não paramétrico utilizado é o teste ANOVA de Friedman (1937) na variável ordinal *rank*. Caso a hipótese nula, H_0 , de igualdade das medianas seja rejeitada, seguem os testes *Post-hoc* de Nemenyi das diferenças (Nemenyi, 1963, citado por Demsar, 2006) comparando todos os rastreadores entre si, uma vez que não há um grupo controle.

O teste ANOVA foi aplicado nas variáveis $Pa^{(1)}$ e $Pa^{(2)}$. A escolha da medida de Pascal para variável de teste e a exclusão da medida F é porque as medidas F e Pascal são fortemente correlacionadas (em média, as correlações amostrais entre F e Pa é 0,9885 com desvio-padrão de 0,0034). Portanto, em termos estatísticos, a informação contida em uma não revela mais sobre o efeito do desempenho dos rastreadores que a outra.

Em ambas as variáveis $Pa^{(1)}$ e $Pa^{(2)}$, o teste não foi rejeitado ao nível de 5% de significância, ou seja, não é possível afirmar se um rastreador possui um desempenho superior aos outros.

4 Conclusão

Pelo experimento realizado e discutido nas seções anteriores, foi apresentado um rastreador de alvos em vídeo por detecção, o **TSFL**, baseado em estratégias de enxames de partículas via algoritmo SFL. O **TSFL** é capaz de rastrear um alvo qualquer em movimento com desempenho compatível com o clássico *Mean Shift*. Porém, evidências empíricas apontam que o **TSFL** produz resultados mais homogêneos (mais estáveis) para as diversas medidas de qualidade do desempenho quando comparadas com os resultados observados do **TMS**.

É necessário, entretanto, comparar as variâncias dos resultados para inferir essa evidência com confiança Estatística. Uma nova versão do **TSFL** está sendo trabalhada com um novo método para controlar a escala das janelas e combater o efeito da deformação de objetos não rígidos. Além disso, futuros trabalhos serão conduzidos com o objetivo de comparar outras estratégias de rastreamento baseadas em superfície contínua do espaço de busca gerada por *kernels* e algoritmos de inteligência coletiva.

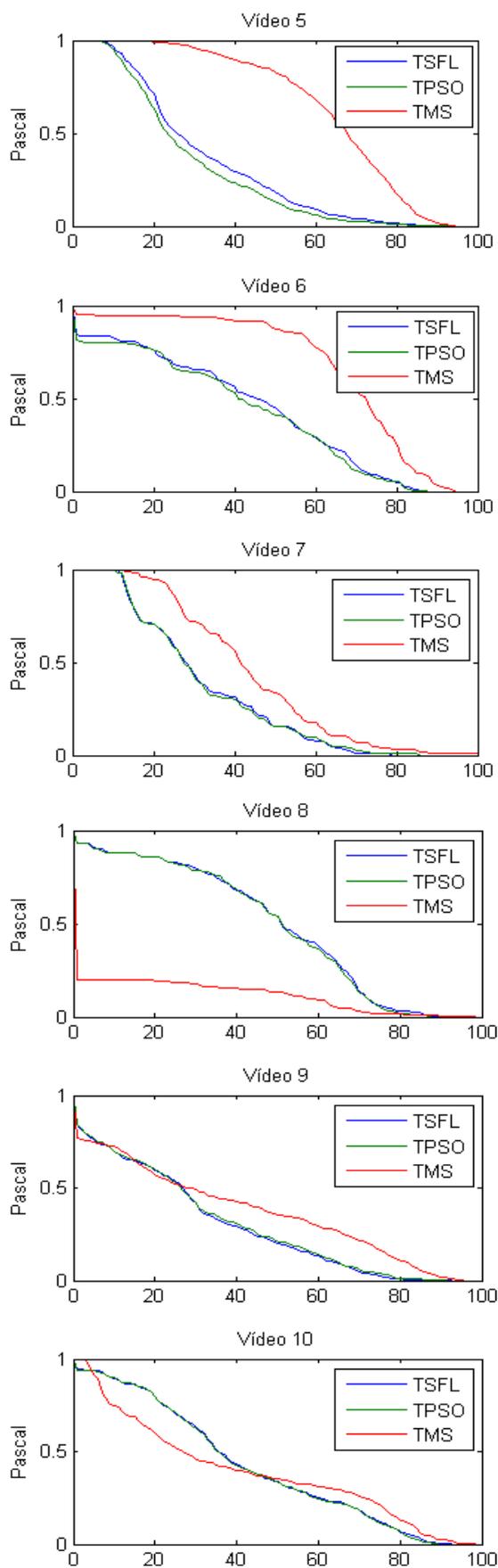


Figura 3. Curvas de detecção por Pascal.

Referências Bibliográficas

- Bae, C., Kang, K., Liu, G. and Chung, Y. Y. (2016). A novel real time video tracking framework using adaptive discrete swarm optimization. *Expert Systems With Applications* 64 385–399.
- Comaniciu, D., Ramesh, V. and Meer, P. (2003). Kernel-Based Object Tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 25, No. 5.
- Demsar, J. (2006). Statistical Comparisons of Classifiers over Multiple Data Sets. *Journal of Machine Learning Research*, 7 1–30.
- Djogatović, M. S., Stanojević, M. J. and Mladenović, N. (2014). A Variable Neighborhood Search Particle Filter for Bearings-Only Target Tracking. *Computers & Operations Research*, 52 192–202.
- Dollar, P. (2016). Piotr's Computer Vision Matlab Toolbox. <https://pdollar.github.io/toolbox/> (último acesso em 05/03/2018).
- Dou, J. and Li, J. (2015). Robust visual tracking based on joint multi-feature histogram by integrating particle filter and mean shift. *Optik* 126 1449–1456.
- Eusuff, M. M., Lansey, K. and Pasha, F. (2006). Shuffled Frog Leaping Algorithm: A Memetic Meta-Heuristic for Discrete Optimization. *Engineering Optimization*, vol. 38, no. 2, pp.129–154.
- Everingham, M., Van Gool, L., Williams, C. K., Winn, J. and Zisserman, A. (2010). The Pascal Visual Object Classes (voc) Challenge. *Int. J. Comput. Vision* 88, 303–338.
- Friedman, M. (1937). The Use of Ranks to Avoid the Assumption of Normality implicit in the Analysis of Variance. *Journal of the American Statistical Association*, 32, 675–701.
- Gao, M. L., Li, L. L., Sun, X. M., Yin, L. J. and Li, H. T. (2015). Firefly Algorithm (FA) Based Particle Filter Method for Visual Tracking. *Optik* 126 1705–1711.
- Gordon, N., Salmond, D. and Smith, A. F. M. (1993). Novel Approach to Non-Linear/Non-Gaussian Bayesian State Estimation. *IEEE Proceedings on Radar Signal Process*, 140, 107–113.
- Kailath, T. (1967). The Divergence and Bhattacharyya Distance Measures in Signal Selection," *IEEE Trans. Comm. Technology*, vol. 15, pp. 52–60.
- Kennedy, J. and Eberhart, R. (1995). Particle Swarm Optimization. In *Proceedings of IEEE International Conference on Neural Networks*, Perth, Australia, vol. 4, pp. 1942–1948.
- Li, M., Pang, B., Hi, Y. and Nian, F. (2013). Particle Filter Improved by Genetic Algorithm and Particle Swarm Optimization Algorithm. *Journal of Software*, Vol. 8, No. 3.

- Martinez, W. L. and Martinez, A. R. (2002). Computational Statistic Handbook with MatLab. Chapman & Hall/CRC, 584 pp.
- Rao, G. M. and Satyanarayana, C. (2013). Object Target Tracking Using Particle Filter: A Survey. *I.J. Image, Graphics and Signal Processing*, 6, 57-71.
- Sardari, F. and Moghaddan, M. E. (2017). A Hybrid Occlusion Free Object Tracking Method Using Particle Filter and Modified Galaxy Search Meta-Heuristic Algorithm. *Applied Soft Computing* 50 280–299.
- Serapião, A. B. S. (2009). Fundamentos de Otimização por Inteligência de Enxames: Uma Visão Geral. *Revista Controle & Automação*, Vol.20, 3 271-304.
- Sheskin, D. J. (2011). Handbook of Parametric and Nonparametric Statistical Procedures. Chapman & Hall /CRC Fifth Edition, 1886 pp.
- Walia, G. S. and Kapoor, R. (2014). Object Tracking: a survey. *Computers & Operations Research*, 52 192–202.
- Weng, S., Kuo, C. and Tu, S. (2006). Video object tracking using adaptive Kalman filter. *J. Visual Commun. Image Represent.* 17 (6) 1190–1208.
- Wu, Y., Lim, J. and Yang, M-H. (2013). Online Object Tracking: A Benchmark. *IEEE Conference on Computer Vision and Pattern Recognition*. 2411-2418.
http://cvlab.hanyang.ac.kr/tracker_benchmark/datasets.html (último acesso em 05/03/2018).
- Yang, H., Shao, L., Zheng, F., Wang, L. and Song, Z. (2011). Recent advances and Trend in Visual Tracking: A Review. *J. Neurocomputing*. 16 190–208.