

Utilização de Otimização por Enxame de Partículas e Algoritmos Genéticos em Rastreamento de Padrões

Yuri Marchetti Tavares

Departamento de Armas
Diretoria de Sistemas de Armas da Marinha
Rio de Janeiro, Brasil
marchetti@dsam.mar.mil.br

Nadia Nedjah¹ e Luiza de Macedo Mourelle²

¹Departamento Engenharia Eletrônica e Telecomunicações

²Departamento de Engenharia de Sistemas e Computação
Universidade do Estado do Rio de Janeiro

Rio de Janeiro, Brasil

¹nadia@eng.uerj.br e ²ldmm@eng.uerj.br

Abstract—O *template matching* é uma técnica importante para rastreamento de padrões em imagens. O objetivo consiste em encontrar um padrão, a partir de um modelo pré-estabelecido, em uma sequência de frames. Para avaliar o grau de similaridade entre duas imagens, o coeficiente de correlação de Pearson é amplamente utilizado. Esse coeficiente é calculado para cada pixel e computacionalmente muito custoso. Este artigo propõe a utilização e comparação de Algoritmos Genéticos e do algoritmo de Otimização por Enxame de Partículas visando diminuir o tempo de processamento para realizar o rastreamento. Para avaliar o desempenho das técnicas, comparou-se o resultado obtido dos algoritmos com aquele produzido pela busca exaustiva em cada pixel. A otimização por enxame de partículas é a mais eficiente, proporcionando uma aceleração de até 196x.

Keywords—*particle swarm optimization; template matching; algoritmos genéticos; correlação; tracking; imagem*

I. INTRODUÇÃO

A obtenção de informações por meio de imagens e vídeos tornou-se uma importante área de pesquisa sobretudo com o desenvolvimento e aprimoramento de sensores e equipamentos inteligentes capazes de capturar, armazenar, editar e transmitir imagens.

Na área de segurança e defesa, esse tipo de pesquisa é de grande importância no reconhecimento e acompanhamento de padrões (alvos) em sequências de imagens e pode proporcionar soluções para o desenvolvimento de sistemas de vigilância [1], monitoramento, controle de fogo [2], guiamento [3], navegação [4], biometria remota [5], armas guiadas [6], entre outros.

Padrão é um arranjo, ou coleção de objetos que são similares entre si, e é caracterizado pela arrumação de seus elementos

Uma das técnicas amplamente utilizadas para encontrar e rastrear padrões em imagens é chamada de *template matching* [9] [10]. Dentre as técnicas de *template matching* utilizadas para rastreamento de alvos, a correlação é um método bastante conhecido e muito utilizado. A tarefa basicamente se resume em achar ocorrências de uma imagem pequena, considerada como modelo (*template*), dentro de uma imagem maior. Esta

operação é muito custosa quando são considerados grandes modelos e extensos conjuntos de imagens [11].

A Fig. 1 mostra um exemplo de imagem com a correlação calculada em cada pixel. Observa-se que o gráfico da correlação possui um pico máximo bem visível, na área onde se encontra o centro do padrão desejado, conforme identificado pelo quadrado.

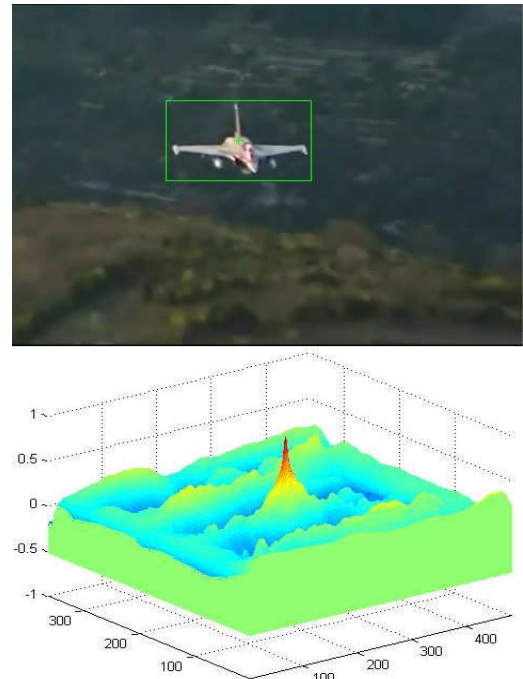


Fig. 1: Correlação ao longo do frame *aeronave₁*

O Algoritmo Genético ou GA (*Genetic Algorithm*) é um método amplamente utilizado para resolução de problemas científicos e de engenharia complexos (*NP-hard*), incluindo aqueles que recaem em tarefas de otimização, programação automática e aprendizado de máquina [17].

Recentemente, o algoritmo de otimização por enxame de partículas ou PSO (*Particle Swarm Optimization*) vem sendo utilizado para otimização de problemas que recaem em complexas funções não-lineares. Este método possui alta

convergência e grande simplicidade de implementação e estruturação [12].

Este artigo propõe a utilização do GA e PSO como forma de diminuir o tempo de processamento e o esforço computacional para encontrar o ponto máximo da correlação entre uma imagem e o padrão desejado. Ademais, é realizada uma comparação dos respectivos desempenhos na resolução do problema de interesse, permitindo a seleção do algoritmo mais adequado. Para tal, a Seção II apresenta alguns trabalhos relacionados; na Seções III é apresentado o conceito de correlação; nas Seções IV e V são descritos brevemente o funcionamento dos algoritmos genéticos e otimização por enxame de partículas, respectivamente; na Seção VI, são descritas a metodologia e forma como os algoritmos foram implementados; a Seção VII apresenta os resultados obtidos e a comparação dos desempenhos alcançados; e finalmente, a Seção VIII conclui o trabalho.

II. TRABALHOS RELACIONADOS

Um exemplo de utilização e relevância da correlação para rastreamento de padrões é verificada em [13], no qual foi realizada uma simulação de navegação aérea autônoma de um Veículo Aéreo Não Tripulado utilizando a correlação cruzada entre imagens de satélite e imagens aéreas obtidas através da câmera simulada para corrigir a posição estimada do veículo.

Em termos de aumento de desempenho, o algoritmo PSO foi utilizado para diminuir o tempo de computação em [14] e aumentar a acurácia do valor de *threshold* para um segmentador 2D de imagens em tons de cinza. Na área de rastreamento, o PSO foi utilizado em [21] para rastreamento de múltiplos objetos com oclusão. A aptidão das partículas foi baseada na comparação do histograma do objeto com o histograma da área localizada na posição da partícula. Os resultados demonstraram que o método permitiu uma busca global na imagem, rápida convergência e possibilidade de processamento em tempo real mas que a utilização do histograma é suscetível a erros quando a cena apresenta plano de fundo complexo e variações de iluminação.

Em [19], o algoritmo genético foi utilizado para reconhecimento de faces, ajustando o tamanho do modelo continuamente, de forma a auxiliar a tarefa de *template matching*. Já em [22], para evitar a busca exaustiva, o algoritmo genético foi utilizado para seleção das melhores características (*features*) em um rastreamento adaptativo por *mean-shift*. Os resultados apontaram que o método é eficiente e que melhora o desempenho.

O presente trabalho apresenta uma comparação entre PSO e GA como soluções para otimizar o tempo de processamento da tarefa de *template matching* e rastreamento de padrões em imagens. Diferentemente dos trabalhos citados, a avaliação da aptidão é calculada por meio da correlação cruzada normalizada.

III. CORRELAÇÃO

O *template matching* é utilizado em processamento de imagens para determinar a similaridade entre duas entidades (pixels, curvas ou formas) de mesmo tipo. O padrão (ou

protótipo de padrão) a ser reconhecido dentro de uma imagem é comparado com o modelo previamente armazenado. Após o cálculo da similaridade para todas as possibilidades, aquele que apresentar o maior valor, acima de um limiar pré-estabelecido, é considerado como o padrão desejado dentro da imagem.

Dentre as técnicas de *template matching* para rastreamento de alvos, a correlação é um método bastante conhecido e utilizado. O coeficiente de correlação de Pearson é utilizado como medida e pode ser entendido como um índice adimensional com valores situados ente -1 e 1 inclusive, que reflete a intensidade do grau de relacionamento entre duas variáveis. Coeficiente igual a 1 significa uma correlação perfeita positiva entre as duas variáveis. Coeficiente igual a -1 significa uma correlação negativa perfeita entre as duas variáveis, sendo inversamente proporcionais. Coeficiente igual a zero significa que as duas variáveis não dependem linearmente uma da outra. Para uma aplicação em imagens, o coeficiente de correlação de Pearson pode ser calculado conforme Equação 1 [15]:

$$corr = \frac{\sum_{i=1}^N (p_i - \bar{p})(a_i - \bar{a})}{\sqrt{\sum_{i=1}^N (p_i - \bar{p})^2} \sqrt{\sum_{i=1}^N (a_i - \bar{a})^2}} \quad (1)$$

onde p_i é a intensidade do pixel i na imagem padrão (*template*); \bar{p} é a média da intensidade dos pixels da imagem padrão; a_i é a intensidade do pixel i no pedaço de imagem A; \bar{a} é a média da intensidade dos pixels no pedaço de imagem A. O *template* e a imagem A devem possuir o mesmo tamanho.

A utilização ideal da correlação dada pela Equação 1 considera que a aparência do alvo permanece a mesma durante todo o vídeo [16]. Qualquer alteração de escala ou rotação do alvo pode diminuir a correlação entre as imagens e prejudicar o rastreamento do algoritmo. A própria iluminação, contraste ou ruído no ambiente (*clutter*) podem ocasionar erros. Uma possível solução para esse problema é atualizar o padrão a cada frame, levando para uma correlação adaptativa.

IV. ALGORITMOS GENÉTICOS

O Algoritmo Genético é um dos métodos mais consolidados da inteligência computacional [18]. É inspirado na teoria da evolução de espécies. Neste caso, a melhor solução para um determinado problema é encontrada a partir da combinação de possíveis soluções que são aprimoradas a cada iteração. A cada iteração ou *geração*, uma nova população de possíveis soluções ou *indivíduos* é criada a partir das informações genéticas dos melhores indivíduos da população da geração anterior. O algoritmo representa uma possível solução utilizando uma estrutura simples, chamada *cromossomo*, para a qual são aplicados operadores, geneticamente inspirados, de seleção, cruzamento e mutação, simulando o processo de evolução da solução. Cada cromossomo é formado de genes, sendo cada um representados por bits. A Fig. 2 apresenta o fluxograma de execução do GA.

No algoritmo genético básico, a população inicial de possíveis soluções é gerada aleatoriamente. Cada indivíduo é formado por cromossomo que são representados por cadeias

binárias. Esta população é avaliada por uma função de avaliação de aptidão (*fitness function*), permitindo atribuir um valor de aptidão proporcional à qualidade da solução em resolver o problema.

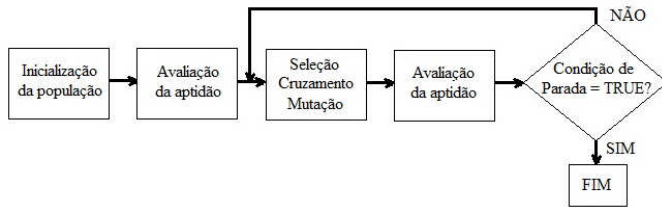


Fig. 2: Fluxograma de um algoritmo genético

Para criação da próxima geração, os indivíduos passam por um processo de seleção, que permite escolher aqueles indivíduos da população que são mais aptos a gerar os descendentes da geração seguinte. Quanto maior a aptidão do indivíduo, maior a chance dele se reproduzir e passar sua informação para a próxima geração. No algoritmo proposto em [18], o processo de seleção é do tipo roleta viciada, onde cada indivíduo é representado por uma fatia do espaço proporcional à sua aptidão. O processo de seleção é repetido, com reposição, até que se atinja o número desejado de indivíduos necessários para formar a nova população.

Para que não haja risco de se perder a melhor solução alcançada até o momento, algumas implementações de algoritmos genéticos utilizam o princípio de elitismo, que consiste na repetição incondicional do indivíduo melhor adaptado de uma geração na geração seguinte.

Após o processo de seleção, realiza-se o cruzamento dos indivíduos selecionados (genitores). Os genitores são separados dois a dois e, com probabilidade P_c , realiza-se o cruzamento (*crossover*). Se o cruzamento não ocorrer, os dois descendentes serão cópias exatas de seus genitores. Os algoritmos genéticos tradicionais geralmente usam o cruzamento simples ou de um ponto, onde os cromossomos de cada um dos genitores é cortado em um ponto aleatório e seus segmentos, a partir do ponto de corte, são permutados.

Após algumas gerações, a população terá indivíduos semelhantes, localizados em uma mesma região do espaço de busca, que pode não ser a região onde encontra-se o ótimo global, indicando uma convergência prematura. Para superar este problema, utiliza-se o operador mutação que fornece um comportamento exploratório, induzindo o algoritmo a amostrar novos pontos do espaço de busca. No operador proposto em [18], a mutação é caracterizada pela inversão, com uma certa probabilidade, de alguns bits do cromossomo.

O processo é repetido, com a avaliação e criação de novas populações, até que um critério de parada seja estabelecido. Esse critério pode ser um valor de aptidão considerado como aceitável ou um número máximo de gerações.

V. OTIMIZAÇÃO POR ENXAME DE PARTÍCULAS

O algoritmo PSO surgiu da observação do comportamento social de bandos de aves e cardumes de peixes [20]. As partículas, consideradas como possíveis soluções, se comportam como os pássaros à procura de alimento, utilizando

o aprendizado próprio (componente cognitiva) e o aprendizado do bando (componente social). O problema é expresso por meio da função objetivo. A qualidade da solução representada por uma partícula é o valor da função objetivo na posição desta partícula. O termo partícula é utilizado, em analogia a física, por possuir posição e vetor velocidade bem definidos mas não possuir massa nem volume. Já o termo enxame, representa um conjunto de possíveis soluções.

A cada ciclo iterativo, a posição de cada partícula é atualizada segundo Equação 2:

$$x_i^{(t+1)} = x_i^{(t)} + v_i^{(t+1)}, \quad (2)$$

onde $x_i^{(t)}$ é a posição da partícula i no tempo t e $x_i^{(t+1)}$ e $v_i^{(t+1)}$ são a posição e a velocidade da partícula i no tempo $t+1$, respectivamente.

O vetor velocidade é a soma de três componentes. As três componentes são inércia, memória e cooperação. A inércia mantém a partícula em uma direção idêntica à que ela vinha seguindo. A memória conduz a partícula na direção da melhor posição que a própria partícula encontrou até o momento. Já a cooperação conduz a partícula na direção da melhor posição até o momento, descoberta pelo enxame. A cada iteração, o vetor velocidade é atualizada conforme Equação (3):

$$v_i^{(t+1)} = wv_i^{(t)} + c_1r_1(pbest_i - x_i^{(t)}) + c_2r_2(gbest - x_i^{(t)}), \quad (3)$$

onde w é uma constante que representa a inércia da partícula, c_1 e c_2 são constantes que atribuem pesos das componentes cognitiva e social, respectivamente, r_1 e r_2 são números aleatórios entre 0 e 1, $pbest_i$ é a melhor posição encontrada pela partícula i e $gbest$ é a melhor posição encontrada pelo enxame. As posições das partículas se confinam no espaço de busca e a velocidade máxima é estabelecida para cada dimensão.

Os passos se repetem até que o critério de parada seja atingido. Esse critério pode ser o número máximo de iterações, um valor desejado da função objetivo, etc. A dinâmica dos passos utilizados no PSO pode ser vista na Fig. 3.

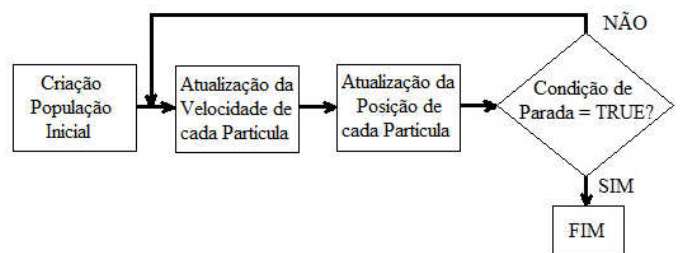


Fig. 3: Fluxograma do PSO

VI. METODOLOGIA

Nesta seção foram descritos a metodologia e os passos utilizados para a implementação dos algoritmos para a busca exaustiva, GA e PSO.

A. Busca Exaustiva (BE)

Para realizar o cálculo da correlação, pelo método da busca exaustiva, foi desenvolvido um script em MATLAB com os seguintes passos:

VII. RESULTADOS

Para realizar as simulações e comparações foi utilizado o software MATLAB 7.10.0 (R2010a) instalado em um computador ASUS, intel Core i7-2630 QM 2GHz, memória de 6Gb e com sistema Operacional Windows 7 Home Premium 64 bits. Para implementação do algoritmo genético, foi utilizado um pacote de ferramentas (*toolbox*) do MATLAB.

Primeiramente foi obtido um vídeo de aviões militares disponíveis no site http://www.youtube.com/watch?v=e3wi-i_hDVQ. Desse vídeo, foram retirados 2 frames, denominados a seguir de *aeronave₁* e *aeronave₂*. As Fig. 1 e Fig. 4 mostram os dois frames, com seus *templates* marcados e o gráfico da correlação ao longo da imagem. O frame *aeronave₁* possui 361×481 pixels e o frame *aeronave₂* possui 355×479 pixels.

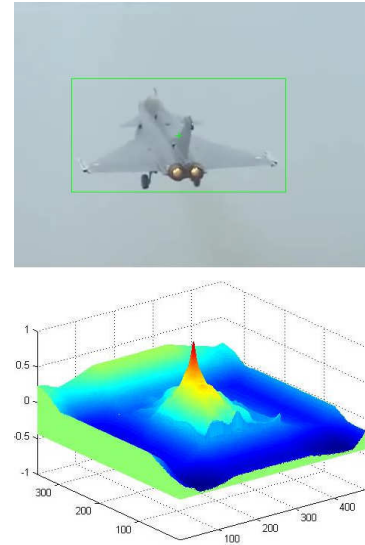


Fig. 4: Correlação ao longo do frame *aeronave₂*

Os algoritmos de busca exaustiva, GA e PSO foram utilizados para encontrar a localização do pixel referente ao centro do *template* na imagem original. Os resultados obtidos, juntamente com o tempo de processamento e valor máximo da correlação para as imagens *aeronave₁* e *aeronave₂* estão exibidos nas colunas 3 e 4 da Tabela 1, respectivamente. Nestes primeiros resultados, os parâmetros do algoritmo genético e do PSO foram configurados, empiricamente, da seguinte forma, nomeada de *config₁*:

- GA_{config_1} : população de 30 indivíduos; seleção por roleta e elitismo; taxa de cruzamento de 80%; taxa de mutação de 10%.
- PSO_{config_1} : enxame de 100 partículas, velocidade máxima permitida V_{max} de 10, $w = 1$, $c_1 = 1,5$ e $c_2 = 2$.

Como critério de parada, foram considerados, também empiricamente, um valor de 0,95 aceitável para o coeficiente de correlação ou um número máximo de iterações. Para o algoritmo genético foi estipulado um número máximo de 200 gerações e para o PSO, 50 iterações.

Durantes as simulações, observa-se que ambos algoritmos conseguem localizar o alvo desejado em um tempo menor do

- 1) transformação da imagem principal e do *template* para uma escala de cinza. Os cálculos são efetuados em termos da informação de intensidade dos pixels;
- 2) extração de um pedaço (recorte), de mesmo tamanho do *template*, para cada pixel da imagem principal. As bordas da imagem principal foram completadas com 0s (preto);
- 3) cálculo do coeficiente de correlação de Pearson tendo como centro cada um dos pixels da imagem principal; e
- 4) busca do ponto em que o valor da correlação foi mais elevado (centro do alvo);
- 5) localização do padrão no frame por meio de um quadrado verde.

B. Algoritmos Genéticos (GA)

Conforme proposto neste artigo, o algoritmo genético foi implementado em MATLAB na tentativa de avaliar o tempo de processamento, observando os seguintes passos:

- 1) transformação da imagem principal e do *template* para uma escala de cinza. Os cálculos são efetuados em termos da informação de intensidade dos pixels;
- 2) geração da população inicial, com cromossomos correspondendo a posições aleatórias na imagem principal;
- 3) para cada indivíduo, extração de recorte, de mesmo tamanho do *template* e cálculo do coeficiente de correlação de Pearson. O valor da correlação de cada indivíduo equivale a sua aptidão. As bordas da imagem principal foram completadas com 0s, representando a cor preta;
- 4) geração de nova população, utilizando a seleção por roleta e elitismo, o cruzamento a um ponto e a mutação. Como a correlação possui valores negativos, antes da seleção pela roleta os valores de aptidão foram normalizados;
- 5) repetição dos passos 3 e 4 até que um valor aceitável do coeficiente de correlação seja atingido ou um número máximo de iterações seja ultrapassado.

C. Otimização por Enxame de Partículas (PSO)

Conforme proposto neste artigo, foi implementado o algoritmo PSO no MATLAB na tentativa de diminuir o tempo de processamento, observando os seguintes passos:

- 1) transformação da imagem principal e do *template* para uma escala de cinza. Os cálculos são efetuados em termos da informação de intensidade dos pixels;
- 2) geração da população inicial das partículas, com posições e velocidades aleatórias;
- 3) para cada partícula, extração de um pedaço (recorte), de mesmo tamanho do *template* e cálculo do coeficiente de correlação de Pearson na posição da partícula. As bordas da imagem principal foram completadas com 0s (preto);
- 4) armazenar melhor valor encontrado pela partícula e melhor valor encontrado pelo enxame.
- 5) repetir os passos 3 e 4 até que um valor aceitável do coeficiente de correlação seja atingido ou que um número máximo de iterações seja ultrapassado.

que utilizando a busca exaustiva. Para a busca exaustiva o tempo de processamento está diretamente ligado ao tamanho da imagem e do *template*. Já para os algoritmos avaliados, esta afirmação não é sempre verdadeira pois o tempo de processamento também é influenciado pela forma como os parâmetros são escolhidos. No caso do PSO, observou-se claramente o comportamento em enxame das partículas conforme animação ilustrada na Fig. 5, onde as cruzes indicam as posições das partículas e o número indicado no canto superior esquerdo informa o número da iteração considerada. A animação referente a movimentação das partículas pode ser encontrado no site <http://youtu.be/CFEmWdjYQag>.



Fig. 5: Comportamento das partículas como um enxame

Para estender um pouco mais a tarefa e dar mais amplitude ao estudo, foi utilizado o benchmark *EgTest02*, do site <http://vision.cse.psu.edu/data/vividEval/datasets/datasets.html>, chamado neste trabalho de *carros*. Foi selecionado o primeiro frame da sequência, reduzindo-o a 320×240 pixels, conforme mostrado na Fig. 7. Observa-se um fator complicativo por meio do gráfico da correlação. Além do valor máximo do centro do alvo, e devido à presença de outros carros parecidos com o *template*, aparecem outros valores de correlação alta, que representam máximos locais. Desta vez, o GA e o PSO precisaram ser configurados de forma a evitar esses máximos locais. Os resultados dessa configuração, denominada *config2*, estão exibidos na última coluna da Tabela 1.

- $GA_{config2}$: população de 35 indivíduos; seleção por roleta e elitismo; taxa de cruzamento mantida em 80%; taxa de mutação de 15%; número máximo de gerações de 400.
- $PSO_{config2}$: enxame de 200 partículas; $w = 1,4$; $c_1 = 1,5$; $c_2 = 1,5$; velocidade máxima mantida em $V_{max} = 10$.

Tendo em vista o caráter estocástico do GA e PSO, é preciso avaliar o seu grau de convergência, considerando as configurações escolhidas. Para isso, realizou-se simulações do tipo Monte Carlo, no qual a busca foi repetida 100 vezes a fim de avaliar a probabilidade dos algoritmos convergirem. Os resultados dessa avaliação encontram-se nas Tabelas 2 e 3, respectivamente.

TABELA 1: Resultados obtidos pela Busca Exaustiva, GA e PSO

		<i>aeronave</i> ₁ (<i>config</i> ₁)	<i>aeronave</i> ₂ (<i>config</i> ₁)	<i>carros</i> (<i>config</i> ₁)	<i>carros</i> (<i>config</i> ₂)
Posição (linha,coluna)	BE	(145,214)	(178,220)	(164,194)	(164,194)
	GA	(145,213)	(178,221)	(164,194)	(164,194)
	PSO	(144,215)	(177,219)	(164,194)	(164,194)
Correlação	BE	0,9997	0,9996	0,9987	0,9987
	GA	0,9700	0,9766	0,9987	0,9987
	PSO	0,9699	0,9546	0,9987	0,9987
Tempo (seg)	BE	34,5	130	4,67	4,67
	GA	0,6626	0,6905	1,6969	2,1382
	PSO	0,3075	0,6610	0,2472	0,4705

TABELA 2: Simulação Monte Carlo para GA

Imagem	Tempo (s)		Gerações		Acertos (%)
	Média	Desvio	Média	Desvio	
<i>aeronave</i> ₁	0,8775	0,7384	58,10	48,8753	98
<i>aeronave</i> ₂	0,5165	0,4157	19,47	15,7207	100
<i>carros</i> (<i>config</i> ₁)	1,8273	0,6522	169,23	60,3503	29
<i>carros</i> (<i>config</i> ₂)	2,7883	1,9209	221,14	152,2567	71

TABELA 3: Simulação Monte Carlo para o PSO

Imagem	Tempo (s)		Iterações		Acertos (%)
	Média	Desvio	Média	Desvio	
<i>aeronave</i> ₁	0,2559	0,1103	11,86	5,1247	100
<i>aeronave</i> ₂	0,4672	0,1485	8,02	2,5623	100
<i>carros</i> (<i>config</i> ₁)	0,1948	0,0992	33,37	17,05	56
<i>carros</i> (<i>config</i> ₂)	0,3163	0,1750	27,02	14,98	81

Para facilitar a comparação dos tempos médios de processamento, foi construído o gráfico de barras da Fig. 6. Observa-se que o PSO é o mais rápido possuindo grau de acerto maior que o GA. Para a imagem *carros* na configuração 2, o ganho obtido com o GA é aproximadamente 2, com taxa de acerto de 71%.

Além disso, os resultados apontam que ambos algoritmos são sensíveis à escolha dos parâmetros. Em um problema real, existe a possibilidade que não haja convergência e que seja indicado um valor errôneo para a posição do alvo. Para mitigar os impactos, sugere-se repetir a busca algumas vezes até que o resultado desejado seja alcançado.

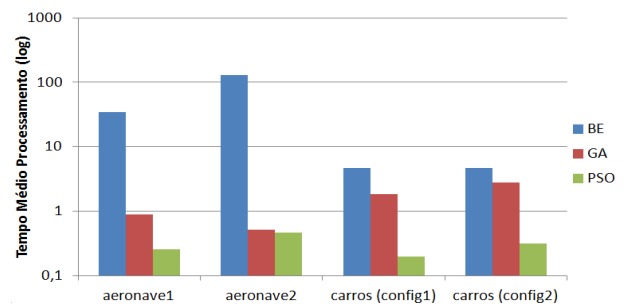


Fig. 6: Diagrama em barras do tempo médio de processamento (escala logarítmica)

Para ilustrar a repetição, ambos os algoritmos foram utilizados para realizar o *template matching* na sequência dos 120 primeiros frames no vídeo do benchmark *carros*. Para evitar a perda do alvo por alteração de escala ou rotação, o

template foi atualizado a cada frame. Tanto a busca exaustiva quanto o GA e o PSO conseguiram acompanhar o alvo satisfatoriamente, mas o PSO levou vantagem ao processar o vídeo em 75,66 segundos contra os 605,28 segundos necessários no caso da busca exaustiva e os 851,1 segundos que o algoritmo genético requisitou. Cada busca no GA e no PSO foi repetida, no máximo, 3 vezes. Como o tempo de processamento do GA é próximo do tempo consumido pela busca exaustiva, a repetição tornou o processo mais lento. Desta forma, o GA não conseguiu atingir o objetivo proposto para o vídeo, com a robustez almejada. O vídeo do rastreamento pelo PSO pode ser assistido em <http://youtu.be/g0lmK5M6xk>.

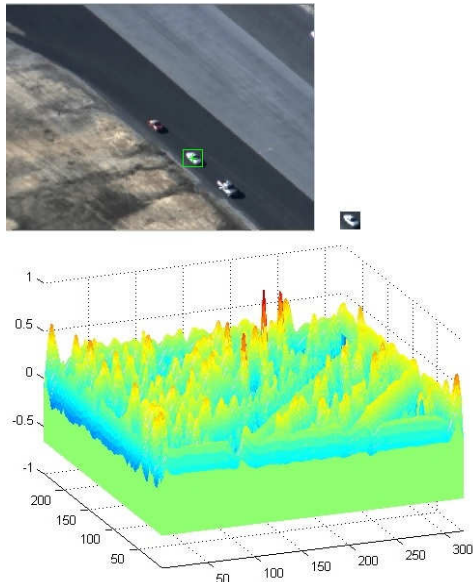


Fig. 7: Correlação ao longo do frame *carros*

VIII. CONCLUSÃO

Tanto o algoritmo genético quanto o PSO conseguem realizar o rastreamento do alvo em menos tempo que a busca exaustiva quando são considerados somente imagens isoladas. No entanto, durante o rastreamento de alvo em vídeo, o PSO levou vantagem em termos de tempo de processamento e robustez. Conclui-se que o PSO é a ferramenta mais adequada para realizar a otimização da busca de padrões em imagem por correlação, trazendo grande agilidade na tarefa. Conseguiu-se acelerar o tempo de processamento das imagens *aeronave₁*, *aeronave₂* e *carros* em 112x, 196x e 10x respectivamente. No caso do vídeo, conseguiu-se um ganho de 8x. Contudo, a desvantagem diz respeito a sua sensibilidade à escolha dos parâmetros. Em um problema real, existe a possibilidade que não haja convergência. Para corrigir tal problema, é possível repetir o algoritmo PSO algumas vezes, até que o padrão seja encontrado.

Agradecimentos

O primeiro autor agradece ao Capitão-de-Fragata (EN) Alexandre de Vasconcelos Siciliano e ao Capitão-de-Fragata (EN) Fernando Antonio Almeida Coelho pelo apoio aos estudos.

Referências

- [1] Narayana, M. "Automatic Tracking of Moving Objects in Video for Surveillance Applications", tese de mestrado da Universidade do Kansas, 2007.
- [2] Ali, A.; Kausar, H.; Khan, M.I. "Automatic visual tracking and firing system for anti aircraft machine gun". International Bhurban Conference on Applied Sciences & Technology, Islamabad, Pakistan, January 2009.
- [3] Choi, H.; Kim, Y. "UAV Guidance using a monocular-vision sensor for Aerial Target Tracking", Control Engineering Practice 22 (2014) 10–19.
- [4] K Forlenza, L.; Fasano, G.; Accardo, D.; Moccia, A. "Flight Performance Analysis of an Image Processing Algorithm for Integrated Sense-and-Avoid Systems", Hindawi Publishing Corporation, International Journal of Aerospace Engineering, Volume 2012.
- [5] Benfold, B.; Reid, I. "Stable Multi-Target Tracking in Real-Time Surveillance Video", Proc Computer Vision and Pattern Recognition (CVPR), Colorado Springs, June 2011.
- [6] Olson, T.L.P.; Sanford C.W. "A Real-Time Multistage IR Image-Based Tracker", SPIE 3692, Acquisition, Tracking, and Pointing XIII, July 1999.
- [7] Gonzalez, R.C. "Object Recognition", in Digital image processing, 3rd ed. Pearson, August 2008, pp. 861-909.
- [8] Watanabe, S. "Pattern Recognition: Human and Mechanical", New York: Wiley, 1985.
- [9] Ahuja, K.; Tuli, P. "Object Recognition by Template Matching Using Correlations and Phase Angle Method", International Journal of Advanced Research in Computer and Communication Engineering Vol.2, Issue 3, March 2013.
- [10] Mahalakshmi, T.; Muthaiah, R.; Swaminathan, P. "An Overview of Template Matching Technique in Image Processing", Research Journal of Applied Sciences, Engineering and Technology, 2012.
- [11] Sharma, P.; Kaur, M. "Classification in Pattern Recognition: A review", International Journal of Advanced Research in Computer Science and Software Engineering Vol.3, Issue 4, April 2013.
- [12] Siciliano, A.V. "Algoritmos Genéticos e Particle Swarm Optimization e suas aplicações problemas de Guerra Eletrônica", IX Simpósio de Guerra Eletrônica, Instituto Tecnológico da Aeronáutica, São José dos Campos - SP, 2007.
- [13] Esteves, C.H.M.S., "Simulação de Navegação Autônoma com Correção de Trajetória por Correlação Cruzada entre Imagens Aéreas e Imagens de Satélite", X Simpósio de Aplicações Operacionais em Áreas de Defesa, Instituto Tecnológico da Aeronáutica, São José dos Campos - SP, 2008.
- [14] Liping Zheng, Q.P.; Li, G.; Liang, J. "Improvement of Grayscale Image Segmentation Based On PSO Algorithm", Fourth International Conference on Computer Sciences and Convergence Information Technology, 2009.
- [15] Miranda, A.N. "Pearson's Correlation Coefficient: A More Realistic Threshold for Applications on Autonomous Robotics", Computer Technology and Application 5 (2014) 69-72.
- [16] Matthews, I.; Ishikawa, T.; Baker, S. "The Template Update Problem", IEEE Transactions on Pattern Analysis and Machine Intelligence, 2004.
- [17] Mitchell, M. "Genetic algorithms: An overview". Complexity, 1 (1) 31--39, 1995.
- [18] Holland, J. H. "Adaptation in Natural and Artificial Systems", Ann Arbor, The University of Michigan Press, 1975.
- [19] Karungaru, S.; Fukumi, M.; Akamatsu, N. "Face Recognition using Genetic Algorithm based Template Matching", International Symposium on Communications and Information Technologies, 2004.
- [20] Kennedy, J.; R. Eberhart. "Particle Swarm Optimization". IEEE International Conference on Neural Network. Perth, Australia, 1995.
- [21] Hsu, C. C.; Dai, G. T. "Multiple Object Tracking using Particle Swarm Optimization", World Academy of Science, Engineering and Technology, Vol:6 2012-08-21.
- [22] Yin, H.; Liu, Z.; Chai, Y.; Yang, X.; "A Novel Tracking Features Selection Method Based on Genetic Algorithm", Journal of Computational Information Systems 7: 12 (2011) 4400-4408.