

Sistema de Aeronaves Remotamente Pilotadas para Detecção Autônoma de Entidades Terrestres Móveis em Áreas Restritas

Ricardo Maroquio Bernardo^{*a}, Luis Claudio Batista da Silva^b, Paulo Fernando Ferreira Rosa^c

^aInstituto Federal de Educação, Ciência e Tecnologia do Espírito Santo, ricardo.maroquio@ifes.edu.br

^bCentro Federal de Educação Tecnológica Celso Suckow da Fonseca, luis.silva@cefet-rj.br

^cInstituto Militar de Engenharia, rpaulo@ime.cb.br

RESUMO: A vigilância de áreas restritas e a respectiva manutenção de um perímetro de segurança para instalações sensíveis à presença de pessoas não autorizadas é uma tarefa necessária tanto no meio civil quanto no meio militar e pode ser realizada de diversas formas. Este artigo apresenta um método de vigilância de áreas restritas em ambientes abertos que faz uso de um sistema de aeronaves remotamente pilotadas para detectar e rastrear, de forma autônoma, entidades terrestres móveis que adentrem um perímetro previamente estabelecido, de forma a apoiar um processo de tomada de decisão em relação a um possível evento de invasão. A solução proposta possui baixo custo aquisitivo e de manutenção, logística de implantação facilitada, redução do risco a vidas humanas e redução do tempo de percepção de uma possível invasão, além de ser portátil e de pronto emprego.

PALAVRAS-CHAVE: SARP. Robótica. Vigilância.

ABSTRACT: Surveillance of restricted areas and the respective maintenance of a security perimeter for installations that are sensitive to the presence of unauthorized persons is a necessary task in both civilian and military environments and can be carried out in different ways. This article presents a method of surveillance of restricted areas in open environments that use a remotely piloted aircraft system to autonomously detect and track mobile terrestrial entities that enter a previously established perimeter to support a decision-making process regarding a possible intrusion event. The proposed solution has low acquisition and maintenance costs, easy deployment logistics, reduced risk to human lives, and reduced time to perceive a possible invasion, in addition to being portable and ready to use.

KEYWORDS: RPAS. Robotics. Surveillance.

1. Introdução

A vigilância de áreas restritas consiste em controlar o acesso a áreas sensíveis à presença de humanos, animais, veículos ou outras entidades terrestres móveis (ETMs). Essa tarefa se torna ainda mais desafiadora quando se trata de uma área externa e sem fechamento adequado. Uma forma clássica de se resolver o problema é através do patrulhamento humano, com ou sem o auxílio de veículos, que permite que invasores sejam detectados tão logo entrem na área de visão de um dos patrulheiros.

Em áreas grandes, se houver necessidade de vigilância persistente visando encurtar o tempo de detecção, o número de patrulheiros requeridos pode inviabilizar o uso desse tipo de solução. Além disso,

essa abordagem pode gerar conflitos entre invasores e patrulheiros, colocando vidas humanas em risco. Uma alternativa ao patrulhamento humano são os circuitos fechados de TV (CFTV), que permitem realizar a vigilância a partir de monitores.

Utilizar CFTV reduz o risco a vidas e requer menos recursos humanos, entretanto, a necessidade de atenção ininterrupta do operador pode levá-lo à fadiga e à consequente falha em pouquíssimo tempo [1]. Além disso, a implantação de CFTV requer infraestrutura adequada e tempo de instalação considerável, sendo inadequado para áreas abertas ou locais que demandem vigilância temporária.

O uso de robôs móveis com capacidade sensorial aprimorada por câmeras e algoritmos de visão computacional pode apresentar vantagens relevantes para atacar o problema da detecção de invasores.

Entretanto, a eficácia desse método requer a solução de subproblemas que por si só constituem temas de pesquisa bastante desafiadores, principalmente quando se pretende realizar a tarefa de forma autônoma.

Um dos desafios inerentes ao uso de imagens capturadas a partir de câmeras embarcadas em robôs móveis diz respeito à obtenção de imagens de boa qualidade, principalmente na tarefa de detectar objetos independentemente móveis em sequências de imagens obtidas a partir de câmeras móveis, pois tanto o plano de fundo quanto o objeto móvel em si se deslocam entre os quadros. Os trabalhos [2, 3, 4] apresentam formas distintas de se lidar com esse problema.

Outra questão importante é que, quando se trata de processamento de imagens e de técnicas de visão computacional, o poder de processamento necessário é relevante e influi diretamente no tempo de resposta do sistema como um todo. Normalmente, os recursos computacionais disponíveis em computadores embarcados em robôs móveis são bastante limitados, demandando, na maioria das vezes, a transferência de responsabilidade de processamento a um computador externo.

Os recursos computacionais tendem a se tornar ainda mais escassos ou até mesmo inexistentes quando se tratam de robôs móveis aéreos de pequeno porte. Nesse tipo de robô, a capacidade de carga é ainda mais limitada e pode até mesmo demandar a transmissão das imagens para processamento em um computador externo.

Apesar dos desafios mencionados, as aeronaves remotamente pilotadas (ARPs) são um tipo de robô móvel cujo uso vem se tornando cada vez mais comum, tanto em aplicações civis quanto militares. O uso militar pode ser interessante em tarefas como vigilância e patrulhamento [5], busca e resgate [6], mapeamento [7], abastecimento [8] e comunicação [9]. No meio civil, as aplicações mais comuns incluem transporte de pacotes [10], aquisição de imagens [11], entretenimento [12] e agricultura de precisão [13].

O rápido crescimento do uso de ARPs nos últimos anos em diversas aplicações pode ser explicado pela facilidade de acesso a equipamentos anteriormente inacessíveis, seja por questões geográficas, técnicas ou econômicas. Os componentes para construção de ARPs têm tido melhorias em precisão, miniaturização, tempo de resposta e robustez, além de terem alta

disponibilidade no mercado e custo cada vez mais reduzido, fomentando seu uso e construção.

Mediante o exposto, este artigo apresenta um método relevante de detecção e rastreamento de entidades terrestres móveis em uma área restrita previamente definida por um operador humano. Como diferencial, o método propõe o uso de uma aeronave remotamente pilotada do tipo quadricóptero concebida especialmente para este fim, que atua de forma autônoma e com processamento totalmente embarcado do software inerente ao cumprimento da missão.

Em continuidade a esta introdução, a Seção 2 apresenta breve fundamentação teórica de assuntos relacionados aos subproblemas intrínsecos da solução geral. Na Seção 3, são apresentados alguns trabalhos relacionados a este. Na sequência, a Seção 4 apresenta a formalização do problema e da solução proposta, enquanto a Seção 5 apresenta detalhes do protótipo de ARP construído para a realização dos experimentos reais. Na Seção 6, são apresentados diagramas e algoritmos do software embarcado. Por fim, a Seção 7 apresenta alguns resultados experimentais e a Seção 8 conclui com algumas considerações.

2. Fundamentação teórica

Esta seção apresenta conceitos fundamentais da área de robótica móvel, aeronaves remotamente pilotadas e visão computacional. As subseções seguintes tratam cada uma desses assuntos individualmente.

2.1 Robótica móvel

Grande parte das técnicas usadas na solução proposta são pertinentes à área da robótica móvel. Os conceitos aqui apresentados foram sintetizados a partir de [14]. Os robôs móveis podem ser terrestres, aquáticos e aéreos, sendo esse último o tipo adotado como plataforma para desenvolvimento da solução aqui proposta. Os robôs aéreos podem ser de asa fixa, de asa rotativa ou mais leves do que o ar. A solução proposta faz uso de quadricópteros, que são robôs aéreos de múltiplas asas rotativas.

O grau de autonomia de um robô móvel para a realização de uma tarefa depende do aparato

computacional, sensorial e de atuação embarcados. A percepção do ambiente é realizada através do processamento de dados oriundos de sensores como câmeras e sonares. Alguns sensores também podem ser necessários para aplicações específicas, como sensores de CO₂ ou a própria câmera.

Sensores podem ser classificados sob diversos critérios. O **campo de visão** corresponde à largura de cobertura do sensor, geralmente expressa em graus. O **alcance** diz respeito à distância máxima para uma medida confiável. A **acurácia** indica o quão correta é a leitura fornecida em relação a um referencial exato. A **repetibilidade**, também tratada como a precisão do sensor, refere-se à capacidade de fornecimento da mesma medida em dada condição. A **resolução** corresponde à menor diferença possível entre dois valores sensoreados.

Outro atributo importante é o **consumo de energia**, que diz respeito à corrente e tensão requeridos pelo sensor. A **confiabilidade** do sensor refere-se à sua independência a fatores externos, como variação de tensão. A **complexidade computacional**, por sua vez, corresponde à quantidade de processamento computacional necessário para processar os dados sensoreados. Por fim, as **dimensões físicas**, a saber, o tamanho e o peso, são atributos bastante relevantes para sistemas embarcados em robôs com restrições de espaço e carga útil.

Um robô móvel pode usar diversos sensores para estimar sua localização em relação a um referencial fixo. Erros e ruídos de sensoriamento são comuns e podem ser tratados com métodos estocásticos e representações baseadas em níveis de crença. O filtro de Kalman é bastante usado para fundir dados de sensores promovendo normalização ou combinação ponderada [15]. A discretização do espaço de trabalho contínuo pode promover redução de dimensionalidade do problema, facilitando a computação [16].

2.2 Aeronaves remotamente pilotadas

Esta subseção apresenta conceitos operacionais e construtivos de ARPs do tipo quadricóptero, que no contexto deste trabalho é tratado como um tipo de robô aéreo.

A Figura 1 mostra um esquema com os dispositivos utilizados durante a operação da ARP e os tipos de mensagens trocadas entre os dispositivos. A ARP deve ter obrigatoriamente um radiocontrole transmissor (RCT) vinculado para possibilitar o controle manual da aeronave. Opcionalmente, o RCT pode receber dados de telemetria, como tensão da bateria, altitude, entre outros. O dispositivo central é uma tela de visualização de fluxo de vídeo em tempo real transmitido por uma câmera embarcada. A ECS é a estação de controle em solo e corresponde ao computador a partir do qual o operador da missão pode definir comandos em alto nível e comandos operacionais de emergência, como pouso ou redefinição de rotas.

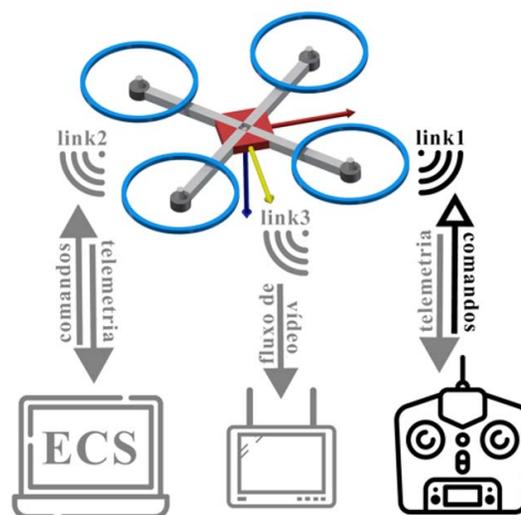


Fig. 1 – Operação básica de uma ARP do tipo quadricóptero.

A Figura 2 mostra um esquema com os componentes principais presentes em uma ARP. Os elementos em cinza claro são exteriores ao veículo. Os elementos amarelados não são obrigatórios, mas desejáveis. Os retângulos tracejados representam grupos lógicos de elementos, não significando necessariamente que eles pertençam à mesma placa de circuito impresso. As setas indicam os sentidos dos fluxos de dados entre os componentes.

A controladora de voo, representada pelo retângulo tracejado central, recebe os dados dos sensores em sua unidade de processamento para computar a posição e a orientação da ARP em relação a um referencial fixo.

A unidade de processamento é dedicada à execução do software de controle da aeronave. O retângulo pontilhado à direita representa os componentes usados para aplicações específicas, enquanto o retângulo pontilhado à esquerda corresponde aos componentes usados no modo de pilotagem em primeira pessoa, em que o operador visualiza em tempo real as imagens captadas por uma câmera embarcada.

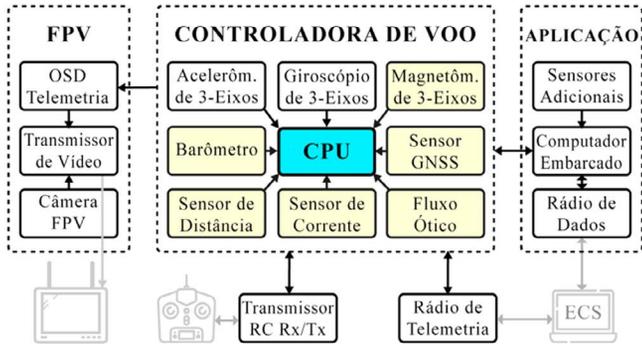


Fig. 2 – Organização básica dos componentes internos e externos do quadricóptero.

O sistema de propulsão de um quadricóptero é constituído por quatro motores equipados com hélices e comandados por controladores eletrônicos de velocidade (ou ESC, do inglês *electronic speed controller*). A combinação correta do motor e da hélice, bem como do ESC e da bateria, é fundamental para definir a qualidade do voo. O diferencial de rotação entre os quatro conjuntos propulsores é o que define o comportamento dinâmico da aeronave. A Figura 3 mostra alguns movimentos de acordo com o diferencial de rotação dos propulsores.

2.3 Visão computacional

Conforme [17], a visão computacional é uma área de estudo que visa compreender uma cena em 3D a partir de imagens 2D, imitando a visão humana usando software e hardware. A qualidade das imagens é fundamental para essa compreensão. É comum a utilização de sistemas de visão implementados em ARPs para uso em tarefas como detecção e rastreamento de objetos independentemente móveis (OIM), estimação de posição, navegação, detecção de obstáculos, pouso autônomo, voo estabilizado, determinação de atitude, reconstrução 3D, entre outras [18-25].

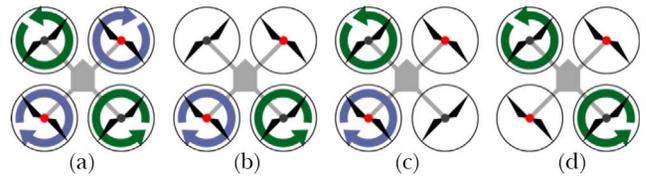


Fig. 3 – Dinâmica básica de voo de um quadricóptero em função da rotação de cada propulsor. a) subida; b) inclinação para a frente; c) inclinação para a direita; d) guinada para a direita.

A extração de características das imagens é fundamental para criar descritores que representem pontos específicos da imagem. A atomicidade de um descritor, bem como sua robustez a variações de iluminação e de rotação, por exemplo, permite rastreá-lo ao longo de vários quadros e obter informações de movimentação de partes da imagem.

Existem diferentes algoritmos para computar descritores de pontos, e normalmente a robustez do descritor incorre em maior tempo de processamento. Cada algoritmo tem suas vantagens e desvantagens, sobressaindo-se melhor ou pior de acordo com a situação.

A detecção de OIMs consiste em segmentar as imagens de uma sequência em plano de fundo e plano móvel. Os OIMs ficam no plano móvel, enquanto o cenário de atuação fica no plano de fundo. As principais técnicas para detecção de OIMs são a diferenciação de quadros, o fluxo ótico denso ou esparsa, e a subtração de plano de fundo [26].

A diferenciação de quadros considera a subtração de dois quadros consecutivos da sequência. Tem a vantagem de ser simples e rápido, porém, como somente parte de um OIM se move entre dois quadros contíguos, o OIM não é detectado por completo. No método do fluxo ótico, vetores são traçados entre descritores correspondentes de dois quadros subsequentes e, de acordo com a magnitude desses vetores, é possível ter uma noção das regiões da imagem que estão em movimento, mas sem obter o contorno completo dos OIMs.

No método de subtração de plano de fundo, um quadro inicial é mantido como referencial e atualizado a cada novo quadro, mantendo uma representação viável do plano de fundo. A cada quadro vindouro, a subtração é feita e a diferença é considerada um OIM. Esse método identifica bem OIMs, mas requer

a atualização adequada do quadro de referência. O rastreamento de um OIM consiste no registro de sua posição ao longo do tempo, identificando continuamente os objetos correspondentes entre os quadros da sequência. Problemas temporários, como oclusão de objetos, podem ser contornados usando técnicas de predição.

3. Revisão de literatura

Os trabalhos [27-29], assim como este, são alguns dos vários casos em que os autores propõem suas próprias ARPs durante o desenvolvimento de sua pesquisa. A Tabela 1 mostra as principais características das aeronaves construídas nos trabalhos supracitados. A coluna referente ao modo de voo disponível diz respeito ao modo de voo mais sofisticado oferecido pela ARP.

Tabela 1: Comparação das principais características de aeronaves remotamente pilotadas utilizadas nos trabalhos acadêmicos citados.

Ref.	Peso Total (g)	Tempo de Voo (min)	Modo de Voo Disponível	Custo Total (US\$)
[26]	4000	11	Estabilizado	NI
[27]	1400	5-10	Estabilizado	300,00
[28]	2250	30	Automático	1500,00

Na seção introdutória, foram citadas diversas aplicações que fazem uso de ARPs, sendo que algumas delas podem demandar detecção e/ou rastreamento de OIMs. Os parágrafos a seguir descrevem sucintamente alguns trabalhos que fazem uso de ARPs para detectar e/ou rastrear ETMs com algum propósito. No contexto deste trabalho, o termo ETM é usado em substituição a OIM para caracterizar melhor o tipo de objeto alvo da detecção.

No artigo [30], que compartilha três autores com o presente trabalho, foi proposto um sistema para controlar várias ARPs de pequeno porte para a vigilância de uma área predefinida. A solução proposta visa ao controle de posicionamento e deslocamento das aeronaves dentro da área alvo da vigilância, sendo que tal área é decomposta em células hexagonais com diferentes prioridades de cobertura. Uma política de distribuição de ARPs baseada nas prioridades define a frequência de visitação de cada célula. A solução

permite que um único operador comande uma frota de ARPs autônomas usando uma interface de usuário de alto nível, de forma que não há necessidade de controle manual individual das ARPs.

Outros trabalhos mais recentes atacam o problema da vigilância persistente de áreas restritas usando enxames de ARPs. No artigo [31], os autores apresentam um estudo teórico envolvendo a análise matemática da capacidade de coleta de dados de um enxame de ARPs voando em formação circular, reta e diagonal, sobre uma área decomposta em células quadradas, mantendo uma taxa de atualização uniforme de cada célula alcançada pelas ARPs.

No trabalho apresentado em [32], os autores combinam o uso de ARPs com veículos terrestres não-tripulados para a cobertura de áreas urbanas, atendendo a demandas de sensoriamento visual detalhada em alguns locais. A solução proposta discretiza o espaço de atuação da ARP em cubos e do veículo terrestre em quadrados projetados no solo a partir desses cubos. Um caminho para a atuação cooperativa dos veículos de forma que resolva o problema da cobertura com restrições é modelado como um problema de otimização, e a solução proposta usa uma abordagem híbrida de algoritmos genéticos e algoritmos de estimação de distribuição.

Em [33], os autores apresentam uma solução que usa um enxame de ARPs comerciais para vigilância aérea semiautônoma. Um operador define manualmente trajetórias circulares para as ARPs de acordo com um método que evita a sobreposição das imagens capturadas pelas câmeras embarcadas. Essas imagens são então enviadas continuamente para uma estação em solo que, por sua vez, processa as imagens para detectar objetos de interesse, que não são necessariamente objetos móveis. A quantidade de ARPs e a frequência de cobertura podem variar de acordo com o tamanho da área.

A vigilância de fronteiras usando drones e uma rede de sensores estacionários em solo é tema do trabalho apresentado em [34]. Uma rede de sensores em terra composta por sonares e sensores infravermelhos é usada para detectar movimentos no ambiente. Quando um movimento é detectado,

uma ARP é acionada para ir até o local e capturar imagens da região. Tais imagens são transmitidas a uma estação em terra que, por sua vez, processa as imagens buscando encontrar pessoas.

O estabelecimento de políticas de posicionamento de ARPs tem sido tema de estudo de pesquisa para algumas aplicações. Em [35], os autores propõem um framework para posicionar uma ARP equipada com os devidos dispositivos, visando maximizar a conectividade de estruturas de rede backhaul, que conectam backbones a redes periféricas. No artigo [36], os autores estabelecem um método de posicionamento de ARPs objetivando prover uma rede wireless para usuários em uma área terrestre quadrada previamente definida, buscando maximizar a quantidade de usuários cobertos pela rede, com a possibilidade de clusterização de grupos de usuários e utilização de múltiplas ARPs. O trabalho apresentado em [37] tem semelhanças com [36] no que diz respeito ao uso de ARPs para provimento de conectividade e serviços de comunicação para usuários em solo. Os autores de [37] focam no estabelecimento de uma altitude ótima de operação considerando a física de propagação do sinal e levando em conta a rugosidade do solo.

4. Formalização do problema

A Figura 4 mostra uma representação 3D em alto nível de um cenário de vigilância simplificado, em que um carro adentra uma área restrita que envolve uma edificação. Os muros no entorno da edificação definem a área restrita. No topo, está a ARP de vigilância com uma câmera apontada para baixo, processando continuamente as imagens capturadas. O campo de visão da câmera é representado pelo volume piramidal semitransparente entre a ARP e o solo. Pode-se notar que o campo de visão abrange todo o círculo que representa a área restrita. Na estrada que dá acesso à edificação, há um veículo dentro da área restrita, que representa uma ETM.

4.1 Formalizações preliminares

Em síntese, os elementos relevantes para o problema são a área restrita a ser vigiada, as entidades terrestres

móveis, a ARP e o seu espaço de trabalho. Portanto, seja W um ambiente externo que compreende um conjunto de coordenadas georreferenciadas, onde cada $w_i \in W$ é um ponto sobre a superfície terrestre dado por $[latitude, longitude, altitude]$, a **área restrita** terrestre A a ser vigiada é um subconjunto de W manualmente definido, cujo contorno C é um polígono fechado de coordenadas $c_j \in A$.

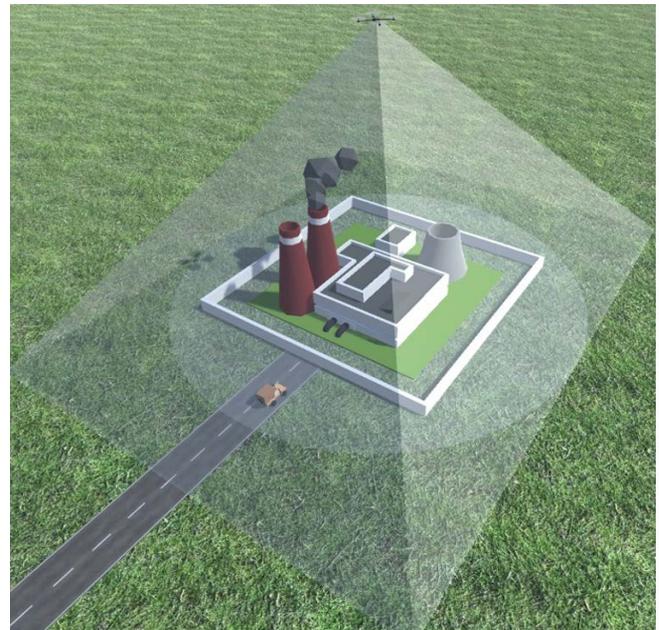


Fig. 4 – Representação 3D de um cenário de vigilância.

A área restrita decomposta em n_d células é representada pelo conjunto $D = \{d_i \mid 0 \leq i < n_d\}$, cujos elementos assumem os valores de coordenadas dos centroides das células. O ambiente W pode conter um conjunto E com n_e **entidades terrestres móveis** (ETMs) que podem adentrar a área restrita a qualquer momento. Portanto, $E = \{e_i \mid 0 \leq i < n_e\}$, cujos elementos correspondem às coordenadas de seus centroides.

Inicialmente, todas as ETMs estão fora da área restrita, isto é, $\forall e_i \in E, e_i \in (W - A)$. Quando nessa condição, as ETMs são representadas apenas para fins de formulação do problema, não tendo elementos correspondentes na solução concreta. A partir do momento em que a ETM adentra a área restrita, ela passa a fazer parte de um subconjunto de E denominado E_{det} , que corresponde às ETMs detectadas.

Seja *ARP* uma **ARP estacionária** do tipo VTOL (Vertical Take Off and Landing) com posição de partida $P^0 \in D$, ou seja, *ARP* se encontra dentro de uma célula de *D*, mas não necessariamente em seu centroide. *ARP* deve ser capaz de pairar em uma posição P^{pairar} previamente calculada, cuja projeção no solo pertence à *A* e, a partir dessa posição, deve ser capaz de realizar a cobertura visual total de *A* usando uma câmera embarcada.

A imagem img^t de dimensões $[w, h]$ obtida a partir da câmera embarcada em *ARP* no tempo *t* é representada por $img^t = \{p^t(x, y) \in \mathcal{R}^3 \mid 0 \leq x < w, 0 \leq y < h\}$, sendo que cada elemento $p^t(x, y)$ da imagem armazena os valores RGB do pixel na coordenada (x, y) da imagem capturada no tempo *t*. O conjunto formado pela sequência de imagens capturadas pela câmera de *ARP* é representado por *Img*.

4.2 Formalização da detecção

A detecção de ETMs invasoras requer o processamento individual dos quadros capturados pela câmera embarcada em *ARP*. De acordo com a formalização apresentada na subseção anterior, o conjunto de imagens capturadas pela câmera embarcada em *ARP* é definido por *Img*. A cada imagem $img^t \in Img$, para $t = 1, \dots, t_{curr}$ onde *t* é um índice sequencial temporal e t_{curr} é o índice mais atual dessa sequência, a função de detecção f_{det} deve retornar como saída um vetor de n_{det}^t pares $(x_{det^i}^t, y_{det^i}^t)$ correspondentes às coordenadas dos pixels $p_{est}^t(x_{det^i}^t, y_{det^i}^t)$ localizados nos centroides dos n_{det}^t OIMs detectados em img^t , para $i = 1, \dots, n_{det}^t$. O problema da detecção, portanto, pode ser representado pela equação 1 a seguir:

$$E_{det}^t = \{(x_{det^i}^t, y_{det^i}^t)\} \leftarrow f_{det}(img^t), 1 \leq i \leq n_{det}^t, \quad (1)$$

$$1 \leq t \leq t_{curr}$$

onde E^t é o conjunto composto pelas ETMs detectadas no tempo *t*, n_{det}^t é a quantidade de OIMs detectados no instante *t* e t_{curr} é o índice sequencial temporal da imagem mais atual da sequência *Img*. Como o resultado da função de detecção é um vetor de centroides baseados em coordenadas de pixels, é necessário ter uma função adicional que converta essas coordenadas para valores georreferenciados,

uma vez que o mecanismo de navegação de *ARP* é baseado em um sensor de localização por satélites.

Portanto, $f_{gps}(x, y, lat, long, alt, cam, heading)$ é a função que converte uma coordenada (x, y) de um pixel de uma imagem *Img* de para uma coordenada georreferenciada, onde $(lat, long, alt)$ são as coordenadas georreferenciadas de *ARP* no momento da captura da imagem, *cam* corresponde aos parâmetros intrínsecos da câmera e *heading* é a orientação do veículo em relação ao norte. Vale observar que OIM é a denominação dada aos objetos independentemente móveis detectados na imagem que, posteriormente, serão ou não caracterizados como uma ETM.

4.3 Formalização do rastreamento

Para a realização do rastreamento das ETMs invasoras, é necessário encontrar a correspondência entre as ETMs detectadas ao longo do tempo. Portanto, seja f_{track} a função capaz de determinar as correspondências entre E_{det}^t e E_{det}^{t-1} . Para isso, como entrada, f_{track} recebe continuamente o resultado de $f_{det}^t(img^t) = E_{det}^t$ para $t > 1$, bem como as entidades detectadas no instante anterior, E_{det}^{t-1} , e o conjunto de rastreamentos de cada entidade em E_{det}^{t-1} , denominado R^{t-1} .

Como saída, a função deve retornar um conjunto E_{det}^t contendo o conjunto atualizado de ETMs invasoras, bem como o conjunto R^t atualizado dos dados de rastreamento de cada ETM do conjunto E_{det}^t . O problema do rastreamento, portanto, pode ser representado pela equação 2 a seguir:

$$(E_{det}^t, R^t) = \left(\left\{ (x_{e^i}^t, y_{e^i}^t) \right\}, \left\{ R_{e^i}^t \right\} \right);$$

$$(E_{det}^t, R^t) \leftarrow f_{track}(E_{det}^t, E_{det}^{t-1}, R^{t-1}) \mid 1 \leq i \leq n_{det}^t \quad (2)$$

$$2 \leq t \leq t_{curr}$$

onde E_{det}^t corresponde às coordenadas de todas as ETMs pertencentes ao conjunto E_{det}^t no instante *t*, R^t corresponde aos vetores de coordenadas de posição ao longo do tempo para toda $e_{det}^i \in E_{det}^t$, $(x_{e^i}^t, y_{e^i}^t)$ corresponde à posição de e_{det}^i no instante *t* e $R_{e^i}^t$ corresponde ao conjunto de posições de e_{det}^i desde o instante 0 até o instante *t*.

Vale observar que manter o conjunto E_{det}^t atualizado corresponde a manter apenas as ETMs presentes na área restrita A no instante t , ou seja, f_{track} deve manter uma política de inclusão e exclusão de ETMs em E_{det}^t , uma vez que algumas ETMs de E_{det}^{t-1} , podem não estar presentes em A no instante t , assim como algumas ETMs de E_{det}^t podem ter acabado de adentrar a área restrita e não terão correspondência em E_{det}^{t-1} .

4.4 Restrições, premissas e requisitos adicionais

Considera-se que ARP possui capacidade de navegação automática por *waypoints* georreferenciados. Como o sistema depende de câmeras do tipo RGB, é fundamental que o cenário de operações possua iluminação adequada. Considera-se ainda que o terreno seja plano e que a área restrita A seja estacionária, isto é, sua localização e forma perimetral não se alteram ao longo da missão. A trajetória realizada entre o ponto de decolagem de ARP e a posição de vigilância desconsidera a presença de obstáculos com altura superior à altitude de operação da aeronave. ARP deve ser do tipo VTOL e capaz de se comunicar com uma estação de controle em solo, que, por sua vez, é a interface com o operador da missão.

5. Solução proposta

A Figura 5 apresenta uma visão simplificada da arquitetura do SARP (Sistema de Aeronave Remotamente Pilotada) usado na solução proposta. Ele é composto por uma ARP estacionária e uma estação de controle em solo que se comunicam por um canal bidirecional entre si. A ARP detecta e rastreia ETMs invasoras usando somente o aparato embarcado no veículo, transmitindo continuamente à ECS dados referentes às ETMs (posição, velocidade e tamanho) e dados de telemetria (bateria, posição, inclinação e estado dos sensores).

Complementando a Figura 5, a Figura 6 apresenta uma visão comportamental do SARP ao longo de uma missão, destacando as tarefas de alto nível envolvidas na solução do problema e a sequência em que tais tarefas são executadas. O retângulo tracejado corresponde às

coordenadas dos vértices que definem um polígono correspondente ao perímetro da área restrita, que são fornecidas pelo operador da missão como dados de entrada. O sistema então inicia computando a posição e orientação (pose) ideais da ARP , de forma a manter cobertura visual total da área restrita.

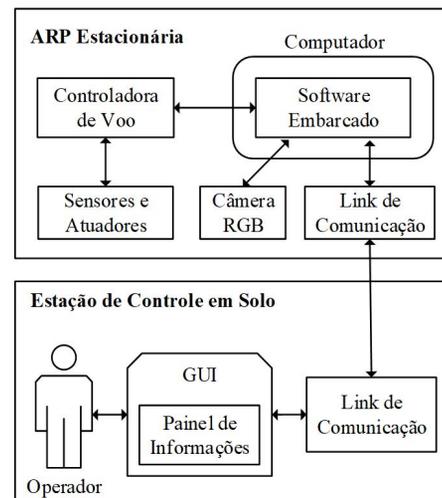


Fig. 5 – Arquitetura simplificada do SARP aplicado na solução.

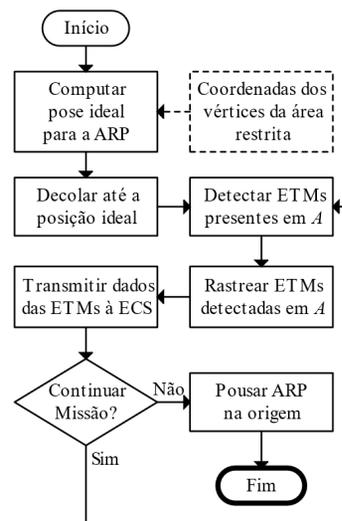


Fig. 6 – Visão em alto nível das tarefas executadas pelo SARP.

Com a pose computada, a ARP decola e navega até alcançá-la. Em seguida, o algoritmo de detecção de ETMs é iniciado, obtendo um quadro de referência e processando os quadros vindouros em busca de alterações que caracterizem OIMs presentes na imagem. Com os resultados da detecção, o algoritmo de rastreamento entra em ação visando manter um registro temporal do

posicionamento das ETMs detectadas. Concluindo um ciclo, informações estáticas e dinâmicas referentes às ETMs são transmitidas para a ECS.

Caso a missão deva prosseguir, um novo ciclo de detecção-rastreamento-transmissão é iniciado. A missão pode ser finalizada de forma deliberada pelo operador ou de forma automática mediante o estado da ARP. As subseções seguintes apresentam detalhes das tarefas de alto nível apresentadas nesta subseção.

5.1 Posicionamento da ARP

Para maior utilidade das imagens capturadas, a ARP deve pairar sobre a área restrita A de forma que o campo de visão seja ajustado para que a área restrita compreenda a maior quantidade possível de pixels, visando maximizar a relação pixels/área. Dessa forma, ETMs que adentrem A ocuparão mais pixels, facilitando a detecção. Considerando que o sensor da câmera é retangular, para maximizar a relação pixel/área, primeiramente, é computado um retângulo $Rect$ de área mínima que envolve toda a área restrita A .

Então, a latitude e longitude da ARP devem coincidir com o centro de $Rect$, e a orientação da ARP deve ser definida de forma que o lado de maior comprimento do retângulo computado e o lado de maior comprimento da imagem capturada sejam paralelos e os centros sejam coincidentes. Com a pose da ARP computada, é necessário calcular a altitude em que a aeronave deve pairar para que a câmera tenha abrangência completa da área restrita. Nesse caso, considera-se que os parâmetros relacionados à lente da câmera são previamente conhecidos.

Conforme a equação 3 [38], as dimensões (W , L) da área de cobertura de uma câmera planar ao solo a uma altitude h podem ser calculadas da seguinte forma, onde α corresponde ao ângulo de abertura vertical da câmera e β ao ângulo de abertura horizontal da câmera:

$$\begin{aligned} W &= 2h * \tan\left(\frac{\alpha}{2}\right) \\ L &= 2h * \tan\left(\frac{\beta}{2}\right) \end{aligned} \quad (3)$$

Portanto, a altitude mínima h_{min} admitida para a câmera da ARP, de forma que ela faça a cobertura total de A , pode ser derivada da equação 3, como segue:

$$\begin{aligned} h_W &= \frac{W}{2 * \tan\left(\frac{\alpha}{2}\right)} \\ h_L &= \frac{L}{2 * \tan\left(\frac{\beta}{2}\right)} \\ h_{min} &= \max(h_W, h_L) \end{aligned} \quad (4)$$

Esses cálculos desprezam deslocamentos horizontais e verticais provocados por intempéries do ambiente e erros nos sensores da ARP. Portanto, é recomendado usar uma margem de segurança para evitar a possível falta de cobertura de regiões próximas às bordas de A . A Figura 7 apresenta uma imagem de uma área restrita fictícia a ser vigiada.



Fig. 7 – Área restrita fictícia a ser vigiada e representações da área de abrangência da câmera em diferentes situações.

Na Figura 7, o polígono com arestas sólidas verde-escuras corresponde à área restrita A . O retângulo com arestas sólidas amarelas corresponde ao retângulo de área mínima que envolve a área restrita. O retângulo com arestas em pontilhado azul corresponde à área coberta pela câmera da ARP quando a aeronave está na pose computada previamente, representada pelo círculo branco com borda vermelha. Para fins de comparação, o retângulo maior, com arestas

pontilhadas amarelas, corresponde à imagem capturada pela câmera na *ARP* quando a aeronave estiver pairando na posição horizontal da pose computada, mas a uma altitude superior e com uma orientação diferente.

O retângulo de arestas vermelhas dentro da área restrita corresponde a um objeto de interesse aleatório. Na imagem compreendida pelo retângulo pontilhado azul, esse objeto ocupa cerca de 2,3% da imagem, enquanto ocupa 1,4% da imagem compreendida pelo retângulo pontilhado amarelo, ou seja, na imagem capturada a partir da pose ideal o objeto teria aproximadamente 64% mais pixels, tornando-o mais “visível” na fase de detecção de ETMs.

5.2 Estabilização de imagens

Vibrações provocadas por motores e movimentos causados por intempéries do ambiente ou por imprecisão de sensores estão entre os causadores de problemas em imagens capturadas a partir de ARPs. Manter o alinhamento dos quadros capturados pela câmera é crucial para algoritmos de detecção de OIMs. Este trabalho utiliza uma solução híbrida baseada em estabilização mecânica com um *gimbal* motorizado e estabilização por software. Para movimentos angulares de rolagem e arfagem, o *gimbal* é acionado. A técnica utilizada foi desenvolvida pelos mesmos autores deste artigo e é apresentada em [39].

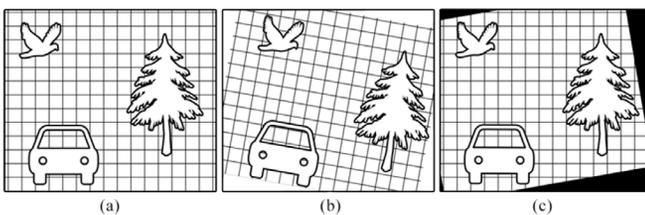


Fig. 8 – Geração de bordas inválidas causada pela transformação geométrica aplicada para correção dos movimentos indesejados da câmera. (a) quadro de referência; (b) quadro a ser alinhado; (c) quadro corrigido.

A Figura 8 mostra um exemplo de alinhamento que cria bordas inválidas, com (a) apresentando o quadro de referência, (b) apresentando o quadro a ser alinhado e (c) mostrando o quadro corrigido pela transformação

geométrica e as bordas pretas surgidas como efeito colateral. Vale mencionar que o tamanho das bordas pretas aumenta de forma proporcional aos movimentos da câmera, portanto, a manutenção de estabilidade da *ARP* é algo de fundamental importância.

5.3 Detecção e rastreamento de ETMs

A detecção e o rastreamento de ETMs consideram que o eixo ótico da câmera embarcada tem a mesma direção e sentido do vetor gravidade, graças ao recurso de estabilização de imagens. Portanto, as imagens capturadas do solo são coplanares, facilitando a etapa de detecção de ETMs. O diagrama da Figura 9 apresenta uma visão geral do algoritmo de detecção e rastreamento de ETMs.

Um ciclo de detecção e rastreamento inicia-se a partir do **fluxo de vídeo** da câmera embarcada em *ARP*. Esse ciclo se repete para cada quadro processado. Primeiramente, captura-se um quadro para usar como **quadro de referência**. Cada novo quadro, representado no diagrama pelo bloco com o rótulo **quadro atual**, passa, primeiramente, por um processo de alinhamento, de forma que a imagem do quadro atual coincida com a imagem do quadro de referência. Em seguida, o quadro atual é subtraído do quadro de referência e o resultado é submetido a um limiar que vai determinar se cada pixel resultante deve ser desprezado (0) ou considerado (1), gerando uma imagem binária.

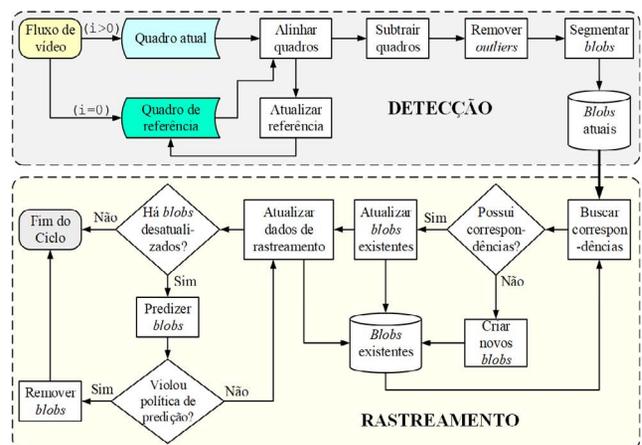


Fig. 9 – Visão geral do algoritmo de detecção e rastreamento de ETMs.

Os agrupamentos de pixels binários contíguos formam *blobs* (do inglês *binary large objects*). Os *blobs* de tamanho superior a um segundo limiar são mantidos, enquanto os menores são considerados *outliers* e removidos da imagem. Por fim, os *blobs* são devidamente segmentados do plano de fundo e armazenados na coleção ***blobs atuais***. Esses *blobs* são os candidatos às ETMs detectadas. A partir desse ponto é iniciada a fase de rastreamento.

No primeiro ciclo, os *blobs* passarão a constituir automaticamente a coleção de ***blobs existentes***. No ciclo seguinte, os *blobs* atuais são comparados com aqueles computados no ciclo anterior, de forma a se **buscar correspondências** entre as coleções de *blobs*. Assim, é possível manter um histórico de posições dos *blobs* ao longo do tempo.

Em vista da taxa de processamento de quadros, da resolução da câmera e da altitude da *ARP*, os *blobs* deslocam-se por poucos pixels entre um quadro e outro. Por conta disso, a busca por uma correspondência se resume a encontrar um *blob* na vizinhança que tenha as mesmas características físicas (razão de aspecto e área) e comportamentais (direção e velocidade) que o *blob* do quadro atual, sendo que o raio de busca é um parâmetro que influi no desempenho.

Quando algum dos *blobs* atuais não possui um *blob* existente correspondente, significa que ele é um novo *blob*, ou seja, trata-se de uma nova ETM que acabou de adentrar a área restrita. Os *blobs* atuais que tiveram *blobs* existentes correspondentes terão suas informações e dados de rastreamento atualizados. Caso algum *blob* existente não tenha correspondente, pode ser que ele tenha deixado a área restrita ou esteja passando por um evento de oclusão.

Os casos de oclusão são tratados através de uma política de predição, em que a próxima posição do *blob* ocluído é computada de acordo com seu comportamento dinâmico, baseado em seu histórico de posições ao longo do tempo. A política de predição pode excluir um *blob* ou atualizar seus dados de rastreamento com a nova posição estimada. A exclusão pode ocorrer quando a posição do *blob* indicar que ele deixou a área restrita ou quando ele está sem correspondência por um tempo acima de um limiar ajustável.

5.4 A ARP construída

Para a condução de experimentos de partes intrínsecas do SARP proposto, foi construída uma ARP com características que atendem a todas as premissas do problema. A metodologia construtiva evoluiu de um projeto anterior publicado em [40]. A ARP usa componentes de prateleira com alta disponibilidade comercial, sendo que cada componente foi criteriosamente selecionado a partir de testes empíricos, individuais e integrados, usando equipamentos computadorizados.

A ARP construída possui 330 mm de envergadura, autonomia de voo pairado superior a 40 minutos e peso final de 1080 g, enquanto seu antecessor tinha envergadura de 450 mm, voava até 23 minutos e pesava 1,5 kg. As evoluções na parte sensorial também foram relevantes. O ganho de autonomia deve-se, principalmente, à bateria de células de íons de lítio construída especificamente para esse veículo. Vale mencionar ainda que o *gimbal* usado para a estabilização mecânica também foi construído usando componentes de prateleira. A Figura 10 mostra uma foto real da ARP construída para este trabalho.



Fig. 10 – ARP de 330 mm construída para desempenhar o papel de ARP estacionária.

Para as tarefas de navegação e controle, foi desenvolvida uma biblioteca com funções em alto nível para movimentação da ARP, disponível em [41]. Ela é baseada em outra de abstração inferior chamada DroneKit [42], que utiliza um protocolo específico para comunicação com ARPs de pequeno porte chamado MavLink [43].

6. Resultados experimentais

Dada a complexidade de se criar um cenário real para validação integral do SARP em uma missão de vigilância, as partes intrínsecas do sistema foram validadas individualmente. Para validar a ARP construída, foram realizados testes de voo reais e simulados por computador. A simulação por computador utilizou uma arquitetura *Software in the Loop* (SITL) disponibilizada pela equipe que desenvolve o ArduPilot [44], que é o firmware da controladora de voo. A Figura 11 mostra os resultados obtidos.



(a)



(b)

Fig. 11 – Trajetórias executadas pela ARP simulada com SITL (traço azul) e pela ARP em voo real (traço laranja). (a) visualização em 2D; (b) visualização em 3D.

Os gráficos da Figura 12 mostram a diferença de latitude, longitude e altitude entre os experimentos de voo real e por SITL. A diferença entre os resultados é desprezível para esse tipo de aplicação e demonstra

claramente que a precisão de navegação da ARP habilita o uso do simulador SITL para validação inicial de algoritmos de navegação.

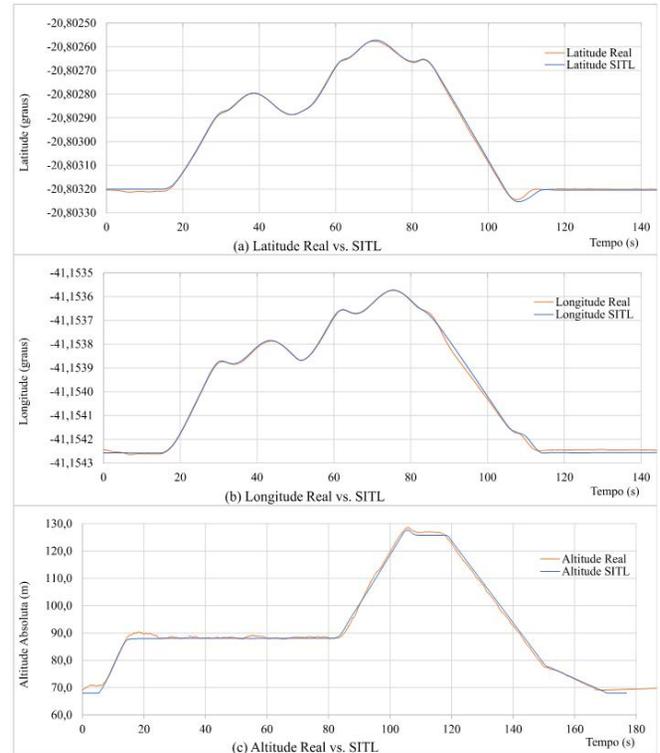


Fig. 12 – Latitude (a), longitude (b) e altitude (c) das trajetórias geradas pela simulação com SITL (linha azul) e com a ARP em voo real (linha laranja).

Para validar o algoritmo de detecção e rastreamento, um simulador computacional foi concebido para gerar animações de ETMs adentrando uma área restrita, passando por situações de oclusão e cruzando com outras ETMs. A Figura 13 mostra capturas de tela do algoritmo de detecção e rastreamento em ação. A primeira captura (a) mostra uma região verde correspondente ao terreno, retângulos marrons representando regiões de oclusão e um retângulo branco representando uma ETM. A segunda captura (b) mostra a imagem binária com a ETM destacada do plano de fundo. A terceira captura (c) mostra o rastro da ETM ao longo do tempo, incluindo posições determinadas por predição.

O simulador também gera dados de *ground truth*. O gráfico da Figura 14 mostra alguns resultados qualitativos do processo de detecção e rastreamento

comparando dados de rastreamento gerados pelo algoritmo com os dados de *ground truth* gerados pelo simulador.

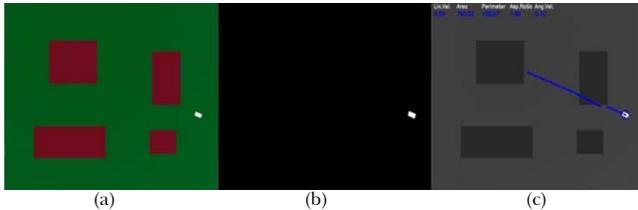


Fig. 13 – Resultado do algoritmo de rastreamento aplicado em animação gerada pelo simulador. (a) imagem original; (b) imagem subtraída do plano de fundo; (c) rastreamento computado.

Na legenda mostrada na parte superior do gráfico da Figura 14, “GT” significa “*ground truth*”, “Rast” significa “rastreamento”, “S.O.” significa “sem oclusão” e “C.O.” significa “com oclusão”. O gráfico possui oito linhas que representam trajetórias percorridas pelas respectivas ETMs ao longo da imagem, sendo quatro geradas pelos dados de *ground truth* (linhas grossas) e quatro computadas pelo algoritmo de detecção e rastreamento (linhas estreitas). Cada ponto gerador de tais linhas refere-se ao centroide da ETM que percorreu a trajetória representada pela linha. A linha relacionada ao *ground truth* foi alargada para facilitar a comparação, uma vez que o gráfico apresenta muitas sobreposições.

O primeiro experimento, representado na Figura 14 pelas linhas cinza e amarela, corresponde a uma ETM movendo-se em linha reta sem sofrer oclusão. A linha amarela sofre variações pequenas, mas não deixa de ocupar a região central da linha de *ground truth*. O segundo experimento, representado na Figura 14 pelas linhas preta e azul-clara, corresponde a uma ETM movendo-se em linha reta e sofrendo algumas oclusões. O rastreamento computado também ficou próximo ao *ground truth*, exceto pela região destacada, onde ocorreu uma situação de oclusão que demandou o uso mais intenso do mecanismo de predição.

O terceiro experimento, representado na Figura 14 pelas linhas vermelha e branca, simulou uma ETM em um trajeto com curvas sem sofrer oclusão. O rastreamento também ficou bem próximo ao *ground truth*. O quarto e último experimento, representado

na Figura 14 pelas linhas azul e verde, simulou uma ETM em uma trajetória com curvas e sofrendo oclusões em alguns pontos. Nesse experimento, o rastro computado ficou próximo ao *ground truth*, porém, destaca um trecho em que uma oclusão provocou um leve desvio na trajetória estimada pela política de predição.

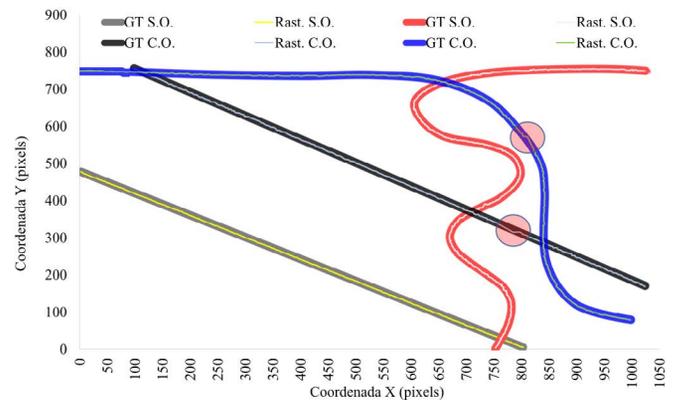


Fig. 14 – Comparação entre trajetórias *ground truth* e trajetórias computadas pelo algoritmo de detecção e rastreamento.

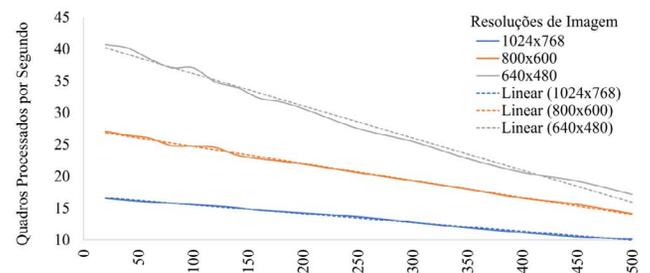


Fig. 15 – Influência da quantidade de ETMs e da resolução de vídeo na taxa de quadros processados por segundo.

Para validação do tempo de execução, foram geradas 16 animações simulando grupos de 20 a 500 ETMs passando simultaneamente pela cena. As animações foram capturadas em três resoluções de vídeo distintas, a saber, 1024×768 , 800×600 e 640×480 , representando diferentes resoluções de câmera. A Figura 15 mostra a relação entre a quantidade de ETMs e a taxa de quadros por segundo para cada resolução testada. As linhas pontilhadas são apenas para demonstrar uma tendência linear na complexidade do algoritmo.

Os experimentos foram executados em um Raspberry Pi 4B com 4GB de RAM, com temperatura mantida entre 41 e 45 graus Celsius. Uma observação já esperada é que o melhor caso ocorre com a presença de 20 ETMs, enquanto o pior caso ocorre com a presença de 500 ETMs. Para a resolução de vídeo 640×480 , a taxa de quadros por segundo variou entre 17,1 (pior caso) e 40,7 (melhor caso), incorrendo em um tempo de processamento por quadro de 58 ms e 24 ms, respectivamente. Na resolução de vídeo 800×600 , a taxa de quadros por segundo variou entre 14,1 e 27,1, demandando 71 ms e 37 ms por quadro, respectivamente. Com a resolução de 1024×768 , a taxa de quadros por segundo variou entre 10,1 e 16,5, demandando 99 ms e 61 ms por quadro, respectivamente.

Nota-se que quanto maior é a resolução do vídeo, menor é o impacto da quantidade de ETMs no tempo de processamento, indicando que o algoritmo passa a maior parte do tempo realizando processamento de imagem e menos tempo gerenciando as ETMs presentes na cena. A tendência linear e o fato da quantidade de ETMs presentes na cena ser menos impactante no tempo de execução do que as demais variáveis envolvidas contribuem para facilitar o dimensionamento dos recursos computacionais necessários para alguma aplicação específica que demande, por exemplo, vigilância de áreas extensas e o uso de câmeras de maior resolução.

Por fim, vale mencionar que, em termos qualitativos, o tamanho mínimo da ETM detectável varia de acordo com a quantidade de pixels que ela ocupa na imagem e com o contraste que ela possui em relação ao plano de fundo (solo). O algoritmo de detecção e rastreamento proposto se mostrou capaz de rastrear ETMs ocupando no mínimo quatro pixels nos vídeos gerados por simulação. Entretanto, em ambientes reais visualmente ruidosos, o tamanho mínimo da ETM detectável tende a aumentar na medida em que a intensidade dos ruídos aumenta, sendo que o estabelecimento dessa tendência demanda estudos mais específicos.

7. Considerações finais

Este artigo apresentou um SARP para detecção autônoma de ETMs em áreas restritas em ambientes abertos, pensado para situações em que os meios convencionais de vigilância, como CFTV e patrulhamento humano, mostram-se inviáveis do ponto de vista de tempo de estabelecimento, estrutura disponível, custo e logística operacional. A solução proposta pode trazer vantagens relevantes nesses aspectos, além de reduzir o risco a vidas humanas por demandar somente um operador agindo à distância.

No que diz respeito à ARP, os resultados obtidos experimentalmente demonstram a viabilidade do protótipo construído em atender às premissas estabelecidas, mostrando-se uma plataforma de baixo custo, portátil e de fácil operação e manutenção, desse modo, útil para diversas aplicações. Quanto ao software desenvolvido, os resultados de desempenho demonstram que a complexidade algorítmica admite execução em tempo real de forma totalmente embarcada na ARP, dispensando a necessidade de transmissão das imagens para processamento externo.

Uma limitação importante de se mencionar é o tempo de voo reduzido das ARPs portáteis do tipo quadricóptero, o que limitaria o tempo útil do sistema de detecção de ETMs e, conseqüentemente, a viabilidade de uso. Algumas soluções podem ser usadas para reduzir essa limitação. Uma possibilidade é o uso de múltiplas ARPs com substituição automática da ARP estacionária. Outra possibilidade é o uso de alimentação cabeada a partir do solo, como ocorre em [45]. Uma sugestão para trabalho futuro é a criação de um algoritmo para determinar o nível de ameaça de uma ETM de acordo com suas características físicas e comportamentais, como seu tamanho, velocidade, padrão de movimentação e direção.

Agradecimentos

Este estudo foi parcialmente financiado pela Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) – Código Financeiro 001.

Referências

- [1] HAERING, N.; VENETIANER, P. L.; LIPTON, A. The evolution of video surveillance: an overview. *Machine Vision and Applications*, v. 19, n. 5-6, p. 279–290, 2008.
- [2] NAYAR, S. K.; NARASIMHAN, S. G. Vision in bad weather. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*. Piscataway: IEEE, 1999. p. 820–827.
- [3] WANG, H.; LI, S. Z.; WANG, Y. Face recognition under varying lighting conditions using self quotient image. In *Proceedings of the Sixth IEEE Int. Conference on Automatic Face and Gesture Recognition*. Piscataway: IEEE, 2004. p. 819–824.
- [4] HAUTIÈRE, N.; TAREL, J. P.; AUBERT, D. Towards fog-free in-vehicle vision systems through contrast restoration. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Piscataway: IEEE, 2007. p. 1–8.
- [5] SEMSCH, E.; JAKOB, E. S. M.; PAVLÍČEK, D.; PECHOUCHEK, M. Autonomous UAV surveillance in complex urban environments. In *IEEE Int. Joint Conference on Web Intelligence and Intelligent Agent Technology*. Piscataway: IEEE, 2009. p. 82–85.
- [6] NAIDOO, Y.; STOPFORTH, R.; BRIGHT, G. Development of an UAV for search & rescue applications. In *IEEE AFRICON 11'*. Piscataway: IEEE, 2011. p. 1–6.
- [7] GOTOVAC, D.; GOTOVAC, S.; PAPIĆ, V. Mapping aerial images from UAV. In *IEEE International Multidisciplinary Conference on Computer and Energy Science (SpliTech)*. Piscataway: IEEE, 2016. p. 1–6.
- [8] WU, C.; QI, J.; SONG, D.; QI, X.; LIN, T.; HAN, J. Development of an unmanned helicopter automatic barrels transportation system. In *IEEE International Conference on Robotics and Automation (ICRA)*. Piscataway: IEEE, 2015. p. 4686–4691.
- [9] MERWADAY, A.; GUVENC, I. UAV assisted heterogeneous networks for public safety communications. In *IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*. Piscataway: IEEE, 2015. p. 329–334, 2015.
- [10] GRIPPA, P. Decision making in a UAV-based delivery system with impatient customers. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Piscataway: IEEE, 2016. p. 5034–5039.
- [11] BEST DRONE FOR THE JOB. *Pro Photography*. [2017]. Acesso em: 28 ago. 2021. Disponível em: <http://bestdroneforthejob.com/product-category/work-drones/pro-photography-drones/>.
- [12] DRONE RACING LEAGUE. *The Drone Racing League*. [2016]. 28 ago. de 2021. Disponível em: <https://thedroneracingleague.com/>.
- [13] GEORGE, E. A.; TIWARI, G.; YADAV, R. N.; PETERS, E.; SADANA, S. UAV systems for parameter identification in agriculture. In *IEEE Global Humanitarian Technology Conference*. Piscataway: IEEE, 2013. p. 270–273.
- [14] SIEGWART, R.; NOURBAKHSH, I. R.; SCARAMUZZA, D. *Introduction to Autonomous Mobile Robots*. 2. ed. Cambridge: MIT Press, 2011. 472 p.
- [15] SABATELLI, S.; SECHI, F.; FANUCCI, F.; ROCCHI, A. A sensor fusion algorithm for an integrated angular position estimation with inertial measurement units. In *Design, Automation & Test In Europe Conference & Exhibition (DATE)*. Piscataway: IEEE, 2011. p. 1-4.
- [16] FRIEDRICHS, S.; HEMMER, M.; KING, J.; SCHMIDT, C. The continuous 1.5 D terrain guarding problem: Discretization, optimal solutions, and PTAS. *Journal of Computational Geometry*, v. 7, n. 1, p. 256–284, 2016.
- [17] SZELISKI, R. *Computer vision: algorithms and applications*. Washington, DC: Springer, 2010.
- [18] AJMERA, J.; SIDDHARTHAN, P. R.; RAMARAVIND, K. M.; VASAN, G.; BALAJI, N.; SANKARANARAYANAN, V. Autonomous visual tracking and landing of a quadrotor on a moving platform. In *Third Int. Conference on Image Information Processing (ICIIP)*. Piscataway: IEEE, 2015. p. 342–347.
- [19] KAKVAND, P.; JABERZADEH, M.; INALLOU, M. M.; ALBORZ, Y. Smart onboard UAV system: using computer vision system to find a movable and stationary target. In *2nd International Conference on Knowledge-Based Engineering and Innovation (KBEI)*. Piscataway: IEEE, 2015. p. 694–699.
- [20] ZHANG, C. H.; CHEN, J. B.; SONG, C. L.; XU, J. H. An UAV navigation aided with computer vision. In *26th Chinese Control and Decision Conference (CCDC)*. Piscataway: IEEE, 2014. p. 5297–5301.
- [21] GUPTE, S.; MOHANDAS, P. I. T.; CONRAD, J. M. A Survey of Quadrotor Unmanned Aerial Vehicles. In *Proceedings of IEEE Southeastcon*. Piscataway: IEEE, 2012. p. 1–6.

- [22] ARAGUÁS, G.; PAZ, C.; PAINA, G. P.; CANALI, L. Visual homography-based pose estimation of a quadrotor using spectral features. In Latin America Congress on Computational Intelligence (LA-CCI). Piscataway: IEEE, 2015. p. 1–6.
- [23] FOWERS, S.; LEE, D. J.; TIPPETTS, B. J.; LILLYWHITE, K. D.; DENNIS, A. W.; ARCHIBALD, J. K. Vision aided stabilization and the development of a quad-rotor micro UAV. In International Symposium on Computational Intelligence in Robotics and Automation. Piscataway: IEEE, 2007. p. 143–148.
- [24] TANG, D.; LI, F.; SHEN, N.; GUO, S. UAV attitude and position estimation for vision-based landing. In Proceedings of 2011 International Conference on Electronic and Mechanical Engineering and Information Technology. Piscataway: IEEE, 2011. p. 4446–4450.
- [25] JIN, Z.; WANG, X.; PAN, Q.; MORAN, B. Optimal UAV localisation in vision based navigation systems. In 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). Piscataway: IEEE, 2016. p. 1021–1025.
- [26] SUKANYA, C. M.; GOKUL, R.; PAUL, V. A survey on object recognition methods. International Journal of Science, Engineering and Computer Technology, v. 6, n. 1, p. 48–52, 2016.
- [27] POUNDS, P.; MAHONY, R.; CORKE, P. Modelling and control of a quad-rotor robot. In Proceedings of the 2006 Australasian Conference on Robotics and Automation. Sydney: ARAA, 2006.
- [28] NAVAJAS, G. H. T.; PRADA, S. R. Building your own quadrotor: A mechatronics system design case study. In III International Congress of Engineering Mechatronics and Automation (CIIMA). Piscataway: IEEE, 2014. p. 1–5.
- [29] TARIQ, A.; OSAMA, S. M.; GILLANI, A. Development of a low cost and light weight UAV for photogrammetry and precision land mapping using aerial imagery. In International Conference on Frontiers of Information Technology (FIT). Piscataway: IEEE, 2016. p. 360–364.
- [30] SILVA, L. C. B.; BERNARDO, R. M.; OLIVEIRA, H. A.; ROSA, P. F. F. Multi-UAV agent-based coordination for persistent surveillance with dynamic priorities. In International Conference on Military Technologies (ICMT). Piscataway: IEEE, 2017. p. 765–771.
- [31] CHO, J.; SUNG, J.; YOON, J.; LEE, H. Towards Persistent Surveillance and Reconnaissance Using a Connected Swarm of Multiple UAVs. IEEE Access, v. 8, p. 157906–157917, 2020.
- [32] WU, Y.; WU, S.; HU, X. Cooperative Path Planning of UAVs & UGVs for a Persistent Surveillance Task in Urban Environments. In IEEE Internet of Things Journal, v. 8, n. 6, p. 4906–4919, 2021.
- [33] BANDARUPALLI, A.; SWARUP, D.; WESTON, N.; CHATERJI, S. Persistent Airborne Surveillance using Semi-Autonomous Drone swarms. In Proceedings of the 7th Workshop on Micro Aerial Vehicle Networks, Systems, and Applications. New York: ACM, 2021. p. 19–24.
- [34] SHARMA, M. K.; SINGAL, G.; GUPTA, S. K.; CHANDRANEIL, B.; AGARWAL, S.; GARG, D.; MUKHOPADHYAY, D. INTERVENOR: Intelligent Border Surveillance using Sensors and Drones. In 6th International Conference for Convergence in Technology (I2CT). Piscataway: IEEE, 2021. p. 1–7.
- [35] ABDEL-MALEK, M. A.; IBRAHIM, A. S.; MOKHTAR, M. Optimum UAV positioning for better coverage-connectivity tradeoff. In IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC). Piscataway: IEEE, 2017. p. 1–5.
- [36] SUN, J.; MASOUIROS, C. Drone Positioning for User Coverage Maximization. In IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC). Piscataway: IEEE, 2018. p. 318–322.
- [37] NASRAOUI, L.; ROY, S. Optimal UAV Positioning for Terrestrial Users. In 91st Vehicular Technology Conference (VTC2020-Spring). Piscataway: IEEE, 2020. p. 1–5.
- [38] NAM, L. H.; HUANG, L.; LI, X. J.; XU, J. F. An approach for coverage path planning for UAVs. In IEEE 14th International Workshop on Advanced Motion Control (AMC). Piscataway: IEEE, 2016. p. 411–416.
- [39] BERNARDO, R. M.; SILVA, L. C. B.; ROSA, P. F. F. Onboard Video Stabilization for Low Cost Small RPAS Surveillance Applications. In 7th Brazilian Conference on Intelligent Systems (BRACIS). Piscataway: IEEE, 2018. p. 450–455.
- [40] BERNARDO, R. M.; SILVA, L. C. B.; ROSA, P. F. F. Concepção de Uma Plataforma de VANT de Baixo Custo para Uso em Pesquisas. In V Workshop de Comunicação em Sistemas Embarcados Críticos (WoCCES). Rio de Janeiro: SBC, 2017.
- [41] BERNARDO, R. M. The DroneAPI Python Library. Disponível em: <https://github.com/marouquio/DroneAPI>. Acesso em: 20 jan. 2021.

- [42] DRONEKIT. Developer Tools for Drones. [2015]. Acesso em: 20 out. 2021. Disponível em: <https://dronekit.io/>.
- [43] MAVLINK. MAVLink Developer Guide. [2017]. Acesso em: 20 out. 2021. Disponível em: <https://mavlink.io/en/>.
- [44] ARDUPILOT. Open-Source Autopilot Software System. [2020]. Acesso em: 20 out. 2021. Disponível em: <https://ardupilot.org/>.
- [45] PAPACHRISTOS, C.; TZES, A. The power-tethered UAV-UGV team: A collaborative strategy for navigation in partially-mapped environments. In 22nd Mediterranean Conference on Control and Automation. Piscataway: IEEE, 2014. p. 1153-1158.