

CONCEPÇÃO DE UM SARP DE BAIXO CUSTO COM ESTABILIZAÇÃO DE VÍDEO EM TEMPO REAL PARA USO EM MONITORAMENTO ESTACIONÁRIO DO SOLO

RICARDO MAROQUIO BERNARDO*, LUIS C. BATISTA DA SILVA†, PAULO F. FERREIRA ROSA‡

**Instituto Federal de Educação, Ciência e Tecnologia do Espírito Santo (IFES)
Cachoeiro de Itapemirim, ES, Brasil*

†*Centro Federal de Educação Tecnológica Celso Suckow Fonseca (CEFET)
Nova Friburgo, RJ, Brasil*

‡*Instituto Militar de Engenharia (IME) - Lab. de Robótica e Inteligência Computacional
Rio de Janeiro, RJ, Brasil*

Emails: ricardo.maroquio@ifes.edu.br, luis.silva@cefet-rj.br, rpaulo@ime.eb.br

Abstract— Visual sensing of remote regions is a useful task for several applications and can be performed by a remotely piloted aircraft system (RPAS) with an embedded camera. However, a ready-to-use RPAS has a high acquisition and maintenance cost, as well as low flexibility for modifications. In addition, capture of images from cameras on remotely piloted aircraft (RPA) suffers from stabilization problems, mainly due to vibrations caused by the propulsion system and movements caused by environmental disturbances and embedded sensors errors. This paper presents a low-cost RPA built specifically to provide stabilized video capturing, using novel image stabilizer system based on hardware and software. Real experiments were conducted and the developed solution proved capable of delivering 31 fps at 480p resolution and 15 fps at 720p using only embedded aircraft resources.

Keywords— RPAS, remote sensing, aerial imagery stabilization.

Resumo— O sensoriamento visual de regiões remotas é uma tarefa útil a diversas aplicações e é passível de ser executado por um SARP (Sistema de Aeronaves Remotamente Pilotadas) com câmera embarcada. Contudo, um SARP pronto para uso possui custo elevado de aquisição e manutenção, bem como baixa flexibilidade para modificações. Além disso, aquisição de imagens a partir de câmeras embarcadas em aeronaves remotamente pilotadas (ARPs) sofrem com problemas de estabilização, devido, principalmente, a vibrações causadas pelo sistema de propulsão e a movimentos causados por perturbações do ambiente e erros dos sensores embarcados. Este artigo apresenta uma ARP de baixo custo construída especificamente para proporcionar uma captura de vídeo estabilizada, usando um sistema inovador de estabilização de imagens baseado em hardware e software. Experimentos reais foram conduzidos e a solução desenvolvida se mostrou capaz de fornecer 31 fps na resolução de 480p e 15 fps em 720p usando somente recursos embarcados na aeronave.

Palavras-chave— SARP, sensoriamento remoto, estabilização de imagens aéreas.

1 Introdução

Os sistemas de aeronaves remotamente pilotadas (SARPs) abrangem aeronaves de diferentes dimensões e tipos, incluindo veículos de asa fixa (p.ex., aviões), de asa rotativa (p.ex., helicópteros e multi-rotor) e veículos mais leves do que o ar (p.ex., dirigíveis). Tais veículos podem ter diferentes graus de autonomia, ou seja, podem realizar pequenas tarefas ou missões inteiras sem o auxílio de um piloto humano. Algumas dessas tarefas incluem decolagem, pouso, seguimento de rotas georeferenciadas, desvio de obstáculos, aquisição de imagens, detecção de objetos em movimento e perseguição de alvos. O grau de autonomia depende dos recursos computacionais e sensoriais disponíveis e o processamento da tarefa pode ser realizado de forma embarcada ou remota. Na forma embarcada, o processamento ocorre em algum dispositivo computacional presente no próprio veículo. Na forma remota, os dados a serem processados são transmitidos para um dispositivo computacional externo, geralmente presente em uma estação de controle em solo (ECS). Nesse último caso, deve haver uma garantia mínima de comunica-

ção entre a ARP e a ECS. As duas abordagens de processamento possuem vantagens e desvantagens. No processamento embarcado, os dados sensorizados tornam-se mais rapidamente disponíveis, podendo viabilizar a geração de resultados em menos tempo. O processamento remoto, por outro lado, pode lidar com dados computacionalmente mais complexos, pois permite o uso de dispositivos computacionais de melhor desempenho, inviáveis de serem embarcados em um SARP de pequeno porte. Portanto, os requisitos de computação e de tempo de resposta são fatores que possuem influência direta na decisão pelo processamento embarcado ou remoto.

As ARPs do tipo quadricóptero, também tratadas na literatura como veículos aéreos não tripulados (VANTs) e atualmente popularizados como drones, são veículos do tipo multi-rotor e possuem características intrínsecas interessantes, incluindo capacidade de decolar e pousar na vertical e de pairar. A disponibilidade desse tipo de veículo aumentou consideravelmente nos últimos anos. Tanto veículos prontos para uso quanto os componentes necessários para se construir tais veí-

2 Revisão de Literatura

culos estão cada vez mais precisos, menores e mais baratos, viabilizando seu uso em uma vasta possibilidade de cenários. Soluções para vigilância de grandes áreas (como fronteiras, instalações carcerárias e campos minados), inspeção de instalações elevadas e de difícil acesso (como usinas eólicas, torres de energia e prédios) são alguns exemplos de cenários que podem fazer uso de veículos multi-rotadores para alguma aplicação.

Quando a aplicação em questão demanda monitoramento visual ininterrupto de uma área predefinida visando detecção de objetos móveis, a precisão do resultado depende diretamente da qualidade da imagem capturada, tornando fundamental o uso de técnicas de estabilização de imagens. Câmeras embarcadas em multi-rotadores sofrem com vibrações causadas pelos motores e com deslocamentos provocados por erros de sensores e por intempéries do ambiente. As vibrações provocam o indesejado “efeito gelatina”, em que a imagem parece ondular. Ainda, quando se deseja pairar sobre uma posição georreferenciada fixa, os erros dos sensores usados pelo sistema de navegação (incluindo sensor de posicionamento global por satélite, acelerômetro, giroscópio, magnetômetro, bússola, barômetro e sonar) provocam pequenos deslocamentos no veículo. Por fim, ventos e quedas de pressão atmosférica também podem fazer com que o veículo sofra pequenos deslocamentos.

Diante do exposto, o presente trabalho visa a construção de uma ARP multi-rotor de baixo custo do tipo quadricóptero capaz de capturar imagens estáveis planares do solo a partir de um voo pairado estacionário, com o intuito de ser usado na detecção de objetos independentemente móveis (do inglês *independently moving objects*, ou IMO) presentes no solo. Para alcançar esse propósito, consideram-se tanto soluções em hardware quanto em software, no intuito de fornecer uma solução integrada capaz de ser executada em tempo real usando somente recursos embarcados. Em aplicações para detecção de IMOs, o alinhamento entre os quadros do vídeo capturado é fundamental para a precisão dos resultados e para se determinar o tamanho mínimo dos IMOs passíveis de detecção. Restrições de tempo real impõem um desafio adicional ao problema, principalmente quando o poder computacional disponível restringe-se àquele dos dispositivos embarcados que, em uma ARP, são pequenos e leves.

O restante do artigo está organizado como segue. A seção 2 apresenta uma breve revisão dos trabalhos relacionados. Na seção 3, são apresentadas as especificações e características dinâmicas da ARP construída. A seção 4 apresenta detalhes do sistema de estabilização. Na seção 5, são apresentados resultados de experimentos simulados e reais e, por fim, na seção 6, estão as conclusões obtidas com o trabalho.

A literatura apresenta diversos trabalhos que fazem uso de ARPs construídas pelos próprios autores, sendo alguns equipados com câmeras, visando a realização de tarefas que requerem sensoriamento visual. Em alguns destes trabalhos, a própria construção da ARP é o foco de estudo, como ocorre nos trabalhos de (Devaud et al., 2012) e de (Davis et al., 2013). No primeiro, os autores desenvolveram um quadricóptero ao custo de 450 euros capaz de voar 15 minutos com uma bateria LiPo de 4500 mAh, com envergadura de 610mm e peso de 1,4kg. No segundo, os autores apresentam o processo de desenvolvimento de um quadricóptero para uso em experimentos com enxames, ao custo de 500 dólares, pesando 500g e capaz de voar por 10 minutos. No trabalho de (Montanari et al., 2015), os autores apresentam um sistema de detecção de veículos com uso de uma ARP construída pelos autores, equipada com câmera e visando a tarefa de vigilância. Resultados experimentais demonstram que o sistema proposto pode ser usado para a tarefa de rastreamento de veículos, porém, sob condições controladas e visando processamento *offline*. No recente trabalho de (Cerón et al., 2018), os autores apresentam um sistema de navegação embarcado baseado em visão para seguimento de linhas de força em redes elétricas usando uma ARP. O sistema foi testado em uma tarefa real de inspeção usando uma ARP com uma câmera apontada verticalmente para baixo. Apesar de cumprirem o que foi proposto, percebe-se claramente o “efeito gelatina” no vídeo capturado pela câmera embarcada, vídeo este disponível no YouTube em <https://youtu.be/pa4fqoGJmDU>, conforme informado no artigo. Isso demonstra que, para algumas tarefas, a estabilização é dispensável.

Os trabalhos descritos em (Ibrahim et al., 2010), (Aguilar and Angulo, 2014), (Aguilar et al., 2017) e (Mingkhwan and Khawsuk, 2017) abordam especificamente a estabilização digital de imagens capturadas por câmeras embarcadas em ARPs e usam diferentes técnicas e contextos para atacar o problema. Em (Ibrahim et al., 2010), os autores propõem um framework para detectar e rastrear IMOs em imagens capturadas em grandes altitudes, onde objetos móveis são pequenos e ocupam apenas alguns pixels dos quadros processados. O primeiro módulo do framework é a estabilização digital de imagem, que computa a correspondência de *keypoints* entre os quadros, calcula a transformação geométrica necessária para o alinhamento e monta um mosaico com as imagens processadas, considerando que o veículo está se movendo ao longo de um caminho. Como o processo de alinhamento leva em consideração a rotação e a translação da câmera nos 3 eixos, uma transformação projetiva (homografia) é requerida

e o método RANSAC (Stewart, 1999) é usado. Os autores usaram diferentes métodos de detecção de *keypoints* e o melhor tempo obtido foi de 38ms/quadro para a resolução de 320x240 pixels, considerando um computador padrão da época.

O trabalho apresentado em (Aguilar and Angulo, 2014) considera o uso do algoritmo RANSAC combinado com uma função para minimizar a diferença de intensidades de cinza nos pixels dos quadros e para estimar os parâmetros de movimento entre os quadros. Um filtro passa-baixa é usado para suavizar a estimação do movimento como um todo, uma vez que o método proposto acumula as transformações estimadas para aplicação em lote *a posteriori*. Para a detecção e descrição de *keypoints*, os autores usaram o algoritmo SURF (Bay et al., 2006). Os experimentos conduzidos foram executados em um notebook conectado via wi-fi a uma ARP de baixo custo. Os resultados apresentados são apenas visuais e mostram quadros estabilizados, bem como gráficos com a suavização das estimativas de translação e rotação.

No trabalho de (Aguilar et al., 2017), os autores desenvolveram um sistema de estabilização de baixo custo que reduz efeitos de movimentos indesejados de uma câmera estabilizada mecanicamente por um gimbal de 3 eixos fixado em um pequeno quadricóptero. Um minicomputador ODROID XU4 intercepta os comandos de rotação no eixo de guinada (yaw) enviados ao drone e, usando um filtro de Kalman, estima a intenção do movimento. Para a detecção de *keypoints*, os autores usaram o algoritmo SURF e, para descrição, usaram o BRIEF (Calonder et al., 2012). A combinação de estimação de movimento pela interceptação de comandos com o rastreamento de *keypoints* ao longo dos quadros, movimentos indesejáveis foram removidos. Resultados quantitativos foram apresentados usando-se três diferentes métricas medidas a partir do quadro original e da sequência estabilizada de quadros, a saber, RMSE (root-mean-square error), PSNR (peak signal-to-noise ratio) e ITF (inter-frame fidelity), cujos percentuais de melhoria sobre o vídeo original foram, respectivamente, 9,7%, 16,7% e 20,6%.

O trabalho descrito em (Mingkhwan and Khawsuk, 2017) apresenta um algoritmo de estabilização de imagens para ARPs de pequeno porte incapazes de transportar um gimbal. Os autores apontaram que um dos propósitos da solução é de melhorar o conforto visual humano, reduzindo o mal estar e náusea provocados por movimentos nos quadros. Alguns algoritmos de detecção de *keypoints* foram analisados e o algoritmo SURF foi escolhido. O processo de estabilização segue o padrão de estimar o movimento de um quadro de referência e do quadro atual aplicando uma transformação geométrica que compensa o movimento computado. Experimentos foram conduzidos em 4 vídeos distintos usando a métrica PSNR e o ví-

deo estabilizado teve uma melhora de aproximadamente 4,5% quando comparado ao vídeo original.

Em suma, a literatura relacionada apresenta trabalhos que focam na construção de ARPs e outros que focam na estabilização digital de imagens oriundas de ARPs. Neste último caso, basicamente, existe uma variação nos algoritmos de detecção, descrição e rastreamento dos *keypoints*. Alguns trabalhos usam técnicas de otimização e redução de dimensionalidade para acelerar o processo. Como já mencionado, o presente trabalho foca em uma combinação de hardware e software para atacar o problema e apresenta técnicas inovadoras para tal.

3 A ARP Construída

Esta seção apresenta detalhes técnicos sobre a ARP construída, bem como questões referentes à estabilidade do veículo. A figura 1 apresenta um projeto 3D de alta fidelidade do quadricóptero construído, ocultando a fiação e suas conexões para melhor visualização dos demais componentes. A aeronave foi construída com estrutura em fibra de carbono e conexões estruturais impressas em PLA, com peso total de 1263g sem bateria. Com uma bateria LiPo de 6 células de 8000mAh o peso aumenta para 2145g e, nessa configuração, a autonomia segura de voo (mantendo 10% de bateria restante) é de 53 minutos. Cada propulsor é capaz de promover um impulso nominal máximo de 1975g com hélices de fibra de carbono 14x4.8. A controladora de voo utilizada foi uma Navio2, da Emlid, acoplada a um minicomputador Raspberry Pi 3 Model B. Em experimentos práticos, a aeronave usou somente 34% de sua potência máxima para pairar, deixando um lastro considerável para se embarcar outros componentes que porventura venham a ser necessários. O custo total de construção da ARP foi de aproximadamente 820 dólares, desconsiderando impostos. Esse valor está bem abaixo do custo de veículos multi-rotorés similares comercialmente disponíveis, mantendo a vantagem de se ter total flexibilidade para modificações. Por questão de segurança, mesmo que a ARP realize tarefas de forma autônoma controladas por software embarcado ou não, é necessário que se tenha a possibilidade de um controlador humano intervir em caso de pane. Para isso, usa-se um rádio-transmissor capaz de assumir o controle da ARP, com preços partindo de 50 dólares (sem impostos) e variando conforme os recursos técnicos disponíveis (número de canais, alcance, duração de bateria etc.).

A figura 2 apresenta a dinâmica básica de funcionamento de um quadricóptero qualquer em formato X. Um fato importante a observar é que os movimentos angulares de arfagem, rolagem e guinada dependem do diferencial de rotação simultâneo entre os motores. A velocidade de execução



Figura 1: SARP de pequeno porte do tipo quadricóptero construído visando estabilidade, tempo de voo e baixo custo.

esses movimentos deve ser ajustada calibrando-se os valores de PID (proporcional integral derivativo) da controladora de voo para cada um dos eixos, objetivando evitar movimentos bruscos que possam comprometer a estabilidade da câmera. A controladora de voo utilizada na ARP construída usa o firmware de código aberto ArduCopter 3.5, que possui uma função automática para calibração dos PIDs, executando um voo semi-autônomo de aproximadamente 10 minutos para a realização dessa tarefa. Além de suavizar os movimentos angulares do veículo, experimentos práticos mostraram que a calibração dos PIDs reduz consideravelmente a vibração do chassi, evitando ruídos nos sensores embarcados. No link <http://www.marquio.com/cba2018.html> pode-se ver um vídeo com a calibração automática de PIDs e testes de voo da ARP construída. O firmware ArduCopter 3.5 proporciona à ARP a realização de diversas tarefas autônomas, incluindo decolagem, pouso, seguimento de trajetórias georreferenciadas, retorno automático ao ponto de decolagem em caso de pane (por comando deliberado, perda de sinal ou pouca bateria), voo circular em torno de uma posição predefinida e pairagem estacionária com manutenção de posição.

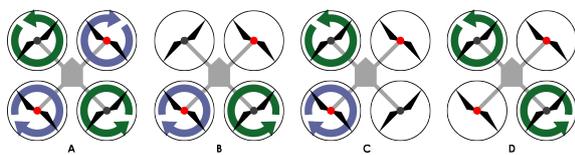


Figura 2: Dinâmica de voo básica de um quadricóptero em relação ao acionamento dos propulsores. A) subida; B) arfagem à frente; C) rolagem à direita; D) guinada à esquerda.

Em comparação com as ARPs dos artigos apresentados na seção anterior, esta apresenta vantagens em relação a tempo de voo, flexibilidade de modificação (inclusive do firmware), capacidade de carga e flexibilidade de comunicação (usando protocolo MavLink), justificando sua construção e uso em diversas aplicações.

4 O Sistema de Estabilização

O sistema de estabilização divide-se em duas partes: hardware e software. A parte de hardware compreende a estrutura física da ARP e o suporte de fixação da câmera ao drone. O software diz respeito aos algoritmos utilizados para a estabilização digital de imagens. As subseções a seguir abordam hardware e software separadamente.

4.1 Hardware

Como já mencionado, a suavização dos movimentos angulares através da calibração dos PIDs é fundamental para a melhoria da estabilidade e redução de vibrações. Questões básicas de construção, como eliminação de folgas nas fixações dos componentes estruturais, também devem ser cheçadas. Outro ponto importante a ser verificado é o balanceamento individual das hélices. Isso pode ser aferido fixando-se a hélice em um eixo livre paralelo ao solo. Se houver diferença de peso entre as pás, a hélice vai girar para um dos lados, indicando desequilíbrio. Para solucionar esse problema, pode-se lixar a pá de maior peso ou pode-se colar uma camada fina de fita adesiva na pá de menor peso, até que o desequilíbrio desapareça. Hélices desbalanceadas são grandes causadoras de vibrações na estrutura da aeronave.

A fixação da câmera na estrutura da ARP é outro ponto crítico, principalmente pela dificuldade prática de se eliminar completamente os efeitos da vibração causada pelos motores. Para superar isso, a fixação da câmera deve ser feita com algum tipo de sistema de amortecimento capaz de absorver as vibrações remanescentes. Um dos sistemas de amortecimento mais utilizados são as conexões flexíveis de borracha, silicone ou material semelhante. A rigidez da conexão flexível deve ser ajustada de acordo com o peso da câmera. Como o foco deste trabalho é a captura de uma imagem planar do solo, um mecanismo que realiza a compensação dos movimentos angulares da ARP é utilizado. Tal mecanismo é conhecido como *gimbal* e seu princípio de funcionamento está bem descrito no trabalho de (Jakobsen and Johnson, 2005). Um bom exemplo de uso de gimbal em uma ARP pode ser visto no trabalho de (Quigley et al., 2005). A figura 3 mostra o gimbal utilizado nos experimentos conduzidos neste trabalho, bem como o mecanismo de fixação que, neste caso, usa 4 conectores de borracha para absorção de vibrações.

O gimbal de 3 eixos permite compensar movimentos angulares de rolagem e arfagem. Mesmo em uma posição estacionária, a ARP pode realizar esses movimentos na presença de vento, visando a manutenção da posição. Movimentos angulares de guinada provocados por vibrações remanescentes também são consideravelmente reduzidos pelo gimbal. No caso da ARP do tipo quadricóptero,



Figura 3: Gimbal de 3 eixos usado nos experimentos conduzidos.

as vibrações no eixo de guinada são mais perceptíveis nas extremidades dos braços. Por conta disso, quanto mais centralizado estiver o gimbal, menor será o efeito causado por esse tipo de vibração.

4.2 Software

Apesar de garantir a planaridade e a estabilidade angular da imagem, os mecanismos apresentados até aqui não são capazes de lidar com problemas de deslocamentos lineares horizontais e verticais do veículo, geralmente causado por erros e/ou limitações de sensores. Sensores de sistemas de posicionamento global por satélite, como o NEO-M8N, usado na ARP aqui construída, possuem precisão horizontal de aproximadamente 2m, conforme especificações do fabricante. Essa precisão acaba ocasionando um movimento de câmera que pode inviabilizar o uso direto de técnicas de detecção de IMOs em solo. Ainda, para aferição da altitude, usa-se um barômetro com precisão nominal de 10cm, segundo o fabricante. Porém, variações de temperatura e pressão podem interferir no sensor e provocar deslocamentos verticais na aeronave. Esses tipos de movimentos podem ser compensados usando-se um estabilizador digital de imagens, que, em suma, consiste em um algoritmo de pós-processamento de imagens capaz de alinhar os quadros realizando a correspondência de pontos unicamente notáveis e naturalmente presentes nas imagens capturadas. Esta seção apresenta detalhes do algoritmo desenvolvido para lidar com os pequenos deslocamentos horizontais e verticais da câmera.

Para realizar o alinhamento dos quadros, é necessário um quadro de referência que, neste caso, é capturado assim que o veículo atinge a posição estacionária e seu estado angular indica ausência de intempéries ambientais. Dessa forma, a transformação geométrica entre qualquer quadro e o quadro de referência é dado pela equação 1, como em (Morimoto and Chellappa, 1996):

$$\begin{pmatrix} X_i \\ Y_i \end{pmatrix} = S \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} X_0 \\ Y_0 \end{pmatrix} + \begin{pmatrix} t_X \\ t_Y \end{pmatrix} \quad (1)$$

Na equação 1, (X_i, Y_i) é a posição do quadro i , para $i = \{1, n\}$, onde n é o número total de quadros; S é o fator de escala devido aos movimentos

verticais; θ é o ângulo de rotação da câmera no eixo ótico; (X_0, Y_0) é a posição do quadro de referência; e (t_X, t_Y) são as translações horizontais paralelas ao solo. Portanto, os quatro parâmetros a serem estimados são S, θ, t_X e t_Y .

A qualidade de estabilização pode ser medida aferindo-se o quão similar são os quadros capturados. Neste sentido, a estabilização ótima pode ser representada por uma função que maximize a similaridade entre os quadros da sequência, usando-se alguma função de medida de similaridade entre quadros, de acordo com a equação 2 a seguir:

$$S_{opt} = arg \max_{(S, \theta, t_X, t_Y)} \frac{1}{n} \sum_{i=1}^{n-1} |Sim(F_i, F_0)| \quad (2)$$

Na equação 2, n é o total de quadros, F_i é o quadro estabilizado i , F_0 é o quadro de referência e Sim é a função que computa a similaridade entre dois quadros. Resumidamente, a função da equação 2 é a função objetivo da estabilização digital de imagens proposta.

Dois funções de medida de similaridade presentes na literatura são a PSNR, já mencionada, e a SSIM (structural similarity). O trabalho de (Hore and Ziou, 2010) apresenta detalhes e comparações entre as duas funções e conclui que a SSIM é mais parecida com a forma com que os humanos medem a similaridade de imagens. Ambos serão usados na próxima seção.

O método utilizado para computar as correspondências de *keypoints* entre dois quadros distintos subsequentes é baseado no algoritmo de Lucas-Kanade descrito em (Bouguet, 2001). Como entrada, o algoritmo requer um conjunto de *keypoints* P_1 previamente extraído de uma imagem I_1 , bem como uma imagem I_2 onde se deseja encontrar as correspondências de P_1 , resultando em um novo conjunto de *keypoints* com as novas posições de P_1 em I_2 , chamado P_2 . Para cada *keypoint* em P_1 , o algoritmo define uma janela quadrada centralizada no *keypoint* e procura por uma janela similar em I_2 . O tamanho dessa janela varia de acordo com o tamanho máximo do deslocamento de *keypoints* entre dois quadros. Para acelerar o processo de correspondência, para cada quadro, o algoritmo usa uma pirâmide de imagens de diferentes resoluções, geradas a partir do quadro analisado. Então, o *keypoint* é primeiramente localizado na imagem de menor resolução e, a cada iteração, a próxima imagem de maior resolução é utilizada, até alcançar a imagem original, onde terá a precisão máxima. Tanto o tamanho da janela quanto o número de níveis da pirâmide influenciam diretamente no desempenho do algoritmo. Um fluxograma da solução com fluxo ótico piramidal é mostrado na figura 4. Na figura 4, a conversão dos quadros para escala de cinza promove uma redução de dimensionalidade sem prejudicar o desempenho do método. Inicialmente, o

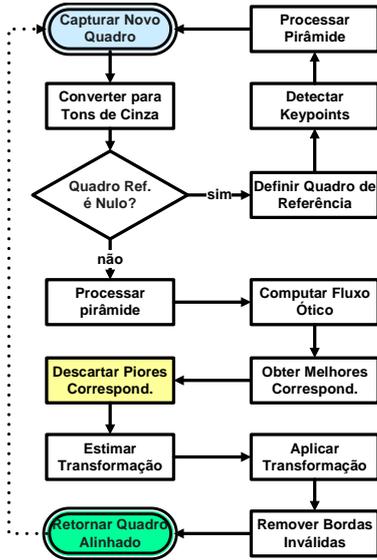


Figura 4: Diagrama da metodologia de estabilização usando fluxo óptico piramidal.

quadro de referência é definido. Os *keypoints* do quadro de referência, bem como sua pirâmide de imagens, são gerados somente uma vez e reusados pelo processo de estabilização a cada novo quadro. Os novos quadros que chegarem terão suas correspondências estimadas pelo algoritmo de fluxo ótico. Então, somente as correspondências cujo erro esteja abaixo do erro médio serão usadas. Por outro lado, no passo com fundo amarelo da figura 4, se um *keypoint* não tem uma correspondência ou se seu erro de correspondência estiver acima da média geral, seu valor de “descarte” é incrementado e, ao longo do tempo, isso servirá de indicador para o quão ruim este *keypoint* é para ser rastreado. Quando esse valor estiver acima de um limiar, o *keypoint* é definitivamente removido da lista de rastreamento. Esse processo ocorre continuamente até que a quantidade de *keypoints* rastreados seja um pouco maior do que o mínimo requerido para se estimar a transformação geométrica. Com os p pares correspondentes dos *keypoints* selecionados, com base na equação 1 e no método descrito em (Zhu et al., 2015), obtém-se $2 * p$ equações lineares que podem ser arranjadas na equação matricial para se obter a estimativa de movimento, como segue:

$$\begin{pmatrix} x'_1 & y'_1 \\ x'_2 & y'_2 \\ \vdots & \vdots \\ x'_p & y'_p \end{pmatrix}_B = \begin{pmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ \vdots & \vdots & \vdots \\ x_p & y_p & 1 \end{pmatrix}_A \times \begin{pmatrix} m_0 & m_3 \\ m_1 & m_4 \\ m_2 & m_5 \end{pmatrix} \quad (3)$$

M_{afim}

Na equação 3, os *keypoints* x' e y' são os *keypoints* correspondentes do quadro de referência, x e y , considerando o movimento do quadro. Os parâmetros m_0, m_1, m_3 e m_4 são os produtos dos parâmetros de rotação e escala, e m_2 e m_5

são os parâmetros de translação. Os parâmetros podem então ser resolvidos usando-se $M_{\text{afim}} = (A^T A)^{-1} A^T B$. Depois de se estimar os parâmetros, uma transformação afim é aplicada no quadro desalinhado, fazendo seus *keypoints* correspondentes se sobreporem. Como efeito colateral, esse processo pode criar bordas inválidas que devem ser tratadas. Neste caso, as bordas inválidas são removidas, o que significa que os quadros alinhados são um pouco menores do que os quadros originais. A janela de corte para bordas inválidas varia de acordo com o nível máximo de deslocamento dos *keypoints* entre os quadros. Uma construção de mosaico pode ser usada para manter o quadro com seu tamanho original, usando partes de quadros anteriores, mas tais partes estariam desatualizadas e poderiam causar resultados ruins em aplicações de detecção de IMOs. O desempenho geral da metodologia proposta é apresentado na seção a seguir.

5 Resultados Obtidos

Experimentos foram conduzidos usando-se vídeos sintéticos e reais. Os vídeos sintéticos foram capturados a partir de um simulador 3D construído especificamente para este fim, capaz de gerar cenários com elementos estáticos (como construções, ruas, relevos, vegetação e rochas) e elementos dinâmicos (como animais, pessoas e veículos). A câmera sintética simula uma câmera apontada para baixo embarcada em uma ARP sofrendo pequenos movimentos, simulando uma situação real. Todo o comportamento do simulador é guiado por um *script* textual configurável. O vídeo real foi capturado a partir da câmera embarcada na ARP real que foi construída para este trabalho. Para mostrar a capacidade de execução em tempo real de forma embarcada, a execução do algoritmo de estabilização ocorreu em um minicomputador Raspberry Pi 3 Model B, idêntico ao embarcado na ARP. O algoritmo de estabilização de imagens foi programado na linguagem C++ e, quando disponíveis, algoritmos otimizados da biblioteca OpenCV 3.3 foram utilizados. A figura 5 mostra uma captura de tela do vídeo sintético e do vídeo real, ambos capturados a uma altitude de 120m do solo. Os vídeos contém objetos móveis, adicionando um complicador extra ao problema.

A tabela 1 mostra o tempo de execução e o nível de estabilização alcançados nas diferentes situações de teste. A primeira coluna indica se é o vídeo sintético ou real. A segunda indica a resolução do vídeo. A coluna TPQ é o tempo de processamento por quadro, em milissegundos. A coluna PSNR indica o nível de similaridade entre quadros após o processo de estabilização, usando a função de similaridade PSNR, com valor em dB. A coluna SSIM apresenta o percentual de similaridade usando a função de similaridade SSIM.



Figura 5: Capturas de tela dos vídeos usados nos experimentos. No topo, uma imagem sintética e, na parte inferior, uma imagem real.

Tabela 1: Níveis de similaridade e tempo de execução dos vídeos depois de estabilizados.

Vídeo	Resol.	TPQ	PSNR	SSIM
Sint.	480p	32	33,81	90,86
Real	480p	32	24,85	72,89
Sint.	720p	65	35,72	93,49
Real	720p	66	24,61	72,63

A tabela 2 apresenta os valores de PSNR e SSIM dos vídeos originais (desestabilizados) sintético e real nas resoluções de 480p e 720p e as colunas correspondem analogamente ao que foi descrito para a tabela 1.

Tabela 2: Níveis de similaridade nos vídeos reais.

Vídeo	Resol.	PSNR	SSIM
Sint.	480p	25,03	55,34
Real	480p	18,36	26,17
Sint.	720p	24,63	50,12
Real	720p	18,07	25,63

A figura 6 apresenta gráficos comparativos com os níveis de estabilização alcançados ao longo dos primeiros 500 quadros. Uma simples análise desses gráficos e dos valores apresentados nas tabelas 1 e 2 mostra claramente a melhoria na qualidade do vídeo. Além de ter obtido desempenho plausível para execução em tempo real mesmo executando em um minicomputador, a abordagem

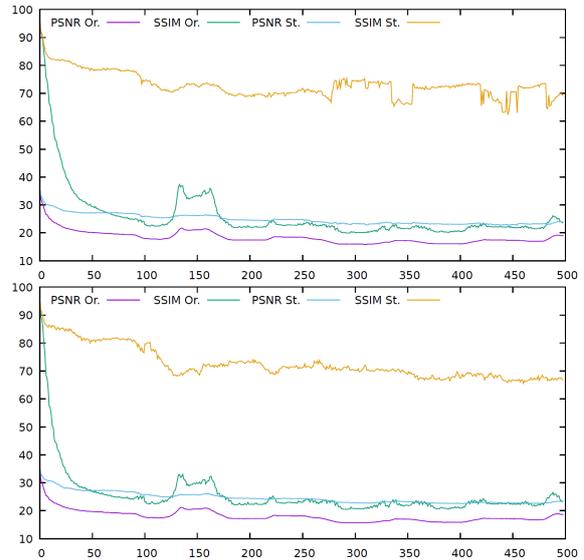


Figura 6: Estabilização dos quadros ao longo do tempo. No eixo X, os primeiros 500 quadros da sequência. No eixo Y, os valores PSNR e SSIM para os vídeos original e estabilizado. O gráfico superior corresponde à resolução de 480x854 pixels e o inferior à resolução de 720x1280 pixels. As linhas plotadas referem-se aos valores PSNR e SSIM dos vídeos originais desestabilizados e dos vídeos estabilizados.

usando fluxo óptico piramidal com otimização da seleção dos *keypoints* rastreáveis também possui simplicidade de uso, requerendo pouquíssima configuração de parâmetros quando há mudanças na resolução do vídeo. Outras abordagens presentes na literatura são mais sensíveis a tais mudanças.

Uma observação importante a ser feita é que, na presença de objetos de altitude elevada e com movimentos locais repetitivos, como árvores altas sendo balançadas pela ação do vento, o nível de similaridade dos vídeos tende a cair, mesmo depois de estabilizados, pois esses movimentos locais influenciam nos resultados das funções PSNR e SSIM. Nestes cenários, o desempenho do algoritmo proposto é também ligeiramente inferior, requerendo um tratamento diferenciado. Basicamente, a situação ocorre porque *keypoints* presentes em regiões com essas características movem-se de forma ligeiramente diferente, pois estão mais próximos da câmera. Entretanto, quanto maior a altitude da ARP, menor será a interferência desse tipo de efeito no desempenho do algoritmo.

6 Conclusões

Este artigo apresentou o processo de concepção de uma ARP de baixo custo do tipo quadricóptero capaz de capturar vídeos estabilizados do solo a partir de uma câmera monocular acoplada ao veículo e apontada verticalmente para baixo. Fo-

ram apresentadas questões técnicas que envolvem a construção do quadricóptero e a calibração dos PIDs. Ainda, um algoritmo inovador de estabilização digital de imagens implementado especificamente para o contexto em questão apresentou resultados satisfatórios e capacidade de execução em tempo real em um minicomputador do tipo Raspberry Pi 3 Model B. A utilização de uma quantidade minimizada de *keypoints* graças a uma política de descarte favoreceu o bom desempenho do algoritmo. Trabalhos futuros incluem a implementação de algoritmos de detecção e rastreamento de IMOs terrestres e a melhoria da estabilização em cenários com muitos elementos dinâmicos de altitude elevada, estabelecendo algum tipo de critério para descarte de *keypoints* nessas regiões da imagem.

Referências

- Aguilar, W. G., Abaroa, T., Fuertes, R., Ibarra, A. and Loza, D. (2017). Video stabilization system without phantom movements for cameras onboard rotocrafts, *Intelligent Transportation Systems (ITS LATAM), 2017 1st Latin American Conference on*, IEEE, pp. 1–6.
- Aguilar, W. G. and Angulo, C. (2014). Robust video stabilization based on motion intention for low-cost micro aerial vehicles, *Systems, Signals & Devices (SSD), 2014 11th International Multi-Conference on*, IEEE, pp. 1–6.
- Bay, H., Tuytelaars, T. and Van Gool, L. (2006). Surf: Speeded up robust features, *European conference on computer vision*, Springer, pp. 404–417.
- Bouguet, J.-Y. (2001). Pyramidal implementation of the affine lucas-kanade feature tracker - description of the algorithm, *Intel Corporation* 5(1-10): 4.
- Calonder, M., Lepetit, V., Ozuysal, M., Trzcinski, T., Strecha, C. and Fua, P. (2012). Brief: Computing a local binary descriptor very fast, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34(7): 1281–1298.
- Cerón, A., Mondragón, I. and Prieto, F. (2018). Onboard visual-based navigation system for power line following with uav, *International Journal of Advanced Robotic Systems* 15(2): 1–12.
- Davis, E., Nizette, B. E. and Yu, C. (2013). Development of a low cost quadrotor platform for swarm experiments, *32nd IEEE Chinese Control Conference (CCC)*, IEEE, pp. 7072–7077.
- Devaud, J.-B., Najko, S., Le Nahédic, P., Mauseire, C., Zante, E. and Marzat, J. (2012). Full design of a low-cost quadrotor uav by student team, *IEEE International Conference on System Engineering and Technology (ICSET)*, IEEE, pp. 1–6.
- Hore, A. and Ziou, D. (2010). Image quality metrics: Psnr vs. ssim, *20th international conference on Pattern recognition (ICPR)*, IEEE, pp. 2366–2369.
- Ibrahim, A. W. N., Ching, P. W., Seet, G. G., Lau, W. M. and Czajewski, W. (2010). Moving objects detection and tracking framework for uav-based surveillance, *Image and Video Technology (PSIVT), 2010 Fourth Pacific-Rim Symposium on*, IEEE, pp. 456–461.
- Jakobsen, O. and Johnson, E. (2005). Control architecture for a uav-mounted pan/tilt/roll camera gimbal, *Infotech@ Aerospace*, p. 7145.
- Mingkhwan, E. and Khawsuk, W. (2017). Digital image stabilization technique for fixed camera on small size drone, *Defence Technology (ACDT), 2017 Third Asian Conference on*, IEEE, pp. 12–19.
- Montanari, R., Tozadore, D. C., Fraccaroli, E. S. and Romero, R. A. (2015). Ground vehicle detection and classification by an unmanned aerial vehicle, *Robotics Symposium (LARS) and 2015 3rd Brazilian Symposium on Robotics (LARS-SBR), 2015 12th Latin American*, IEEE, pp. 253–258.
- Morimoto, C. and Chellappa, R. (1996). Fast Electronic Digital Image Stabilization for Off-Road Navigation, *Real-Time Imaging* (5): 285–296.
- Quigley, M., Goodrich, M. A., Griffiths, S., Eldredge, A. and Beard, R. W. (2005). Target acquisition, localization, and surveillance using a fixed-wing mini-uav and gimballed camera, *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE international Conference on*, IEEE, pp. 2600–2605.
- Stewart, C. V. (1999). Robust parameter estimation in computer vision, *SIAM review* 41(3): 513–537.
- Zhu, J., Li, C. and Xu, J. (2015). Digital image stabilization for cameras on moving platform, *International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP)*, IEEE, pp. 255–258.