

REALIZAÇÃO DE UM POUSO AUTÔNOMO DE UM DRONE UTILIZANDO TAGS DE REALIDADE AUMENTADA

Caique de S. F. Duarte (IC), Alexandre C. B. Ramos(PQ)¹

¹ Universidade Federal de Itajubá

Palavras-chave: Visão computacional. Drones. Realidade Aumentada. Posicionamento.

Introdução

Drones autônomos, ou inteligentes, são drones que podem realizar missões sem a ajuda de um piloto, eles são controlados por um computador, a bordo da aeronave ou na estação de solo.

Os drones estão dominando o mercado em espaços *outdoor* para serviços como mapeamento e inspeção de áreas de difícil acesso, apenas com a utilização de um GPS para seu posicionamento. Porém, apesar de poder voar em áreas remotas sem muito risco, ao voltar para a plataforma de pouso, o GPS pode não ser preciso o suficiente, ou a plataforma de pouso pode estar em movimento, impossibilitando um pouso seguro.

A Realidade Aumentada, é a mistura do espaço real com um espaço virtual, isso é possível graças a algoritmos de visão computacional que detectam a posição da câmera no espaço real, e projetam um espaço virtual na imagem produzida. A utilização de tags de realidade aumentada, ajudam estes algoritmos a descreverem a posição e a rotação de planos do espaço real com mais facilidade e precisão.

Portanto, este projeto visa utilizar essas tags em plataformas de pouso, ajudando um drone multirrotor inteligente, a saber sua posição relativa à plataforma, e ajustar sua velocidade para que consiga pousar com segurança, mesmo que ela esteja em movimento.

O algoritmo descrito neste documento foi desenvolvido a fim de ser utilizado pela equipe de competição acadêmica Black Bee Drones no “Desafio Drone Autônomo” como demonstração de inteligência na Droneshow L.A. em maio de 2017, e foi adaptado para ser utilizado na competição IMAV 2017 (International Micro Air Vehicle Competition and Conference) , que ocorreu em Toulouse na França em setembro do mesmo ano, onde a equipe ficou em quarta colocação entre outras vinte e duas equipes.

Este estudo tem como objetivo desenvolver um algoritmo que efetue pousos seguros com drones autônomos, sem a necessidade de utilização de sistemas de posicionamento caros ou imprecisos. Além de auxiliar a equipe Black Bee Drones a realizar a missão

Precision Landing na competição IMAV 2017, com um sistema que faz um drone multi rotor pousar em uma plataforma em movimento e chacoalhando.

Metodologia

Nesta pesquisa foi utilizado um drone do modelo Bebop 2 (Figura 1) da marca Parrot como principal instrumento de teste. Este drone era um dos mais personalizáveis em termos de *software* do mercado na época e já vem configurado para que possa ser facilmente programado. Ele possui uma câmera de 180 graus e um algoritmo de *gimball* virtual que estabiliza a câmera apontada sempre para a direção desejada, independentemente do ângulo do drone.



Figura 1 – Bebop 2

Fonte: Black Bee Drones, 2017

Para controlar o drone foi utilizado um notebook pessoal com o sistema operacional Ubuntu 14.04 devidamente preparado. Por segurança, outros softwares também estavam sendo executados no drone, como um nó python que permitia a um controle de xbox 360 sobrescrever os comandos autônomos em casos de emergência.

Como plataforma de pouso, foi utilizada uma

estrutura de $1m^2$ com uma tag de $20cm^2$ ID 4 no centro. Nesta plataforma foi amarrado um barbante para que alguém pudesse movê-la enquanto o drone tenta pousar.

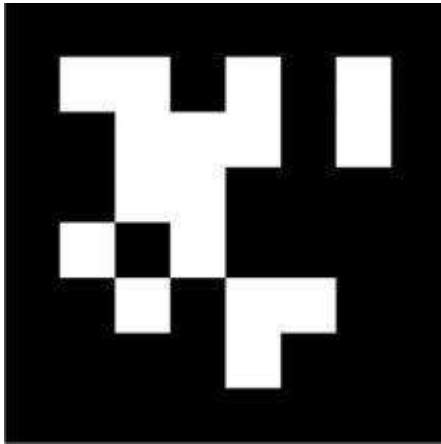


Figura 2 - Apriltag
Fonte: FIEDLER, 2019

Depois de concluir a instalação do ambiente adequado, foi desenvolvido um software em solo que pegava imagens da webcam do notebook e encontrava a *april tag* mostrando sua posição e rotação nos três eixos do espaço real. O sistema ROS facilita muito este tipo de desenvolvimento, pois você pode criar um nó separado para encontrar as *april tags*, e criar dois nós que pegam imagens de duas fontes diferentes, reaproveitando a parte do código que reconhece as tags.

Quando os testes com a webcam estavam satisfatórios, começamos a utilizar o tópico fornecido pelo próprio Bebop, com a imagem de sua câmera frontal, para fornecer imagens para o nó de detecção das tags.

O nó de detecção das tags publica em um tópico a posição e a orientação de cada uma delas. Então, um nó que recebe estes dados, processa e publica em um tópico de movimento foi criado e chamado de `comunicacao.cpp`. E finalmente, para ler os comandos de movimento e publicar no tópico “`bebop/cmd_vel`” do drone, para que ele se mova de forma adequada, foi desenvolvido um nó chamado `precision_land.cpp`.

Inicialmente o software encontrou problemas como velocidade de processamento e estabilização do drone no ponto central da tag. Este erro ocorre pois o drone sempre passa do ponto que tenta alcançar e ao tentar compensar este erro acaba passando novamente, o que é chamado em inglês de *overshoot*.

Para solucionar este problema foi implementada uma biblioteca PID que controla a velocidade do drone em função da distância entre ele e a tag, além disso foi adicionada uma margem de erro em decímetros

considerada aceitável para o alinhamento, isso fez com que o drone começasse a parar antes de chegar ao ponto, e se mantivesse estável uma vez que o alcançava.

Combinando esta adaptação com uma otimização do código, para que consumisse menos processamento o resultado se tornou satisfatório.

Outro problema encontrado foi a perda da visão da tag quando o drone começava a descer, o que fazia com que ele ficasse parado em um único ponto 3D esperando a tag reaparecer.

Foi implementada uma constante em decímetros que informava em que altitude o drone consideraria aceitável continuar o pouso mesmo após a perda da visão da tag, além desta constante, uma nova variável foi adicionada que fazia o raio aceitável dos eixos horizontais diminuir conforme a altitude do drone, para que ele ficasse cada vez mais próximo da tag enquanto descia, diminuindo o risco de perdê-la de visão.

Resultados e discussão

Por fim o drone começou a apresentar um resultado ótimo, mesmo quando a plataforma de movimento era movida de forma não uniforme, e passou a conseguir realizar o pouso na maior parte dos testes controlados.

Como mencionado na introdução, em 2017, o código objeto de estudo deste artigo foi apresentado pela equipe Black Bee Drones na Droneshow, o maior evento de drones da América Latina, que ocorre anualmente em São Paulo. A figura abaixo é uma foto tirada pela equipe durante o evento.



Figura 3 - Demonstração do pouso autônomo
Fonte: Black Bee Drones, 2017

O código também foi utilizado na competição

IMAV em 2017 em Toulouse na França, onde a equipe ficou em quarto lugar na modalidade outdoor, com uma combinação de diversas provas.

Conclusões

Durante a pesquisa foi verificada a importância da otimização do código fonte em algoritmos embarcados, além disso, foi provado que apesar do projeto conceitual do código ser válido, quando o cenário de testes é transferido para o mundo real problemas surgem e ajustes devem ocorrer.

Com algumas modificações, o algoritmo pode e será utilizado para outras aplicações da vida real, como o alijamento de carga de precisão, transporte de uma boia para vítimas na água, recolhimento de amostras, entre outras.

Agradecimento

Ao professor Alexandre Ramos, por seu essencial apoio e paciência, à FAPEMIG pelo apoio financeiro e à equipe Black Bee Drones pelo companheirismo, meus agradecimentos.

Referências

- MALYUTA, D. BROMMER, C. HENTZEN, D. STASTNY, T. SIEGWART, R. e BROCKERS, R. Long-duration fully autonomous operation of rotorcraft unmanned aerial systems for remote-sensing data acquisition. *Journal of Field Robotics*. 2019. p. arXiv:1908.06381.
- BROMMER, C. MALYUTA, D. HENTZEN, D. e BROCKERS, R.. Long-duration autonomy for small rotorcraft UAS including recharging. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE. 2018. p. arXiv:1810.05683.
- WANG, J. and OLSON, E. AprilTag 2: Efficient and robust fiducial detection. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2016.
- LENTIN, J. *ROS Robotic Projects*. 2017. Birmingham, United Kingdom: Packt Publishing.
- STANFORD ARTIFICIAL INTELLIGENCE LABORATORY et al., *Robotic Operating System*. 2018. Disponível em: <<https://www.ros.org>>. Acesso em: 23 set. 2022.
- FIEDLER, NIKLAS. Tracking with Direct FCNN Inclusion in RoboCup Humanoid Soccer. 2019. https://www.researchgate.net/publication/336614405_Distributed_Multi_Object_Tracking_with_Direct_FCNN_Inclusion_in_RoboCup_Humanoid_Soccer