HAL
open science

Estabilización Visual de un Quadrotor

Eder Rodriguez, David Lara, Gerardo Romero Galvan, Guillaume Caron, Claude Pegard

To cite this version:

Eder Rodriguez, David Lara, Gerardo Romero Galvan, Guillaume Caron, Claude Pegard. Estabilización Visual de un Quadrotor. Congreso de Instrumentacion, SOMI XXX, Oct 2015, Durango, México. hal-01270425

HAL Id: hal-01270425

https://hal.science/hal-01270425v1

Submitted on 8 Feb 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

SOMI CONGRESO DE INSTRUMENTACIÓN XXX
XXX Edición, Durango, Dgo., México, octubre 2015

Estabilización Visual de un Quadrotor

Eder A. Rodríguez, David Lara, Gerardo Romero Galván
Universidad Autónoma de Tamaulipas Carr. Reynosa - San Fernando, cruce con Canal Rodhe, Arcoíris, Reynosa Tamps., México eder_rdz_mtz@hotmail.com, dlara@uat.edu.mx, gromero@uat.edu.mx

Guillaume Caron, Claude Pegard
Université de Picardie Jules Verne
Chemin du Thil, 80000 Amiens, Francia
guillaume.caron@u-picardie.fr, claude.pegard@u-picardie.fr

RESUMEN

El presente documento trata sobre la estabilización basada en la imagen de un vehículo aéreo no tripulado UAVs tipo quadrotor (AR-Drone 2.0 de Parrot) que obtiene información mediante su cámara frontal. Se aplica la Técnica de Control Visual Basada en Imagen (IBVS, por sus siglas en ingles), y se hace uso del meta sistema operativo ROS mediante el lenguaje C++, algunas librerias de ViSP y una configuración del tipo ojo-en-mano (eye-in-hand) para procesar los datos obtenidos por la cámara. El seguimiento de las imágenes se logra utilizando el algoritmo KLT y las velocidades se calculan utilizando la pseudo inversa de la aproximación de la matriz de interacción en la ley de control. Los resultados se muestran de forma cuantitativa y cualitativa.

PALABRAS CLAVE: quadrotor, Control, características de interes, Técnica de Control Visual Basada en Imagen, ojo-en-mano, ROS, KLT.

1 INTRODUCCIÓN

En las últimas décadas el avance de la robótica ha impulsado el desarrollo de los UAVs. Estos vehiculos tienen muchas aplciaciónes en situaciones que ponen en riesgo a los seres humanos y en las que su acceso es limitado, por ejemplo en un desastre natural, un terremoto, una inundación, una catastrofe nuclear, o en tareas que faciliten el trabajo del hombre, vigilancia de tráfico, tomas aereas, busqueda de personas entre otras [1]. El auge de los UAVs ha propiciado el desarrollo de algunas disciplinas tales como la aerodinamica, la mecanica, el control automatico y el control. Esta ultima disciplina tiene un gran campo de estudio y resultados muy importantes en la robotica. El control visual combina procesamiento de imágenes con técnicas de control de tal forma que la información se usa para la retroalimentación del sistema. El problema principal del control visual es el hecho de obtener una interpretación robusta, fiel y en tiempo real del campo visual mediante estructuras de control y algoritmos. Las soluciones provistas por el control visual típicamente se dividen en Técnicas de Control Basadas en Imagen

(IBVS) y Técnicas de Control Basadas en Pose (PBVS). Por otra parte, las imágenes pueden ser obtenidas mediante una cámara montada directamente en el manipulador del robot o en un robot movil (ojo-en-mano), en estos casos, el movimiento del robot induce el movimiento de la cámara, o la cámara puede estar fija en el espacio de trabajo y así poder observar el movimiento del robot desde una configuración estacionaria (mano-a-ojo). Otras configuraciones pueden ser posibles, por ejemplo, montar un conjunto de imágenes sobre una base fija [2].

El quadrotor AR Drone 2.0 de Parrot fue elegido para realizar esta investigación. Es controlado por medio de una red Wi-Fi desde algún dispositivo tal como una laptop, un ipad o un smartphone con sistema operativo android. Puede ser manipulado utilizando el meta-sistema operativo ROS [3] y el controlador ardrone_autonomy desarrollado por Autonmy Lab [4]. Cuenta con una cámara frontal HD 720p a 30FPS, un giroscopio de 3 ejes con una presición de 2000°/segundo, un acelerometro de 3 ejes con una presición de $\pm$50mg, un magnetometro con precision de hasta 6°, un barometro con una presición de $\pm$10Pa, un sensor ultrasónico para medir la altura y una cámara vertical QVGA que mide velocidad con respecto al suelo. Los motores son sin escobillas de 14.5W, 28500RPM [5].

2 CONCEPTOS BASICOS DE VISÓN

Las imágenes son simplemente un arreglo grande de valores de los pixeles, pero, para las aplicaciones de robótica, las imágenes tienen muchos datos e insuficiente información [1]. Las características de interes en la imagen son medidas que pueden contener la información de la pose de algun objeto, el tipo de objeto o su velocidad. Existen diferentes tipos de características de interes en la imagen, por ejemplo, puntos, lineas o áreas. En este artículo se emplean los puntos de interes de la imagen que se obtienen por medio del algoritmo KLT.

3 CONTROL VISUAL

Consiste básicamente en integrar, en tiempo real, la información visual en el bucle de control de un robot manipulador [6]. Para realizar un control visual se necesitan encontrar las características de interes en la imagen, procesar la información y generar los valores de consignia correspondientes. El propósito de todo el control basado en visión se centra en minimizar el error $\mathbf{e}(t)$, típicamente definido por $\mathbf{e}(t) = \mathbf{s}(\mathbf{m}(t), \mathbf{a}) - \mathbf{s}^*$. El vector $\mathbf{m}(\mathrm{t})$ es un conjunto de medidas de la imagen. Estas medidas de imagen se usan para calcular el vector de k elementos visuales, $\mathbf{s}(\mathbf{m}(\mathrm{t}), \mathbf{a})$, en el cual $\mathbf{a}$ es un conjunto de parámetros que representa información potencial adicional sobre el sistema. El vector $\mathbf{s}^*$ contiene los valores deseados de

los elementos. Una vez seleccionado, el diseño del esquema de control puede ser más simple. Tal vez el enfoque más directo sea el diseño de un control de velocidad. Para hacer esto, requerimos de una relación entre la variación del tiempo de $\mathbf{s}$ y la velocidad de la cámara. Sea la velocidad de la camara denotada por $\mathbf{v}_c = (\boldsymbol{v}_c, \boldsymbol{\omega}_c)$, $\boldsymbol{v}_c$ la velocidad lineal instantanea del origen del marco de la cámara y $\boldsymbol{\omega}_c$ la velocidad angular instantanea desde el marco de referencia de la cámara, la relacion entre $\dot{\mathbf{s}}$ y $\mathbf{v}_c$ esta dada por $\dot{\boldsymbol{s}} = \mathbf{L_s}\mathbf{v}_c$, en donde $\mathbf{L_s} \in \mathbb{R}^{\hbar\times 6}$ es llamada la *matriz de interacción* relacionada a $\mathbf{s}$. Entonces la relación entre la velocidad de la cámara y la variación del error en el tiempo:

$$\dot{\mathbf{e}} = \mathbf{L_e}\mathbf{v}_c, \tag{1}$$

donde $\mathbf{L_e} = \mathbf{L_s}$. Considerando $\mathbf{v}_c$ como la entrada del controlador del robot, y si quisieramos, por ejemplo, intentar asegurar el desacople del decrecimiento exponencial del error (i.e., $\dot{\mathbf{e}} = -\lambda\mathbf{e}$), usando (1) obtenemos:

$$\mathbf{v}_\mathrm{c} = -\lambda\mathbf{L}_\mathbf{e}^{+}\mathbf{e}, \tag{2}$$

donde $\mathbf{L}_\mathbf{e}^{+} \in \mathbb{R}^{\hbar\times 6}$ es elegido como la pseudo-inversa Moore-Penrose de $\mathbf{L_e}$, que es $\mathbf{L}_\mathrm{e}^{+} = (\mathbf{L_e}^{\intercal}\mathbf{L_e})^{-1}\mathbf{L_e}^{\intercal}$ cuando $\mathbf{L_e}$ es de rango completo 6. Esta elección permite $\|\dot{\mathbf{e}} - \lambda\mathbf{L_e}\mathbf{L}_\mathbf{e}^{+}\mathbf{e}\|$ y $\|\mathbf{v}_c\|$ ser mínimo. Cuando $\hbar = 6$ si el determinante $\mathbf{L}_\mathrm{e} \neq 0$, es posible invertir $\mathbf{L}_\mathrm{e}$, resultando en el control $\mathbf{v}_c = -\lambda\boldsymbol{L}_\mathbf{e}{}^{-1}\mathbf{e}$. En sistemas reales de control visual, es imposible conocer perfectamente en práctica $\mathbf{L_e}$ o $\mathbf{L}_\mathbf{e}^{+}$. Entonces se requiere realizar una aproximación o una estimación de una de esas dos matrices. A continuación, denotamos la aproximación de la pseudo inversa de la matriz de interacción con el símbolo $\widehat{\mathbf{L}_\mathbf{e}^{+}}$. Usando esta notación, la ley de control es de hecho

$$\mathbf{v}_c = -\lambda\mathbf{L}_\mathbf{e}^{+}\mathbf{e}. \tag{3}$$

Este es el diseño básico implementado en la mayoria de los controladores visuales [7].

3.1 Control Visual Basado en Imagen

Los esquemas tradicionales de control basado en la imagen [8], [9] usan las coordenadas del plano de la imagen de un set de puntos para definir el set $\mathbf{s}$. Las medidas de la imagen $\mathbf{m}$ son, usualmente, coordenadas de pixel en un conjunto de imágenes, y los parametros $\mathbf{a}$ en la definición de $\mathbf{s} = \mathbf{s}(\mathbf{m}, \mathbf{a})$ son los parámetros intrínsecos de la cámara que relacionan las medidas de la imagen expresadas en pixeles con las cáracteristicas en la imagen [7].

3.2 La Matríz de Interacción

Para un punto 3-D con coordenadas $\mathbf{X} = (X,\ Y,\ Z)$ en el marco de referencia de la cámara, que proyecta en la imagen un punto en 2-D con coordenadas $\mathbf{x} = (x, y)$, tenemos:

$$\begin{cases} x = \frac{X}{Z} = (u - c_u)/f\alpha \\ y = \frac{Y}{Z} = (v - c_v)/f \end{cases}, \tag{4}$$

donde $\mathbf{m} = (u, v)$ son las coordenadas de un punto de la imagen expresadas en unidades de pixel, y $\mathbf{a} = (c_u, c_v, f, \alpha)$ es el conjunto de los parámetros intrinsecos de la cámara: c_u y c_v son las coordenadas del punto principal, f es la distancia focal, y α es el cociente de las dimensiones del pixel. En este caso, tomamos $\mathbf{s} = \mathbf{x} = (x, y)$, como las coordenadas del punto del plano de la imagen.

Derivando (4) obtenemos

$$\begin{cases} \dot{x} = \frac{\dot{X}}{Z} - \frac{X\dot{Z}}{Z^2} = (\dot{X} - x\dot{Z})/Z \\ \dot{y} = \frac{\dot{Y}}{Z} - \frac{Y\dot{Z}}{Z^2} = (\dot{Y} - y\dot{Z})/Z \end{cases}. \tag{5}$$

La velocidad de los puntos en 3-D de la cámara se relacionan con la velocidad espacial en

$$\dot{\mathbf{X}} = -\boldsymbol{v}_c - \omega_c \times \mathbf{X} \Leftrightarrow \begin{cases} \dot{X} = -v_x - \omega_y Z + \omega_z Y \\ \dot{Y} = -v_y - \omega_z Z + \omega_x Y \\ \dot{Z} = -v_z - \omega_x Z + \omega_y Y \end{cases}. \tag{6}$$

Combinando (5) y (6) obtenemos

$$\begin{cases} \dot{x} = -\frac{v_x}{Z} + \frac{xv_z}{Z} + xy\omega_x - (1 + x^2)\omega_y + y\omega_z \\ \dot{y} = -\frac{v_y}{Z} + \frac{yv_z}{Z} + (1 + y^2)\omega_x + xy\omega_y - x\omega_z \end{cases} \tag{7}$$

puede ser escrita como $\dot{\mathbf{x}} = \mathbf{L_x}\mathbf{v}_c$, donde la matríz de interacción $\mathbf{L_x}$ se relacionan con $\mathbf{x}$ en

$$\mathbf{L_x} = \begin{bmatrix} -\frac{1}{Z} & 0 & \frac{x}{Z} & xy & -(1 + x^2) & y \\ 0 & -\frac{1}{Z} & \frac{y}{Z} & 1 + y^2 & -xy & -x \end{bmatrix}. \tag{8}$$

En la matríz $\mathbf{L_x}$, el valor de la profundidad Z es el punto relativo en el marco de la cámara. Por lo tanto, en cualquier esquema de control que use esta matríz de interacción se debe de estimar o aproximar el valor de Z. Similarmente, los parámetros intrínsecos de la cámara estan involucrados en el cálculo de x e y. Así, $\mathbf{L_x}$ no puede ser directamente usado en (3), entonces se debe de usar $\widehat{\mathbf{L_x}}$ como una estimación o aproximación. Se necesitan al menos 3 puntos para controlar 6 grados de libertad. Si usamos un vector de característica $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3)$, al apilar los puntos en la matríz de interacción, se obtiene

$$\mathbf{L_x} = [\, \mathbf{L_{x_1}} \quad \mathbf{L_{x_2}} \quad \mathbf{L_{x_3}}]^T. \tag{9}$$

En este caso, existen algunas configuraciones para cual $\mathbf{L_x}$ es singular [10]. Además, existen cuatro poses de la cámara para cual $\mathbf{e} = 0$, es decir, existen cuatro mínimos globales, y es imposible diferenciarlos [11]. Por estas razones, generalmente se consideran más de tres puntos.

4 METODOLOGÍA

Algoritmo KLT: El seguidor de puntos de interes Kanade-Lucas-Tomasi (KLT) está diseñado para detectar y seguir un conjunto de características en la imagén dentro de una secuencia de imágenes. Se asume un desplazamiento pequeño entre dos imágenes consecutivas. El seguimiento esta basado en los gradientes de la imagen. Si d es el desplazamiento de un punto de interes x entre dos imágenes I_t e I_{t+1} entonces $I_{t+1}(x) = I_t(x + d)$, esta expresión puede ser aproximada por:

$$I_{t+1}(x) \approx I_t(x) + dI_t^{'}(x), \tag{10}$$

solamente una región que rodea al punto de interes es utilizada para calcular su desplazamiento. d es calculada por minimización. Los puntos de interes $\mathbf{s}$ son usualmente detectados usando los valores característicos de la matriz de la derivada de segundo orden de la imagen. Solo los elementos con valores característicos altos son consideradas como puntos de interes [12].

5 RESULTADOS

La Figura 1 muestra como las velocidades calculadas decrecen cuando el dron se desplaza de izquierda a derecha hasta llegar a su posición deseada. La rapidez $|v_y|$ es mayor que la rapidez lineal en los demas ejes y $\omega_z \neq 0$ debido a que las velocidades v_y y ω_z no estan desacopladas.

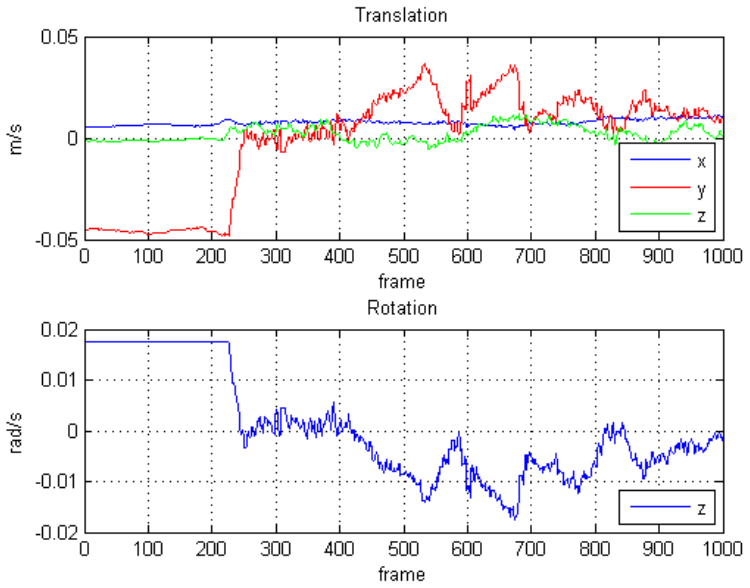

Figura 1. Velocidades calculadas cuando la traslación se realiza sobre el eje y

La aplicación del control visual se llevó a cabo en un lugar cerrado y con suficiente iluminación donde las corrientes de aire era pequeñas. La Figura 2 muestra las imagenes tomadas por el UAV mientras realizaba el control visual desplazandose lateralmente hasta obtener la imagen deseada (primera imagen).

Figura 2. Estabilización visual

6 CONCLUSIONES

El código funciona bien, pero, en algunos casos, el comportamiento del quadrotor no es el ideal debido a las corrientes de aire, el cambio en la iluminación, la latencia generada por los procesos de computación y al guardar imágenes, el desplazamiento del quadrotor en el modo hovering y que la rapidez lineal entre .01 y .1 no son obedecidas correctamente por el AR Drone 2.0.

7 AGRADECIMIENTOS

Este trabajo de investigación fue desarrollado en la Universidad Autónoma de Tamaulipas, UAT Reynosa Rodhe en colaboración con la Université de Picardie Jules Verne de Amiens, Francia.

REFERENCIAS

[1] Corke P. Robotics, Vision and Control. 1st ed. Berlin: Springer-Vertag 2011.

[2] Pascual Campoy. Campoy P, Mondragón F, Iván O. et al. Visual Servoing for UAVs. InTech, Rijeka Croatia 2010.

[3] ROS (Robot Operating System) provides libraries and tools to help software developers create robot applications. Avalible at: http://wiki.ros.org/es (Accesed on: August 21, 2015)

[4] Parrot Wi-Fi quadricopter. Augmented Reality games on iPhone, iPod touch and iPad & Android. Avalible at: http://ardrone2.parrot.com/ (Accesed on: August 21, 2015)

[5] ROS driver for Parrot AR-Drone 1.0 and 2.0 quadrocopters. Avalible at: https://github.com/AutonomyLab/ardrone_autonomy (Accesed on: August 21, 2015)

[6] Pomares J, Gutierrez R, López M, et al. Seguimiento de trayectorias 3D mediante el Control Visual Basado en Imagen. Universidad de Alicante. Alicante, España.

[7] Chaumette F, Hutchinson S. Visual Servo Control Part1: Basic Approaches. **IEEE Robotics & Automation Magazine**. France 2006.

[8] Feddema J, Mitchell O. Vision-guided servoing with featurebased trajectory generation." IEEE Trans. Robot. Automat. 1989.

[9] Weiss L, Sanderson A, Neuman C. Dynamic sensor-based control of robots with visual feedback. IEEE J. Robot. Automat. 1987.

[10] Michel H, Rives P. Singularities in the determination of the situation of a robot effector from the perspective view of three points. INRIA Research Report, Tech. 1993.

[11] Fischler M, Bolles R. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Communicat. ACM. 1981.

[12] Chaumette F, Marchand E, Melchior N, et al. Image manipulation and processing. ViSP tutorial, Lagadic project 2012.