Local navigation of quadrocopter using integrated inertial/GPS/camera sensors

Martin Stokkeland

December 9, 2013

Abstract

Contents

1 Introduction

1.1 Background and Motivation

Over the past years the unmanned aerial vehicle (UAV) has emerged as a subject of great interest in robot research. The quadrocopter in particular, possesses a multitude of properties which makes it attractive for a wide range of operations. It has excellent maneuverability thanks to the distributed rotor structure. This combined with the light weight and small size makes it suitable for many tasks which would otherwise not be possible with a conventional helicopter.

Among fields where quadrocopters have been employed are military operations, search and rescue, surveillance, structure inspection and even pizza delivery. In this paper, however, the focus will be on wind turbine inspection. **?** described several reasons why using a small UAV is preferable in a similar case, namely inspection of power lines. One reason is the hazards involved in flying a manned helicopter at close range. An autonomous or remotely controlled quadrocopter on the other hand would be able to approach the target closely without high risk, and in case of impact damage would be minor. Another argument is the cost, which is significantly smaller than it would be by using a heavy manned helicopter or a time consuming manual inspection by foot.

The main approaches of wind turbine inspection utilized presently are using telescopes and climbing or using remotely controlled micro air vehicles (MAVs). In order to eliminate the need of human time investment it would be preferable to perform this task autonomously. Both the GPS system and camera sensors are useful for this purpose, however for local navigation GPS tends to become inaccurate. This paper will focus on how a computer vision system can be implemented in order to effectively track a wind turbine.

1.2 Organization of this Report

First, the problem is investigated in order to identify the main characteristics. A strategy is then formed and an overview of the algorithm is given, before the individual steps are described. Next, the program is run on a variety of test data and the results are presented. The results are then compared and discussed, before finally some future improvements are suggested.

1.3 Nomenclature and Notation

UAV - Unmanned aerial vehicle MAV - Micro air vehicle BGR - Blue/green/red color space.

2 Materials and Methods

2.1 Key Features of a Wind Turbine

In order to track an object it is of key importance to identify the characteristics which distinguish the object from its surroundings. The following properties will now be investigated:

- Texture
- Color
- Geometry
- Size
- Movement
- Point of view
- Environment

2.1.1 Texture

The smooth surface of a wind turbine has virtually no texture, which could cause problems with motion detection up-close. This is however outside of scope of this report, since the wind turbine will be observed from a distance where the turbine hub is inside the image frame. On the other hand, lack of texture means that the color is rather homogeneous, a quality which will be exploited.

2.1.2 Color

Despite being white in color, a wind turbine can appear in any shade of gray depending on the lighting conditions. However, no particular hue should be present if a proper camera is used.

2.1.3 Geometry

The distinct shape of the wind turbine is of great benefit. In particular, the straight lines along the rotor blades and the tower are favorable subjects for edge or line detection algorithms. Furthermore, the radial shape with four angles (or three when a blade on the tower overlap) is recognizable when analyzing contours.

2.1.4 Size

Since GPS is more practical to use at a broad scale, the UAV is expected to be close to the wind turbine when the algorithm is used, in which case the wind turbine usually has all the rotor blades stretching outside of the image frame.

2.1.5 Movement, time variance

2.1.6 Point of view

It will generally be assumed that the UAV is positioned in front or back of the wind turbine such that the rotors or tower do not overlap. When directly in the front or back the angle between the rotor blades are 120 degrees. A change of perspective would skew the image and thus the new angles could be used to determine orientation.

2.1.7 Environment

Unless the terrain highly inclined or the image is taken from a high angle, the sky will compose most of the background. Under clear weather conditions a blue sky serves as excellent contrast to the wind turbine. However, clouds and gloomy weather are anticipated to be a significant cause of error in the program.

2.1.8 Summary of Key Features

2.2 Programming environment

The program was written in C++ utilizing the open source computer vision library, OpenCV. c++ opencv pandaboard?

2.3 Program Overview and Strategy

The program was designed in an incremental fashion where each module contribute as an individual step towards the final result. Introduce the plan/strategy, include UML or similar diagram. Should all modules be introduced at once, or should the program be presented incrementally as each module is described.

2.4 Module Descriptions

Work independently, more could be added to improve the algorithm.

2.4.1 Overhead Image Processing

2.4.2 Downsampling using Gaussian pyramid

The resolution of an image has a direct impact on computation time for most of the basic image processing functions. Hence, reducing the resolution is often the easiest and most effective way to reduce computational costs. Since the UAV on which the program is developed for will have limited processing power it would be preferable to keep the resolution as small as possible without prohibiting a usable result.

This part can be omitted if a camera which can readily stream images at an acceptable resolution is chosen. Otherwise the method halves the image resolution incrementally until it becomes lower than a user specified threshold.

The challenge of downsampling resides in deciding how to reduce pixel resolution while simultaneously minimizing change in visual appearance. Usually this is done by using image pyramids. An image pyramid refers to a representation where an image is repeatedly filtered and downsampled, in which the

Figure 1: Cylindrical representation of the HSV color space.

filtering method defines the type of image pyramid. For this program, a Gaussian image pyramid is used, elaborated by **?**. The Gaussian filter is a smoothing (lowpass) filter where every pixel has its new value given as a weighted average of its surrounding pixels. In this particular case the kernel given by (1) is used, indicating how the surrounding pixels are weighed centered around the pixel being calculated.

$$H = \frac{1}{256}\begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 16 & 24 & 36 & 24 & 16 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix} \tag{1}$$

This is achieved by using the function *pyrDown* in OpenCV.

2.4.3 HSV color space convertion

An image is initially represented in the BGR color space when imported using OpenCV, which is a rearranged version of the RGB color space. BGR and RGB are both additive color models in the sense that each pixel's color value is defined by the indiviual intensitiey of the colors red, green and blue. Every other possible color is then produced by mixing these primary colors. The wind turbine, however, has a color distribution for which BGR is not the most preferable color space. As mentioned in section 2.1.2, no particular color should dominate the tone of the wind turbine, and thus BGR has no apparent advantage in terms of object recognition. In addition, RGB is often counterintuitive for humans to work with compared to other color space representations.

For the reasons stated above, another the HSV color space was chosen instead, **?**. It consists of the channels hue, saturation and value, which can be represented in a cylindrical coordinate system, as seen in figure 1. The hue parameter defines what one would call the pure color, e.g. blue, green, magenta. How intense these colors appear are given by the saturation parameter. This works in such a way that reducing the saturation to a minimum would convert the image to a grayscale image. Lastly, the value parameter can be equated to brightness. Increasing value reduces the amount of shade in the color, but does not add white. The result is that saturation and brightness, which are the properties of interest of the wind turbine in terms of colors, can be easily manipulated or tracked.

2.4.4 Multiband Thresholding

Now that the image is preprocessed into a more practical form, the act of tracking the wind turbine can begin. First off, the wind turbine needs to be separated from the background, which will be done through thresholding. This is a segmentation technique that outputs a binary image based on which pixels pass a threshold value. For a grayscale image this corresponds to highlighting the pixels which are above a certain intensity, see (2)

$$dst(x,y) = \begin{cases} 1 & \text{if src(x,y)} \geq \text{threshold} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

Since the image is represented in the HSV color space the method must be augmented to include all channels. Hence, a separate threshold value is added for each parameter, thus creating the multiband thresholding method. In addition, a lower bound is added in order to permit an arbitrary range to be chosen. The resulting method is expressed in (3)

$$dst(x,y) = \begin{cases} 1 & \text{if lowerH} \leq \text{src(x,y,H)} \leq \text{upperH} \\ & \text{and lowerS} \leq \text{src(x,y,S)} \leq \text{upperS} \\ & \text{and lowerV} \leq \text{src(x,y,V)} \leq \text{upperV} \\ \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

An alternative approach could involve performing some analysis to identify the wind turbine *prior* to background separation. In this algorithm however, separation is done first and with the intent to later apply a quality check on the extracted wind turbine image. This choice was made under the assumption that the color of the wind turbine can be predicted somewhat accurately, as mentioned in section 2.1.2. Hence with the right initial parameters it should be possible to quickly find the right color range for separation.

2.4.5 Canny Edge Detection

2.4.6 Dilation Filter

2.4.7 Finding Contours

2.5 Autonomous Parameter Tuning Using Moments

2.6 Input Samples

Single image, video, bad conditions. Isolate problems. Image: initial algorithm, finding parameters, controlled/fixed environment Ill conditioned: check for robustness, how far before impossible Video: investigate frame to frame variations,

The program is capable of both taking still images as well as video clips or video streams as input, which allows more control and diversity on testing environment. Any resolution is accepted, but in order to limit computation costs the input array will be downscaled past a user-defined limit using the Gaussian pyramid.

To begin with, still images will be used as input in order to investigate the basic functionality of the algorithm. This way it will be clear how the image filtering methods handle specific image conditions. Images will range from very good to very ill-conditioned. An image can typically be classified as good by possessing the following properties: Clear foreground to background distinction, absence of objects of similar color and shape as the wind turbine, insignificant amounts of noise and homogeneous colors on the wind turbine (i.e. few shadows and no overlapping objects).