Measurement 81 (2016) 264–269

Contents lists available at ScienceDirect

Measurement

journal homepage: www.elsevier.com/locate/measurement

Embedded, real-time UAV control for improved, image-based 3D scene reconstruction

Jean Liénard [a], Andre Vogs [b,1], Demetrios Gatziolis [c], Nikolay Strigul [a,*]

[a] Department of Mathematics and Statistics, Washington State University Vancouver, Vancouver, WA 98686, USA
[b] Intel Inc., Hillsboro, OR 97124, USA
[c] USDA Forest Service, Pacific Northwest Research Station, Portland, OR 97204, USA

ARTICLE INFO

Article history:
Received 11 August 2015
Received in revised form 3 December 2015
Accepted 14 December 2015
Available online 18 December 2015

Keywords:
3D reconstruction
Aerial robotics
Computer vision
Robotics in agriculture and forestry
Real-time photogrammetry

ABSTRACT

Unmanned Aerial Vehicles (UAVs) are already broadly employed for 3D modeling of large objects such as trees and monuments via photogrammetry. The usual workflow includes two distinct steps: image acquisition with UAV and computationally demanding post-flight image processing. Insufficient feature overlaps across images is a common shortcoming in post-flight image processing resulting in the failure of 3D reconstruction. Here we propose a real-time control system that overcomes this limitation by targeting specific spatial locations for image acquisition thereby providing sufficient feature overlap. We initially benchmark several implementations of the Scale-Invariant Feature Transform (SIFT) feature identification algorithm to determine whether they allow real-time execution on the low-cost processing hardware embedded on the UAV. We then experimentally test our UAV platform in virtual and real-life environments. The presented architecture consistently decreases failures and improves the overall quality of 3D reconstructions.

1. Introduction

The interest in and popularity of Unmanned Aerial Vehicles (UAVs) have been increasing rapidly [12,7]. In particular, UAV-based, photogrammetric 3D reconstruction of large objects has been employed in forestry [10] and archaeology [26]. In these practical applications, UAVs serve simply as platforms carrying low-cost, off-the-shelf digital cameras that acquire images along a preprogrammed, GPS-enabled trajectory at a predetermined frame rate [12]. Dynamic trajectory adjustments enable obstacle avoidance [22], aggressive maneuvers [17], landing [2,18,25], and frontier-based exploration [11]. Image processing can be either outsourced to a remote computer through a WiFi connection [21,22] or performed on-board, as demonstrated by Meier et al. [16] who developed a complete UAV platform with an embedded computer. On-board image processing, a promising direction for photogrammetry applications, is currently heavily constrained by the computational abilities of embedded hardware [5,16]. Real-time processing of imagery acquired by a UAV and based on a standard stereo camera arrangement has been often used for Simultaneous localization and mapping (SLAM, [16]) while other UAV architectures have relied on two or more cameras in various configurations [12,27,31]. For example, Yang et al. [31] combined non-overlapping downward- and forward-oriented cameras to track features across a larger field of view. Shen et al. [27] relied on two cameras with different frame rates and fusion of monocular and stereo approaches to estimate the UAV's position.

In the typical UAV-based photogrammetry workflow, acquired HD photographs (5–12 MP) are processed after

---

[*] Corresponding author.
E-mail address: nick.strigul@wsu.edu (N. Strigul).

[1] A. Vogs contributes to this work as a volunteer at Washington State University.

http://dx.doi.org/10.1016/j.measurement.2015.12.014
0263-2241/

the flight, on a computer system equipped with adequate hardware resources [26,10,30]. The independence between the image acquisition and image processing phases is the most serious shortcoming of this workflow because it often leads to scene reconstruction failures attributed to insufficient feature overlap between images [10]. Such failures can be costly, as when the scene is located in a remote area, or irreversible, as where the scene has been altered permanently after the UAV flight (e.g. forest stand burned). A naive solution to avoiding these failures would be to increase the acquisition frame rate while reducing the speed of the UAV platform [10]. However, such measures address only indirectly the core problem of non-overlapping features across images, which also depends on other factors relative to image quality during the flight (such as changing illumination conditions, complexity of dominant scene objects, or movement of objects by wind). In practice, increasing the frame rate reduces image clarity and poses technical challenges relative to the time required to store HD photographs, resulting in a sharp increase in cameras cost. An affordable UAV system allowing consistently complete reconstructions of scene objects must therefore be able to handle changes in image feature density during a flight, a condition that requires simultaneous image acquisition and processing. We address this requirement in the present study and introduce a low-cost solution that detects and compensates for insufficient feature matches between sequentially acquired images by employing on-board, real-time image processing and ultimately guarantees complete 3D reconstruction of the targeted objects.

2. Methods

2.1. UAV platform

We built a custom UAV (Fig. 1) using a DIJ F550 Hexacopter Frame, a 3DR Pixhawk autopilot equipped with GPS and compass (3D Robotics Inc.), a 915 MHz telemetry radio to the ground station computer, a FrSky receiver, a Spectrum DX7 transmitter, a Tarot T-2D Brushless gimbal, and 3 cell LIPO batteries which sustained a 20-min flight. The UAV carries an open-source, single-board Minnowboard Max computer (Intel Inc.) featuring a dual core ATOM CPU at 1.33 GHz, 2 GB of DD3 RAM, a GPU chipset, and a set of input/output ports adequate for platform control and functionality. This board has low power consumption (less than 10 W), which is convenient for UAV applications.

Our UAV platform shares similarities to the one developed by [16], where image frames acquired with low resolution (0.3 MP), high frame rate, low latency, stereo CCD cameras were processed in real-time on an embedded computer. Instead, our platform carries two imaging systems: (1) an inexpensive, low-resolution web camera that delivers a stream for real-time processing and (2) a high-resolution GoPro camera (5–12 MP) triggered through its serial connection. We use a single-board computer (Intel's Minnowboard Max) in lieu of the combination of the custom-built baseboard and Computer-on-Module used by [16]. Our configuration supports expeditious system assembly and has lower cost, but also limited customization options for its base components.

2.2. Real-time keypoint detection

SIFT [14] is the most widely used algorithm for detection of image keypoints; more recent alternatives include SURF [3], ORB [24] and BFROST [9]. SIFT is part of our current postprocessing workflow [10] and while slower than its competitors, it allows more efficient keypoint matching [9,4,6]. We benchmarked several open-source implementations of the SIFT algorithm with the on-board computer including Ezsift,[2] a C++ implementation; the OpenCV implementation[3]; SIFT_PyOCL, an implementation exploiting GPU with OpenCL [20]. To compare these libraries and assess the feasibility of real-time feature detection, we measured the time needed to process the same set of images. We also considered the idealized scenario where pictures have already been loaded into memory, and the on-board computer does not perform trajectory control nor communicates with the autopilot.

2.3. Real-time active control

Prior experience [10] suggests that the number of keypoints matched across images is a reliable indicator for a successful 3D scene reconstruction. Our prior data acquisition approach was acquiring images at specific GPS waypoints along the drone trajectory preprogrammed with PIXHAWK autopilot. The addition of the single board computer enables a feedback control loop in this image acquisition scheme. The overall goal of the active control strategy is thus to ensure that the number of keypoint matches is adequate for the post flight reconstruction of the scene. The implemented active control strategy consists of UAV backtracking to the halfpoint between the last two GPS waypoints to take an additional high-resolution picture, if the number of keypoints matched between two sequential images falls below a predefined threshold.

Specifically, when the UAV arrives at a mandatory waypoint, it starts loitering and the following two events occur simultaneously. The high-resolution camera is triggered to take a picture. At the same time, the single-board computer starts identifying keypoints from the latest low-resolution frame of the web camera. As soon as keypoints are extracted and matched against those at the previous location the UAV moves toward the next waypoint if the matching ratio is high enough, or backtracks to the halfway point if the matching ratio is below the set threshold.

2.4. Virtual reality experiments

We relied on simulation to validate the active control algorithm and troubleshoot our implementation. We first generated a virtual environment and rendered it using the open-source program POVRAY (Persistence of Vision

---

[2] `http://sourceforge.net/p/ezsift/wiki/Home/`.

[3] Version 2.4.11, `http://opencv.org`.