HoppingDuster : Self-adaptive Cleaning Robot based on Aerial Vehicle

Kenji Tei
National Institute of Informatics
2-1-2 Hitotsubashi
Chiyoda-ku, Tokyo, Japan
tei@nii.ac.jp

Kazuya Aizawa
Waseda University
1-104 Totsukamachi
Shinjuku-ku, Tokyo, Japan
zawawa1991@gmail.com

Shunichiro Suenaga
National Institute of Informatics
2-1-2 Hitotsubashi
Chiyoda-ku, Tokyo, Japan
suenaga@nii.ac.jp

Ryuichi Takahashi
Waseda University
1-104 Totsukamachi
Shinjuku-ku, Tokyo, Japan
ryuichi-t@aoni.waseda.jp

Shun Lee
the University of Tokyo
7-3-1 Hongo
Bunkyo-ku, Tokyo, Japan
ijun.slee@gmail.com

Yoshiaki Fukazawa
Waseda University
1-104 Totsukamachi
Shinjuku-ku, Tokyo, Japan
fukazawa@waseda.jp

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the Owner/Author.Copyright is held by the owner/author(s).
UbiComp'14 Adjunct, September 13–17, 2014, Seattle, WA, USA ACM 978-1-4503-3047-3/14/09.
http://dx.doi.org/10.1145/2638728.2638767

Abstract
Home cleaning robots have become popular. Most of the home cleaning robots are based on ground vehicles. While the cleaning robots based on ground vehicles can vacuum or wash floors robustly and efficiently, but they only clean on floors, not on stairs or furnitures. In this demo, we show a new concept of cleaning robot, called HoppingDuster. HoppingDuster is based on an aerial vehicle, which can fly to stairs or furnitures to be cleaned, and hop or hover to wipe or blow down dust on them toward the floor so that the ground cleaning robot can vacuum the dust. HoppingDuster adapts its behavior to finish it cleaning within designated time and battery capacity.

Author Keywords
Cleaning robot, quadrotor, self-adaptiveness

ACM Classification Keywords
H.5.m [Information interfaces and presentation (e.g., HCI)]: Miscellaneous.

General Terms
Design, Reliability

Introduction

Home robots attract much attention to automate house housework. Cleaning robots are most successful home robots, which are widely used in many families. Most of the home cleaning robots are based on ground vehicles, like Roomba. They can vacuum or wash floors robustly and efficiently.

Although the ground cleaning robot can climb over a small difference in level on the floor, it cannot reach other places far from the floor in a house, such as stairs or furnitures. Thus, many parts in a house are still uncleaned only by the ground cleaning robots.

Our idea is to utilize unmanned aerial vehicles for house cleaning. However, many challenges exist to apply the aerial cleaning robot for house cleaning. One challenge is limitation of payload. Aerial cleaning robots should be small to work in a house, but payload of it is quite small. It is hard to attach a vacuum device used in ground cleaning robot to the aerial cleaning robot. Another challenge is limitation of battery capacity. In particular, flying consumes battery. For example, CrazyFlie[1], which is a small aerial vehicle, can fly up to 7 minutes. It is difficult to attach large battery to aerial vehicle due to the limitation of payload. Aerial cleaning robots should deal with the limitations of payload and battery capacity.

HoppingDuster

We propose a new concept of cleaning robot, called HoppingDuster. HoppingDuster is based on a quadrotor, which can fly to stairs or furnitures to be cleaned.

Two ways for cleaning

Main responsibility of the HoppingDuster is not to vacuum, but to wipe or blow down dust. The HoppingDuster is equipped not with vacuum device, but with mop to reduce additional weight. It moves designated path to wipe dust by the mop, or blow away dust by wind generated the HoppingDuster, as shown in Figure 1. Then ground cleaning robot vacuums the dust fallen on the floor.

Figure 1: Cleaning by HoppingDuster

Three ways for migration

HoppingDuster moves along with a migration path determined for each place to be cleaned. It provides three modes for migration: hopping mode where HoppingDuster repeatedly gives small jumps by rotating propellers for a short period of time, hovering mode where it flies low, and flying mode. The flying mode is used to reach places to be cleaned. Remaining two modes are used for cleaning, and have some drawback and advantage. The hopping mode will consume less energy than the hovering mode, because it activates motors for a short period of time. The hovering mode will move faster than the hopping mode, but it may not clean better than the hopping mode.

Self-adaptiveness

HoppingDuster is a self-adaptive system. It tries to finish its cleaning within designated energy consumption and designated time, by changing its behavior. It periodically monitors its progress and status, and selects and switches

an appropriate mode.

Prototype Development

Hardware

We developed a prototype of HoppingDuster on the basis of CrazyFlie. CrazyFlie is an open souce, palm size aerial vehicle, which has four motors and propellers, 3-axis gyros, 3-axis accelerometer, 3-axis magnetometer, an altitude sensor, and on-board low-energy radio to communicate with a server. It can lift a payload of up to 5 - 10 g, and can fly up to 7 minutes. We attached bumpers made of aluminum and mops as shown in Figure 2.

Figure 2: Prototype of HoppingDuster

It can measure its 3-axis acceleration, tilt, direction, altitude by using the build-in sensors. It also can collaborate with external sensors via a server. We use Kinect to localize the HoppingDuster. HoppingDuster can determine its current position from data reported by Kinect.

Demo scenarios : Cleaning on Stairs and Desk

In this demo, we will show two case studies cleaning on stairs and a desk. Figure 3 illustrates migration paths for stairs and a desk. For stairs, HoppingDuster cleans at every stair. It detects walls by using acceleration and direction data, and change direction of migration. Then it flies to the down stair, and detect landing by using acceleration and altitude data. By repeating these steps, it clean the stairs.

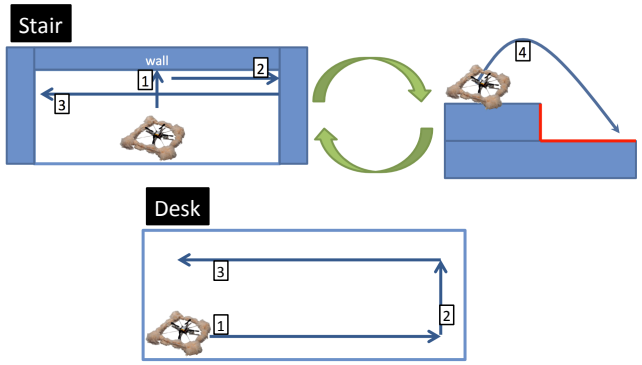

Figure 3: Migration paths for stairs and a desk

For a desk, HoppingDuster cleans a desk by go the circuit of a desk. It detects edges of a desk by using location data obtained by Kinect. In this case, we will also show how self-adaptiveness of HoppingDuster works. The HoppingDuster monitors current progress of the cleaning, execution time, and energy consumptions, and changes migration mode to finish the desk cleaning within designated time.

Related works

There are many sophisticated algorithm to control precisely aerial vehicles to avoid crash [2, 4, 3]. These

algorithm will be useful for HoppingDuster to fly to designated place by avoiding obstacles. However, flight time of aerial cleaning robot should be minimized to improve lifetime of the robot. HoppingDuster provides hopping and hover mode to save energy consumption.

Conclusion

We proposed HoppingDuster, which is a self-adaptive cleaning robot based on an aerial vehicle. HoppingDuster improve capability of cleaning by robots, by collaborating with traditional ground cleaning robots, and shows new application of small aerial vehicles. For the future work, we will integrate sophisticated flying control algorithm into HoppingDuster to enable cleaning of several places in one execution. In addition, we will enable collaborations between ground cleaning robots and HoppingDusters.

References

[1] The Crazyflie Nano Quadcopter. http://www.bitcraze.se/crazyflie/.

[2] Gillula, J. H., Huang, H., Vitus, M. P., and Tomlin, C. J. Design and analysis of hybrid systems, with applications to robotic aerial vehicles. In *ISRR*, C. Pradalier, R. Siegwart, and G. Hirzinger, Eds., vol. 70 of *Springer Tracts in Advanced Robotics*, Springer (2009), 139–149.

[3] Mellinger, D., Michael, N., and Kumar, V. Trajectory generation and control for precise aggressive maneuvers with quadrotors. *The International Journal of Robotics Research 31*, 5 (2012), 664–674.

[4] Purwin, O., and D'Andrea, R. Performing and extending aggressive maneuvers using iterative learning control. *Robot. Auton. Syst. 59*, 1 (Jan. 2011), 1–11.