HAL
open science

A Vision and GPS-Based Real-Time Trajectory Planning for MAV in Unknown Urban Environments

Gerardo Ramon Flores Colunga, Shuting Zhuo, Rogelio Lozano, Pedro Castillo

To cite this version:

Gerardo Ramon Flores Colunga, Shuting Zhuo, Rogelio Lozano, Pedro Castillo. A Vision and GPS-Based Real-Time Trajectory Planning for MAV in Unknown Urban Environments. International Conference on Unmanned Aircraft Systems (ICUAS 2013), May 2013, Atlanta, Georgia, United States. pp.1150 - 1155. hal-00939191

HAL Id: hal-00939191

https://hal.science/hal-00939191v1

Submitted on 30 Jan 2014

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.

2013 International Conference on Unmanned Aircraft Systems (ICUAS)
May 28-31, 2013, Grand Hyatt Atlanta, Atlanta, GA

A Vision and GPS-Based Real-Time Trajectory Planning for MAV in Unknown Urban Environments

Gerardo Flores*, Shuting Zhou*, Rogelio Lozano†, and Pedro Castillo*

Abstract— **This paper addresses the issue of real-time optimal trajectory generation of a micro Air Vehicle (MAV) in unknown urban environments. The MAV is required to navigate from an initial and outdoor position to a final position inside a building. To achieve this objective, we develop a safe path planning method using the information provided by the GPS and a consumer depth camera. With the purpose to develop a safe path planning with obstacle avoidance capabilities, a model predictive control approach is developed, which uses the environment information acquired by the navigation system.**

I. Introduction

Recently, there has been an ever-growing interest on development of micro air vehicles (MAV) due to its capabilities to fly in indoor/outdoor environments. MAVs can be used to explore terrains and acquire visual information in scenarios where it would be impossible for land vehicles. Additionally, they are very suitable for various applications such as surveillance and reconnaissance operations, traffic monitoring, rescue missions in disaster sites, etc., where manned or regular-sized aerial vehicles are not able to accomplish these missions, even with their full operational capabilities.

To accomplish an efficient exploratory navigation in cluttered environments, the MAV must be able to plan and follow three dimensional trajectories avoiding collisions with obstacles and leading through objects. Traditional navigation systems based on the wireless transmitted information, such as Global Positioning System (GPS), is widely used to assure the self-position task. However, most indoor environments remain inaccessible to external positioning systems, limiting the navigation ability of the satellite-based GPS systems.

Vision-based navigation arises as a complementary system for the GPS. Although based on stereo techniques and share many properties with stereo cameras, RGB-D cameras achieve better performance in the spatial density of depth data. Since RGB-D cameras illuminate a scene with a structured light pattern, they can estimate depth in areas with poor visual texture [1]. Thus structured light RGB-D camera is chosen as the vision-based system which plays a supplementary role in the GPS-denied environment. However most RGB-D cameras function in a limited range and cannot achieve a satisfactory navigation when used as the only sensor for long distances. As a result, we combine GPS with an on-board RGB-D camera to provide the MAV with fast and reliable state estimation and collision-free path planning.

There have been previous studies conducted on the MAVs' path planning with avoidance of collision. The Rapidly-exploring Random Tree (RRT) variant is proposed by Yang [2] to generate collision free piecewise paths and linear Model Predictive Control (MPC) is applied to follow this path. Yang has evaluated the robustness of the system by flying over, flying beneath or flying through obstacles - using doors and windows of a building. Rasche and Stern [3] applied the approach based on artificial potential fields and a gradient method to calculate paths, which ensures the multiple UAVs complete a fast exploration of unknown, partially or completely known environments consisting of complex objects. In terms of on-board vision system for determining obstacles and objects, Huang [1] developed a system for visual odometry and mapping applying an RGB-D camera which enables an autonomous flight in cluttered environments using only onboard sensor data. Similarly, Henry [4] presented a RGB-D Mapping system that utilizes a novel joint optimization algorithm to generate dense 3D maps of indoor environment.

In this paper, we require that the MAV accomplishes the task of identifying a window and fly through it, in order to access into a building. The fulfillment of this objective will be quite significant for various military and civil missions of MAVs. In this work, we present a solution to the real-time optimal trajectory generation of a MAV by integrating MPC and vision-based window estimation.

This paper is organized as follows: Section II addresses the problem of real-time trajectory planning. Section III presents the path-planning algorithm and obstacle avoidance method. Simulation results of the proposed path-planning algorithm are presented in Section IV. The vision-based window detection algorithm is introduced in Section V. Finally, Section VI draws a conclusion and gives a perspective on future work of the related research.

II. Problem Statement

Fig.1 shows a three dimensional example of an obstacle avoidance problem. The first goal is to obtain a piecewise continuous function $u(t)$ that drives the MAV from the starting point x_0 to the intermediate state x_i, using the optimal MPC approach. As one can see in Fig.1, the presence of obstacles in an arbitrary urban environment are inherented. Urban elements like lampposts, trees, light poles, power cables and other civil buildings are considered as obstacles

This work is partially supported by the Institute for Science & Technology of Mexico City (ICyTDF).

* are with the Heudiasyc UMR 6599 Laboratory, University of Technology of Compiègne, France. Gerardo Flores (gfloresc@hds.utc.fr), Shuting Zhou (shuting.zhou@hds.utc.fr), Pedro Castillo (castillo@hds.utc.fr)

‡ is with the Heudiasyc UMR 6599 Laboratory, UTC CNRS France and LAFMIA UMI 3175, Cinvestav, México. (rlozano@hds.utc.fr)

for the considered system. Hence, the state $x(t)$ of the dynamical system cannot take arbitrary values. We proceed with the problem statement:

Problem Statement

Find a trajectory that allows the MAV to navigate from an arbitrary point p_0 with coordinates $(x_0,\ y_0,\ z_0)$ to the intermediate point p_i with coordinates (x_T, y_T, z_T, ψ_T), avoiding collisions with n_o obstacles in the environment. Once the MAV has achieved the point p_i, detects the parameters of the target(window model) and minimizes the distance between the centroid of the target and the center of gravity of the MAV, then flies through the window and finally enters the building.

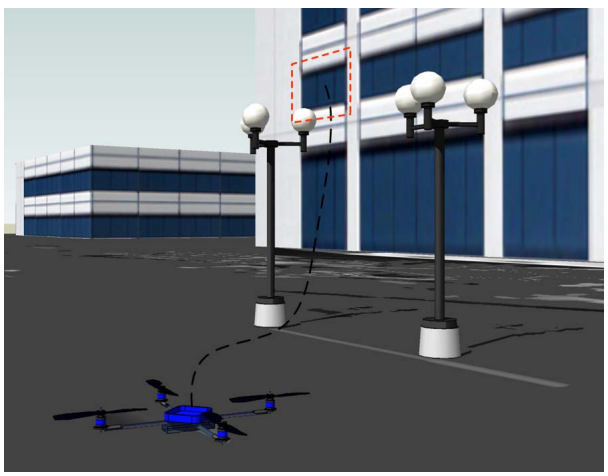


Fig. 1. Scheme of the MAV application. The vehicle should compute a trajectory with obstacle avoidance capabilities using a vision system and GPS information.

A. Optimal control formulation

The equations representing the motion of the MAV can take different forms such as 1) nonlinear fully coupled, 2) nonlinear semi-coupled, 3) nonlinear decoupled, 4) linear coupled, and 5) linear decoupled [5]. Due to the inherent load of any optimization process, the linear decoupled version of the MAV dynamic model is chosen to generate the desired trajectory. The position dynamics of such model can be represented in its discrete form as follows

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{x}(k) + T\mathbf{u}(k) \\ \mathbf{y}(k) &= \mathbf{C}x(k) \end{aligned} \tag{1}$$

where $\mathbf{x} = (x, y, z)$ is the vector representing the MAV position in the inertial frame, $\mathbf{u} = (v_x, v_y, v_z)$ is the input control, $\mathbf{y}$ is the output vector, C is a matrix of appropriate dimensions and T is the sampling time.

It is important to note that the linear and decoupled system (1) will be used by the optimization algorithm only to produce the desired trajectory. The trajectory block containing the path-following controller, will be responsible of following the generated trajectory, as shown in Fig.2.

We assume that the MAV is capable of receiving obstacle information at any path-planning trajectory by means of

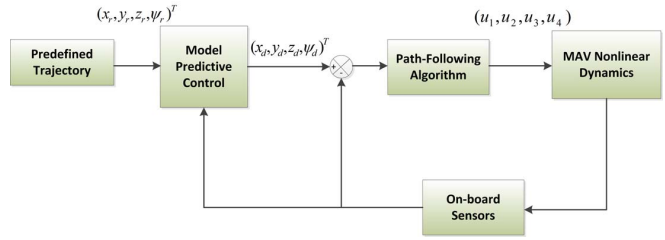


Fig. 2. Control scheme.

onboard sensors. In order to achieve the desired trajectory, we need to solve the following discrete-time optimal control problem.

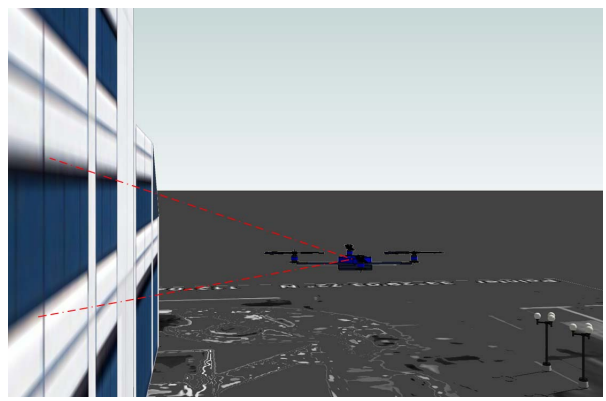


Fig. 3. Window model estimation. In this scenario, the MAV has achieved the intermediate point p_i provided by the GPS system.

B. Optimal trajectory generation using MPC approach

Find the optimal sequence $\{u^*(k)\}_{k=1}^{T-1}$ such that

$$\{u^*(k)\}_{k=1}^{T-1} = \arg\min W(\mathbf{x}, \mathbf{u}, k) \tag{2}$$

where the positive definite cost function $W(\mathbf{x}, \mathbf{u}, k)$ is chosen as

$$W(\mathbf{x}, \mathbf{u}, k) = \Phi\left(\mathbf{x}(T)\right) + \sum_{k=1}^{T-1} \mathfrak{L}\left(\mathbf{x}(k), \mathbf{u}(k)\right) \tag{3}$$

In order to find the optimal control law (2), for system (1), we propose a nonlinear model predictive control approach.

We proceed by starting from the initial state $\mathbf{x}(1)$ and then implementing the optimal input $\{u^*(k)\}_{k=1}^{T-1}$ from $1 \leq \tau \leq T$ to the state $\mathbf{x}(\tau+1)$ at $k = \tau + 1$.

The key idea behind the MPC approach is the combination of the potential field concept with the online optimization with preview. In this way, the function $W(\mathbf{x}, \mathbf{u}, k)$ is minimized by the optimal control sequence (2), i.e.

$$W(\mathbf{x}, \mathbf{u}^*, k) \leq W(\mathbf{x}, \mathbf{u}, k), \qquad \forall u \in U \tag{4}$$

III. Real-Time Trajectory Planning

In this section, we present trajectory generation algorithm for the task described in the previous section.

A. Trajectory Planning

Consider the navigation problem from p_0 to p_i, where p_0 is the initial position of the MAV given by the GPS system, and p_i is a point near to the window to be estimated, given by the user, also provided by the GPS system (Fig.3). In order to achieve the desired position in the presence of obstacles, we proceed by using the procedure developed in Section II-B. We consider an initial reference trajectory given by a straight line, but any other reference trajectory can be used.

Consider a cost function term for (2)-(3) as

$$\begin{aligned} \Phi\left(\mathbf{x}(T)\right) &\triangleq \frac{1}{2}\mathbf{x}^T(T)S\mathbf{x}(T) \qquad (5)\\ \mathfrak{L}\left(\mathbf{x}(k),\mathbf{u}(k)\right) &\triangleq \mathfrak{L}^t\left(\mathbf{x}(k),\mathbf{u}(k)\right)+\mathfrak{L}^o\left(\mathbf{x}(k),\mathbf{u}(k)\right) \end{aligned}$$

where

$$\begin{aligned} \mathfrak{L}^t\left(\mathbf{x}(k),\mathbf{u}(k)\right) = \frac{1}{2}\left(\mathbf{x}_r-\mathbf{x}\right)^T Q\left(\mathbf{x}_r-\mathbf{x}\right) \\ + \frac{1}{2}\mathbf{u}^T R\mathbf{u} \end{aligned} \qquad (6)$$

B. Obstacle Sensing

For collision avoidance, we choose $\mathfrak{L}^o\left(\mathbf{x}(k),\mathbf{u}(k)\right)$ in (5) such that

$$\mathfrak{L}^o\left(\mathbf{x}(k),\mathbf{u}(k)\right) = \sum_{i=1}^{n_o} f(\mathbf{x},\mathbf{x}_i) \qquad (7)$$

and the function $f(\mathbf{x},\mathbf{x}_i)$ is defined as

$$\begin{aligned} f(\mathbf{x},\mathbf{x}_i) = a_x \exp\left(-\frac{(\mathbf{x}(k)-\mathbf{x}_i(k))^2}{2c^2}\right) \\ + a_y \exp\left(-\frac{(\mathbf{y}(k)-\mathbf{y}_i(k))^2}{2c^2}\right) \\ + a_z \exp\left(-\frac{(\mathbf{z}(k)-\mathbf{z}_i(k))^2}{2c^2}\right) \end{aligned} \qquad (8)$$

where $(\mathbf{x}_i,\ \mathbf{y}_i,\ \mathbf{z}_i)$ are the coordinates of the i-th obstacle in the inertial frame. The parameters a_x, a_y, a_y, c are chosen to determine how far the vehicle can be approximated to the obstacle. Thus, the penalty function (8) serves as a repelling field.

IV. Simulation results

In order to test the effectiveness of the derived controller, in this section we will show some simulations results.

The aforementioned optimization procedure will result in the desired trajectory which will be tracked by the path-following algorithm, see Fig.2. First, we apply the optimization procedure for a navigation problem in which a MAV is requested to fly from the initial point $p_0 = (0,0,0)$ to the intermediate point $p_i = (10,10,10)$. We have simulated the presence of an obstacle in the coordinates $(\mathbf{x}_i,\ \mathbf{y}_i,\ \mathbf{z}_i) = (5,5,5)$. This type of situation often arises in an urban area, as can be seen in Fig.1.

The MAV resolves the collision maintaining a safe distance from nearest point of nearby obstacles as it travels. Fig.4 shows the output trajectory. In Fig.5 the velocities generated by the algorithm are depicted. Such velocities and the corresponding positions, are used in the feedback control loop as a reference to the position control loop of the MAV.

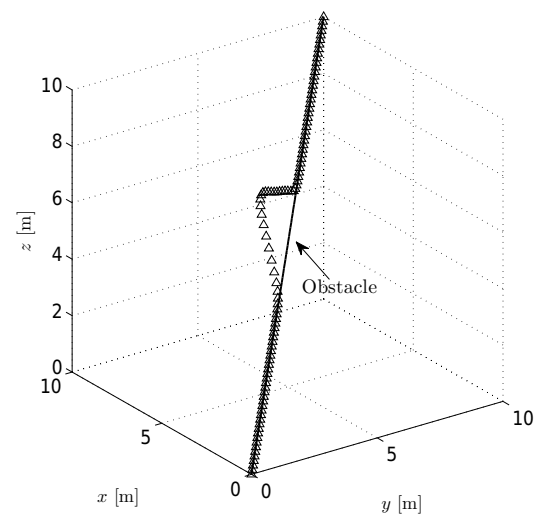


Fig. 4. Computed path. An obstacle is positioned in the coordinate $(5,5,5)$.

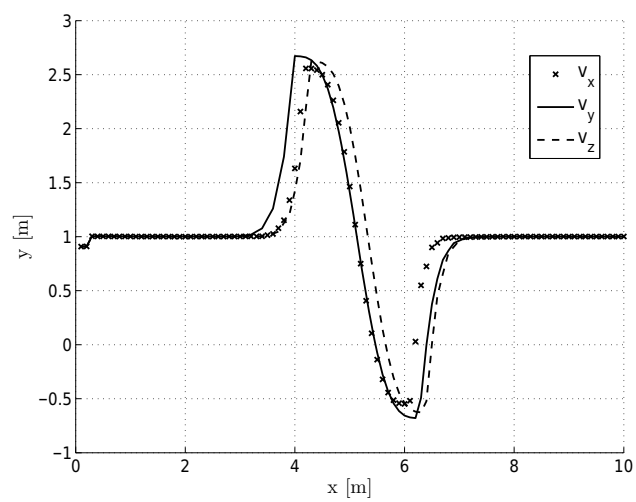


Fig. 5. Velocities generated by the algorithm.

By choosing different values of the parameter c, it can be possible to avoid obstacles from a longer distance as can be seen in Fig.6.

V. Vision-based Window estimation

In order to solve the second part of the problem statement presented in Section II, a vision-based algorithm is proposed in this part of the paper. An RGB-D camera, which will be used by the MAV to acquire the images, can capture RGB color images accompanying with the corresponding depth data at each pixel. Such cameras manage all the information with a set of vertices in a three-dimensional coordinate system called *point cloud*. *Point cloud* is a collection of massive points in the same spatial reference system, which expresses the spatial distribution and the surface properties

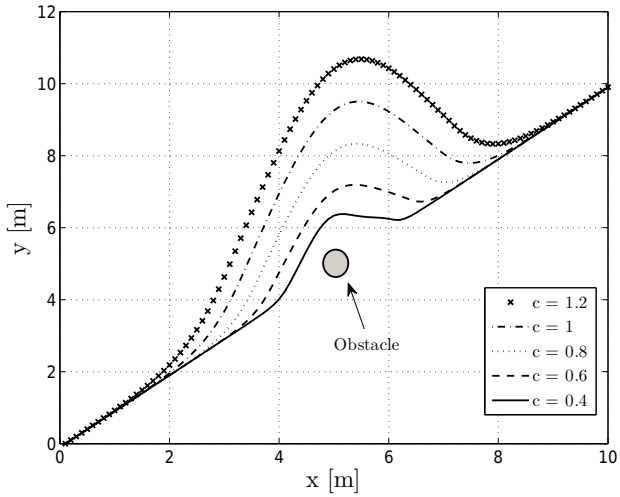


Fig. 6. Trajectories.

Procedure 1 EstimateTargetPlane (PC: input point cloud, n: number of points in PC)

```
PC_d = DownSample(PC)
cloud_filtered = PC_d
i = 0
while size_cloud_filtered >n*8% do
    plane_i = ExtractInliers(cloud_filtered)
    cloud_filtered = cloud_filtered - plane_i
    i++
end while
return plane_1
```

of the target. Three-dimensional coordinates of each sample point on the target surface (x, y, z) and the corresponding color information (RGB) can be acquired from the point cloud.

In hope of identifying the window model and estimate the window centroid precisely and effectively, we take advantage of the programs available in the open source point cloud library [6].

A. Algorithm Description

In order to estimate a model representing a window, a vision algorithm is developed in this section, as it is explained below.

The algorithm executes down-sampling process on the input point cloud to improve the running efficiency. Then the iterative estimation process is performed to extract the inliers of planes existing in the point cloud until the target plane is detected. The approach applied for plane detection is RANdom SAmple Consensus (RANSAC) algorithm. The basic idea of RANSAC is to estimate the plane parameters using the minimum number of data possible (random three points) and then to check which of the remaining data points fit the model estimated [7]. Here in our case two planes are present in the input point cloud, the larger one is the plane of the wall, the smaller one is the plane representing the surface of the window model, which we are interested in. Procedure 1 in Fig.7 describes the target plane identification process.

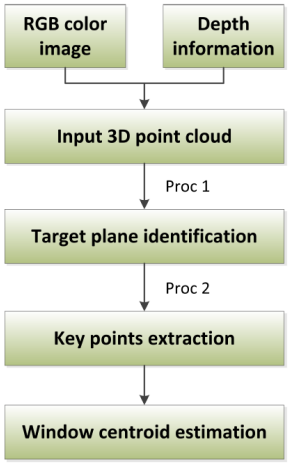


Fig. 7. Window identification process scheme.

Procedure 2 ExtractKeyPoints (PC: point cloud of plane)

```
range_image = CreateRangeImage(PC)
points[] = DetectNarfKeypoints(range_image)
return points[]
```

With the estimated target plane, we continue to extract some key points representing the critical features to estimate the centroid of target surface. Procedure 2 in Fig.7 describes how the key points are extracted from the point cloud of the estimated plane. Here we apply the Normal Aligned Radial Feature method (NARF) for interest point detection and feature descriptor calculation in 3D range data [8]. As NARF method takes information about borders and the surface structure into account, the position of target plane centroid can be estimated by applying the edge information extracted by NARF detection algorithm. Some preliminary results are presented in the following subsection.

B. Experiments

In this subsection, several tests have been carried out just using the RGB-D camera.

To verify the algorithm described previously, a simple hypothesis is proposed to simplify the vision-based estimation problem. By sticking a box on the wall, the surface of the box is viewed as the target window.

Placed opposite to the target surface of the box, the RGB-D camera collects sufficient position information of the target plane, see Fig.8. The execution time of the vision-based window estimation algorithm lasts about 800ms, which can be performed in real-time MAV navigation system.

The results of our first test are shown in Figs.10 and 11. The distance between the RGB-D camera and the box is 0.92m. Fig.9 shows the input 3D point cloud captured by RGB-D camera.

From the figure we can clearly recognize the surfaces of the wall and the target box. Fig.10 shows the plane point cloud detected by the target plane identification algorithm. The target surface of the box is completely extracted from the input point cloud. Fig.11 shows four key points extracted

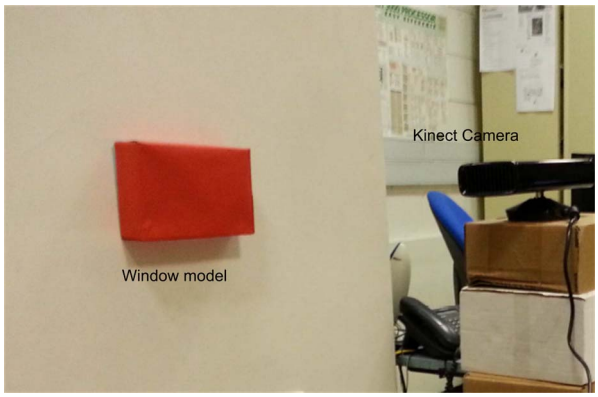


Fig. 8. Experimental setup.

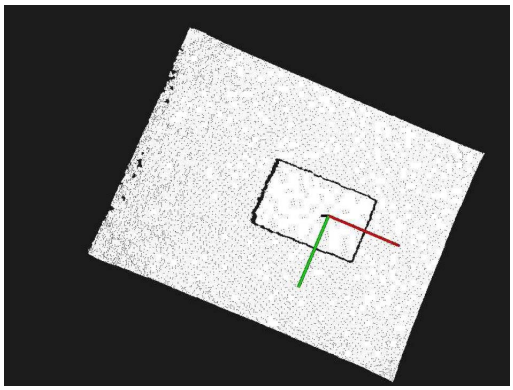


Fig. 9. Input point cloud.

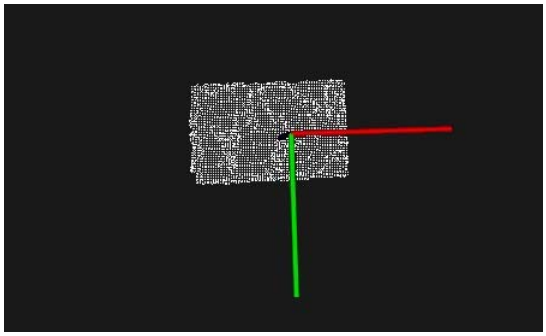


Fig. 10. Detected target plane.

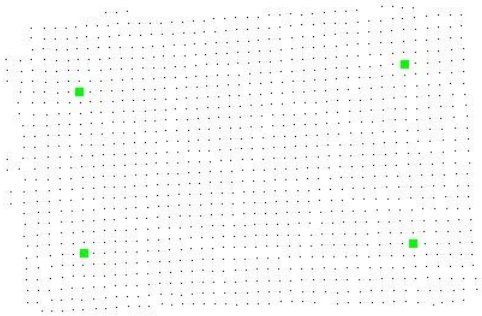
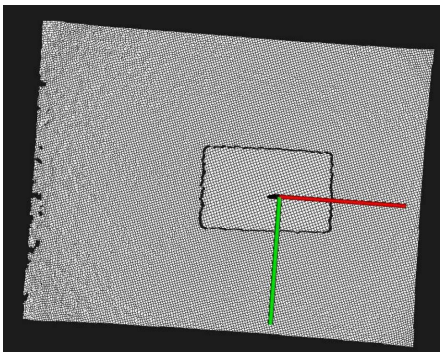
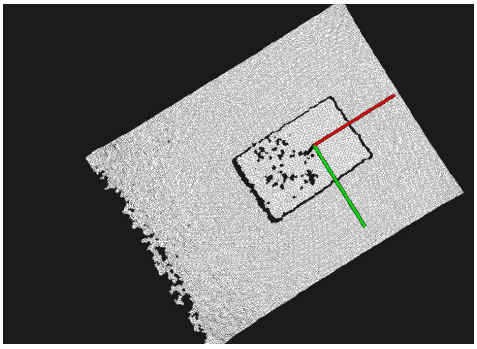


Fig. 11. Extraction of four key points.



(a) distance = 0.81m



(b) distance = 0.59m

Fig. 12. Input point cloud at closer distances.

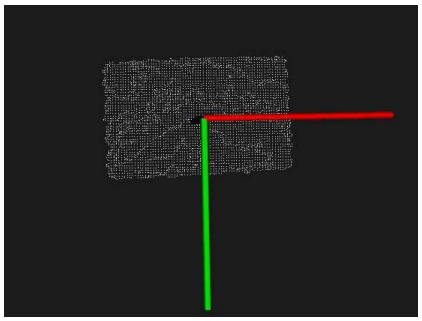
from the range image of target plane. They can be approximately regarded as the four vertices of the target plane. Then the centroid of the target plane can be obtained by intersecting the diagonals of the four key points. By pushing the table that places RGB-D camera towards the target box, the distance between the camera and the box decreases. Figs.12,13 and 14 show the corresponding results.

C. Experimental setup

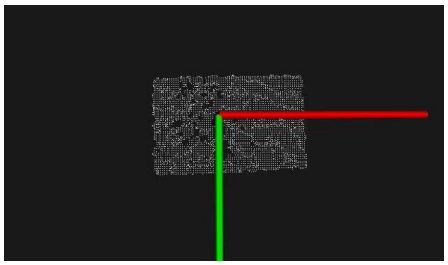
In order to perform the real-time implementation of our strategy, we use a Microsoft Kinect sensor (http://www.xbox.com/kinect, Fig.15). As a low-cost RGB-D sensor developed by PrimeSense, Kinect is equipped with three lenses, the lens in the middle is the RGB color camera, and the lenses in the left and right side are the infrared transmitter and infrared CMOS camera which constitute a 3D structured light depth sensor.

Based on the light coding, Kinect projects a known infrared pattern onto the scene and determines depth based on the pattern's deformation captured by the infrared CMOS imager [9]. Functioning in this way, Kinect can provide a 320X240 depth image at 30fps and a 640X480 RGB color image at 30fps. When stripped down to its essential components, the Kinect weighs 115g - light enough to be carried by a MAV.

Here there is one thing to note is that the function of kinect is affected by the light intensity because of the infrared lens it uses. The kinect always achives better performance in poor

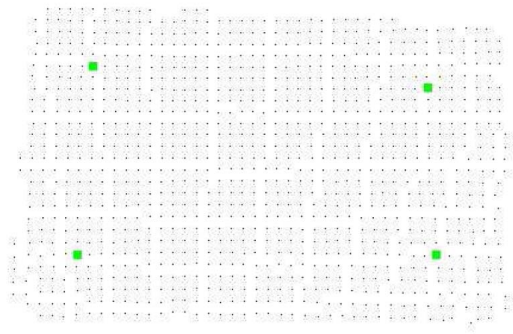


(a) distance = 0.81m

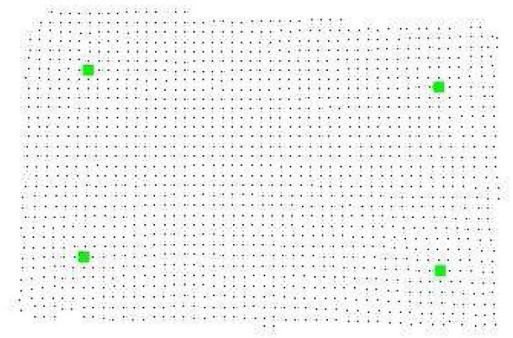


(b) distance = 0.59m

Fig. 13. Target plane at closer distances.



(a) distance = 0.81m



(b) distance = 0.59m

Fig. 14. Key points at closer distances.

Fig. 15. Microsoft Kinect.

light or cloudy days since the infrared pattern produced by the Kinect will not be overridden by the infrared radiation of the sun.

VI. Conclusion

In this work we present a new approach to the real-time optimal trajectory generation of a MAV by integrating MPC and vision-based window estimation algorithm. We have also extended some preliminary experimental results to verify the vision algorithm.

There also exists some limitations in our algorithm which requires further improvements. Firstly, the real window model is more complicated and shows some special properties such as poor reflected information, so we should make our algorithm more robust and adapt to the real environment. Besides, to ensure the MAV enter the Kinect depth sensor range (from 0.8m to 4.0m), it is not sufficient that the GPS serves as a standalone sensor for local positioning, maybe other sensors should be involved to improve the positioning accuracy such as the barometer.

Future work will be developping a more robust vision-based algorithm, establishing the experimental platform and implementing the proposed navigation control strategy for a MAV.

References

[1] A. S. Huang, A. Bachrach, P. Henry, M. Krainin, D. Maturana, D. Fox, and N. Roy, "Visual odometry and mapping for autonomous flight using an rgb-d camera," *Int. Symposium on Robotics Research (ISRR)*, may 2011.

[2] K. Yang, S. Gan, and S. Sukkarieh, "An efficient path planning and control algorithm for ruav's in unknown and cluttered environments," *Journal of Intelligent and Robotic Systems*, vol. 57, no. 1-4, pp. 101–122, 2010.

[3] C. Rasche, C. Stern, L. Kleinjohann, and B. Kleinjohann, "A distributed multi-uav path planning approach for 3d environments," in *Automation, Robotics and Applications (ICARA), 2011 5th International Conference on*, dec. 2011, pp. 7 –12.

[4] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox, "Rgb-d mapping: Using depth cameras for dense 3d modeling of indoor environments," *Journal of Robotics Research*, vol. 31, no. 5, pp. 647–663, 2012.

[5] M. Sadraey and R. Colgren, "Uav flight simulation: Credibility of linear decoupled vs. nonlinear coupled equations of motion," in *AIAA Modeling and Simulation Technologies Conference and Exhibit*, San Francisco, California, Aug. 2005.

[6] R. B. Rusu and S. Cousins, "3d is here: Point cloud library (pcl)," *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9–13, May 2011.

[7] F.M. and R. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM 24*, pp. 381–395, 1981.

[8] B. Steder, R. Rusu, K. Konolige, and W. Burgard, "Narf: 3d range image features for object recognition," 2010.

[9] X. Xiang, Z. Pan, and J. Tong, "Depth camera in computer vision and computer graphics," *Journal of Frontiers of Computer Science and Technology*, 2011.