



HAL
open science

Planificateur Photométrique pour Suivi de Chemin Visuel

Eder Alejandro Rodríguez-Martínez, Guillaume Caron, Claude Pégard, David
Lara-Alabazares

► **To cite this version:**

Eder Alejandro Rodríguez-Martínez, Guillaume Caron, Claude Pégard, David Lara-Alabazares. Planificateur Photométrique pour Suivi de Chemin Visuel. ORASIS 2021, Centre National de la Recherche Scientifique [CNRS], Sep 2021, Saint Ferréol, France. hal-03339788

HAL Id: hal-03339788

<https://hal.science/hal-03339788>

Submitted on 9 Sep 2021

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.

Planificateur Photométrique pour Suivi de Chemin Visuel

Photometric-Planner for Visual Path Following

Eder A. RODRÍGUEZ M., Guillaume CARON¹
Claude PÉGARD

David LARA A.²

¹ Laboratoire Modélisation, Information & Systèmes, Université de Picardie Jules Verne, Amiens, France

² Institut Technologique Supérieur de Misantla

33 rue Saint Leu - 80039 Amiens Cedex 1 - France.
eder_rdz_mtz@hotmail.com

Résumé

La navigation robotique combine la perception, quelques informations de l'environnement et un groupe de poses à atteindre pour commander les déplacements du robot pendant une mission. La navigation basée vision est une instantiation de la navigation robotique qui utilise la perception visuelle pour commander le robot et représenter l'environnement. Cet article présente un système de navigation basé vision embarquée et chemin visuel pour représenter la scène à partir d'un groupe d'images. Bien que le système soit basé mémoire, la navigation est conçue comme un enchaînement de tâches de positionnement par des asservissements visuels. La contribution du travail repose sur la génération des images qui composent le chemin visuel. Ces images sont obtenues à partir d'un modèle texturé préalable de la scène. Les expérimentations montrent la performance du système sur trois scènes différentes, à l'intérieur et à l'extérieur, et en utilisant un drone.

Mots Clef

Navigation basée vision, asservissement visuel direct, robotique aérienne.

Abstract

Robotic navigation is the aspect of cognition related to robot robust mobility. It combines perception, some knowledge of the environment and a set of goal poses to reliably control the robot during a mission that involves displacement. Vision-based autonomous navigation is an instantiation of the latter discipline where visual perception is used to control the robot and to represent the environment. This paper presents a vision-based navigation system that uses its onboard camera to navigate and a visual path that represents the scene with a set of images. Being a memory-based system, the navigation is conceived as a concatenation of positioning tasks in the visual servoing scheme. The novelty on the proposed system relies on the generation of the images that compose the visual path. These im-

ages are rendered from a preobtained model of the scene. The experiments evaluate the performance of the system over three different scenes, contemplating indoor and outdoor environments, and using an Unmanned Aerial Vehicle as testbed.

Keywords

Vision-Based Robotic Navigation, Direct Visual Servoing, Aerial Robotics.

1 Introduction

Vision-based autonomous robot navigation is the process of making the robot being able to reach a final pose, which is out of sensing range from a start pose, using vision for perception. The robot, in order to navigate from the start to the final pose, requires a path, that is usually planned offline, online or both [1]. Once the planned path is known, most of the approaches estimate the robot pose, i.e. localization, while moving, using visual odometry [2] or Simultaneous Localization And Mapping (SLAM) [3]. Then, the estimated current pose is taken as input to control the robot in order to successively follow the intermediate poses that compose the path [4]. The common scheme described above presents three distinct stages: path planning, visual perception and motion control.

The visual perception stage of the autonomous robotic navigation mostly deals with keypoints and geometric features detected and matched in images. Such geometric information can be related to the robot pose, considered in the planning as well as in the control stages. Nevertheless, robust extraction and matching of the features over a large environment is still a bottleneck for the visual navigation schemes [5]. Therefore, an alternative to the geometric information is to consider the direct one. Indeed, a conventional camera captures light and not geometric features. However, due to the difference between the natures of photometry (visual acquisition) and geometry (3D space in which the robot moves), the direct information is less

trivial to be considered for navigation. Recent progresses in direct photometric-based autonomous robot localization systems opened the way to reach similar practical performance than geometry-based approaches, and even better, since detection of geometric features and matching process are avoided [6]. Furthermore, dense methods are more precise for positioning tasks thanks to the visual feature redundancy [5]. Although implicitly, the aforementioned system directly relates two common stages of vision-based navigation: image acquisition and pose estimation (robot attitude and position). Similarly, the system proposed in [7] directly relates the stages of visual perception with motion control. This system uses Photometric Visual Servoing (PVS) [8], which considers the pixel intensities of the image as visual feature, to follow a visual path.

1.1 Problem Statement

On the one hand, the process of building a visual path generally starts with the acquisition of images from a first tele-operated trajectory [9]. This process requires the robot *in situ* and is time consuming. However, the visual path might also be planned from a preobtained dataset of images [10]. In the latter work, the visual path is represented by a weighted graph where each node is associated with an image. Then, the edges are weighted accordingly to the number of feature points matched between the images. Nevertheless, the dataset of images is required to be acquired *in situ* and by the same camera to ensure the point matching. On the other hand, the scene might be completely represented by a model, e.g. a Computer-Aided Design model [11], obtained *a priori*. Nevertheless, some zones may lack of texture. Moreover, localization is required when navigating.

We propose to solve these problems by generating a visual path \mathcal{K} from a preobtained model of the scene. First and foremost, the model is a rectangular mesh \mathcal{M} , subdivided into two triangles (orange dotted lines in Fig. 1). The mesh \mathcal{M} is textured by aligning its 4 vertices with those of an image of the scene, e.g. a satellite or high altitude aerial image. By using an Unmanned Aerial Vehicle (UAV) as testbed, the scene and the mesh \mathcal{M} are considered flat in this paper. Secondly, the visual path \mathcal{K} is composed by key images $K \in \mathcal{K}$ rendered from the mesh \mathcal{M} . The generation of the mesh \mathcal{M} and the rendering of each key image K are computed offline. Lastly, the autonomous navigation is defined as a concatenation of positioning tasks which goal is to sequentially servo each key image K ; called **visual path following**.

The system proposed in this paper is presented next.

1.2 Proposed System

The main contribution in this paper is a vision-based navigation system composed by the **Photometric-Planner** and the **Visual Path Follower**.

The goal of the Photometric-Planner is to generate a navigable visual path \mathcal{K} . Generally speaking, this process begins by rendering virtual images I_V from the mesh \mathcal{M} .

Then a Gaussian filter is applied to these images and a Zero-mean Normalization transformation to their pixel intensities. Finally, some of these I_V are selected; thus, they become the elements in the visual path \mathcal{K} .

The Visual Path Follower is in charge of carrying out the autonomous navigation. Being a memory-based system, the UAV is controlled by comparing the onboard images I with those stored in the visual path \mathcal{K} using a variation of PVS, called Enhanced PVS. Indeed, to servo a desired image I^* using an UAV as testbed is a secondary contribution since this image is virtual (so, denoted as I_V^*).

The diagram in Fig. 1 illustrates the general workflow of the proposed system.

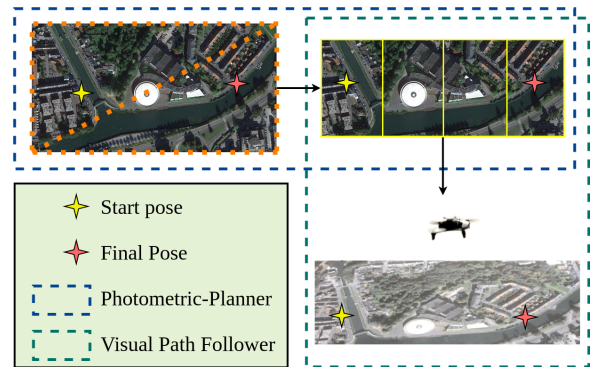


Figure 1: Proposed vision-based navigation system. Once the start and final poses are defined over the mesh (top left), the Photometric-Planner generates the visual path (top right) to be used for the Visual Path Follower.

1.3 Paper Structure

The rest of the paper follows the next structure. Section 2 compares current works in the state-of-the-art of vision-based navigation with ours and Section 3 formally presents the novel Enhanced PVS and highlights two advantages over PVS. Then, Section 4 details the process of the Photometric-Planner and the criterion used to select the key images K among other virtual images I_V . Later, the Visual Path Follower is detailed in Section 5 as well as the criterion to switch from the current key image K to the next one in the visual path \mathcal{K} . The evaluation of the complete system, implemented on the experimental setup, is presented in Section 6. The experimentation is composed by a group of positioning and navigation tasks performed over indoor and outdoor environments using an UAV as testbed. Finally, the paper concludes in Section 7 by summarizing the main ideas, the interpretation of the results of the experiments and by offering new perspectives for future works.

2 Related Works

This Section reviews the works related to the one presented in this paper. The processes of visual perception and motion control used in [7] are similar of those used in the pro-

posed system since a direct visual feature is considered for the path-planning and navigation. In this paper we introduce a novel feature, called Gaussian Zero-mean Normalized (GZN) luminance $\check{\mathbf{I}}$, which is a secondary contribution. GZN luminance $\check{\mathbf{I}}$ is a variation of the luminance feature \mathbf{I} , being the stacking of every pixel intensities of an image. It is better adapted to compare virtual with onboard images I_V and I respectively, and to compute the motion control of the UAV.

In order to accomplish the goal of the Photometric-Planner, the main difficulty is to select the key images K to build the visual path \mathcal{K} . As stated in Section 1.1, a first example to handle this problem is by maximizing common feature points between successive key frames of the visual path [10]. A second one is by means of the visual odometry and SLAM-based approaches, which minimize (relative) pose uncertainty. These approaches had been implemented for the case of ground robot navigation [12], [13] or in the case of a flying robot [14] along the followed path. The last two works are referred as *Planning in Information Space*, in the more general frame of *active perception*. In practice, this approach has the implicit behavior of maximizing the number of feature points acquired along the path, a critical constraint for the success of visual odometry and SLAM. However, the Photometric-Planner addresses the key image selection problem based on the cost C of the Enhanced PVS which directly relates path-planning with motion control since the same visual feature is used.

The main difficulty of the Visual Path Follower is to compare the key images K with onboard images I using only photometry. This problem had been also solved in the literature [7] where a wheeled robot navigates by comparing the onboard images I from an omnidirectional camera with those stored in the visual path K . However, its visual path is obtained from a tele-operated navigation and not from a textured mesh \mathcal{M} . Another similar approach is the active perception that extends the pose uncertainty minimization-based planning to a photometric criterion [15]. In that approach the robot pose uncertainty is computed as a function of the photometric information of the scene, called Perception-aware Path Planning. However, the system proposed in this paper does not require to build nor update a model while navigating, it rather represents the environment to navigate by a minimum number of key images K .

3 Enhanced Photometric Visual Servoing

The luminance feature \mathbf{I} , used in PVS, is a **dense** feature more accurate than a **sparse** one since it stacks pixel intensities of the entire image I to perform a positioning task. However, it is not robust to light variation [5]. Therefore, its implementation to vision-based navigation is restricted to indoor environments only.

Furthermore, the illumination in the textured mesh \mathcal{M} and the scene are unlikely to be the same; thus, PVS is not suitable to compare virtual images I_V with onboard images

I by the Visual Path Follower. Therefore, a Zero-mean Normalization transformation is applied to the image I in order to make the system robust to global affine illumination changes. Additionally, a Gaussian filtering is also considered to prevent undesired behavior when applying the control input \mathbf{u} to the UAV. The proposed system uses this slight variation of PVS, called **Enhanced PVS**.

3.1 Control Law

The Enhanced PVS considers the GZN luminance feature $\check{\mathbf{I}}$ which is obtained from the pixel intensities of the image I after a Gaussian filtering with the following kernel:

$$h(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}, \quad (1)$$

where σ is standard deviation, and a Zero-mean Normalization, becoming:

$$\check{\mathbf{I}}(\xi) = (\check{\mathbf{I}}_{1\bullet}, \check{\mathbf{I}}_{2\bullet}, \dots, \check{\mathbf{I}}_{N\bullet})^\top \quad (2)$$

where $\check{\mathbf{I}}_{i\bullet} \in \mathbb{R}^{1 \times M}$ is the i -th line of the GZN image \check{I} , N is the width and M the height of \check{I} .

Then, the GZN photometric error is defined as:

$$\check{\mathbf{e}}(\xi) = \check{\mathbf{I}}(\xi) - \check{\mathbf{I}}(\xi^*), \quad (3)$$

where $\check{\mathbf{I}}(\xi)$ and $\check{\mathbf{I}}(\xi^*)$ are obtained from the current and desired GZN images \check{I} and \check{I}^* , respectively, at the current pose $\xi = [\mathbf{X}^\top, \Theta^\top]^\top \in \mathbb{R}^6$, where $\mathbf{X} = [X, Y, Z]^\top \in \mathbb{R}^3$ is its location and $\Theta = [\Theta, \Phi, \Psi]^\top \in \mathbb{R}^3$ its orientation, and desired pose ξ^* . Analogously to PVS, the goal of Enhanced PVS is to minimize the value of the cost function:

$$C(\xi) = \frac{1}{2} \|\check{\mathbf{e}}(\xi)\|^2, \quad (4)$$

by non-linear optimization. The chosen control law that handles the motion of the robot is based on the Gauss-Newton method:

$$\mathbf{u} = -\lambda \widehat{\mathbf{L}}_{\check{\mathbf{I}}(\xi^*)}^+ (\check{\mathbf{I}}(\xi) - \check{\mathbf{I}}(\xi^*)), \quad (5)$$

where $\mathbf{u} = \dot{\xi} = [\mathbf{v}^\top, \boldsymbol{\omega}^\top]^\top$ is the control input composed by linear $\mathbf{v} = [v_x, v_y, v_z]^\top$ and angular $\boldsymbol{\omega} = [\omega_x, \omega_y, \omega_z]^\top$ velocities, $\widehat{\mathbf{L}}_{\check{\mathbf{I}}(\xi^*)}^+$ is the Moore-Penrose pseudoinverse of the approximation of the interaction matrix $\mathbf{L}_{\check{\mathbf{I}}(\xi^*)} = -\nabla \check{\mathbf{I}}(\xi^*)^\top \mathbf{L}_x$ related to $\check{\mathbf{I}}(\xi^*)$, $\mathbf{L}_{\check{\mathbf{I}}(\xi^*)}$ is of full rank, $\nabla \check{\mathbf{I}}(\xi^*)$ is the gradient of $\check{\mathbf{I}}(\xi^*)$ and:

$$\mathbf{L}_x = \begin{bmatrix} -1/Z & 0 & x/Z & xy & -(1+x^2) & y \\ 0 & -1/Z & y/Z & 1+y^2 & -xy & -x \end{bmatrix}, \quad (6)$$

is the interaction matrix that relates the variations of the image point $\mathbf{x} = [x, y]^\top$ to the velocity of the camera, as depicted in [16].

Although the Gauss-Newton optimization method is chosen to minimize the cost C , the Visual Path Follower is not restricted to any particular method.

3.2 Cost Function Analysis

An experimental study shows the main differences between PVS and Enhanced PVS. The study consists into computing the cost functions $C(\xi)$ of PVS and Enhanced PVS by rendering virtual images $I_V(\xi)$ over the image plane around the desired pose ξ^* . These images are generated by translating ξ along t_X and t_Y within the intervals of $[-30\text{mm}, 30\text{mm}]$ in the X and Y axes with a sampling step of 0.5mm . Fig. 2 shows four desired virtual images I_V^* used in the experimental study of the cost functions $C(\xi)$. All of these images are rendered from the textured mesh \mathcal{M} at the desired pose ξ^* . Notice that the global illumination difference is drastically reduced after applying a Zero-mean Normalization transformation to the obscured I_V^* with respect to its corresponding non-obscured I_V^* .

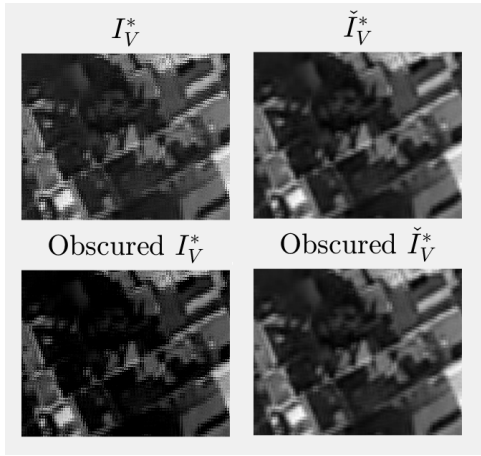


Figure 2: Four virtual images rendered from the textured mesh at the desired pose for the cost functions study.

Two considerations are made for the experimental study:

- All the feature vectors (\mathbf{I} , \mathbf{I}^* , $\check{\mathbf{I}}$ and $\check{\mathbf{I}}^*$) are generated from virtual images I_V . The experimental study uses virtual images I_V since the Photometric-Planner renders I_V from the mesh \mathcal{M} .
- The desired features (\mathbf{I}^* and $\check{\mathbf{I}}^*$) are generated from obscured I_V or \check{I}_V . This consideration simulates the global illumination difference between the onboard image I and the key image K during the autonomous navigation.

The cost functions $C(\xi)$ of PVS, defined as the Sum of Squared Differences (SSD) of pixel intensities, and (4) (GZNSSD) appear in Fig. 3.

Notice that the value at the minimum of GZNSSD cost function $C(\xi)$ reaches zero whereas the corresponding one of SSD does not. Moreover, the shape of the domain of convergence around the global minimum is smoother for GZNSSD which will prevent to compute drastic changes in the input velocity \mathbf{u} . Such comparison justifies the choice of Enhanced PVS over PVS for the Photometric-Planner

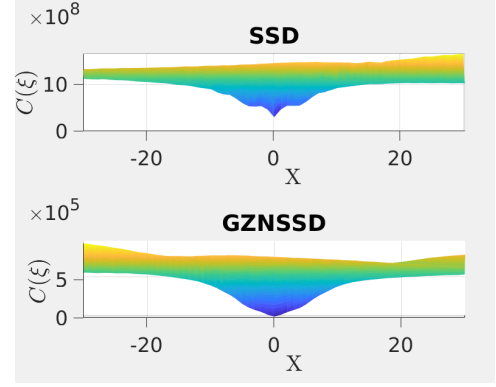


Figure 3: Comparison of SSD (top) and GZNSSD (bottom) cost functions. Distances are shown in millimeters

and Visual Path Follower, which are formally introduced in the next two Sections.

4 Photometric-Planner

The goal of the Photometric-Planner is to generate a navigable visual path \mathcal{K} that allows the Visual Path Follower to reach the final pose ξ_f from the start pose ξ_s . This path contains key images $K = I_V(\xi_k) \in \mathcal{K}$ which are rendered from the textured mesh \mathcal{M} , using any graphics engine, at key poses ξ_k . Although several virtual images I_V are rendered from the mesh \mathcal{M} , just few of them are selected as key images K .

The Photometric-Planner begins by defining the start ξ_s and final pose ξ_f in the textured mesh \mathcal{M} . For the sake of simplicity, ξ_s and ξ_f are considered to have the same orientation Θ . Then, the straight path from ξ_f to ξ_s is regularly sampled in 3D translation steps $\Delta\mathbf{X}$, into intermediate poses ξ_i . Indeed, the same orientation Θ is preserved in ξ_i since $\Delta\mathbf{X}$ does not involve rotation. Later, ξ_f is considered as the first intermediate pose ξ_i and directly selected as the first key pose ξ_k . Thus, its virtual image $I_V(\xi_f) = I_V(\xi_k)$ is stored in the visual path \mathcal{K} as the first key image K . Analogously, the virtual image rendered from the textured mesh \mathcal{M} at the start pose $I_V(\xi_s)$ is considered as the last K in the visual path \mathcal{K} . For the rest of the intermediate poses ξ_i in the path from ξ_f to ξ_s , the selection process is carried out as follows. Let $\xi_{i+1} = \xi_i + \Delta\mathbf{X}$ be the next intermediate pose and ξ_k the last key pose selected by the Photometric-Planner. Then, ξ_i is considered as the next key pose ξ_{k+1} , if the cost increase ΔC_{i+1} falls below a threshold ϵ_s :

$$\Delta C_{i+1} < \epsilon_s, \quad (7)$$

where $\Delta C_{i+1} = C_{i+1} - C_i$, $C_i = \frac{1}{2} \|\check{\mathbf{I}}(\xi_i) - \check{\mathbf{I}}(\xi_k)\|^2$, $C_{i+1} = \frac{1}{2} \|\check{\mathbf{I}}(\xi_{i+1}) - \check{\mathbf{I}}(\xi_k)\|^2$ and the GZN luminance features, $\check{\mathbf{I}}(\xi_i)$ and $\check{\mathbf{I}}(\xi_{i+1})$, are obtained from $\check{I}_V(\xi_i)$ and $\check{I}_V(\xi_{i+1})$, respectively (Section 3). Generally speaking, such **selection criterion** does not depend on the image content but it is rather related to the control law convergence.

5 Visual Path Following

The process of visual path following (Section 1.1) is carried out by the Visual Path Follower, which computes the control input $\mathbf{u} = \dot{\xi}$ from:

$$\mathbf{u} = -\lambda \hat{\mathbf{L}}_{\mathbf{I}(\xi_k)}^+ (\check{\mathbf{I}}(\xi) - \check{\mathbf{I}}(\xi_k)), \quad (8)$$

where $\check{\mathbf{I}}(\xi_k)$ is obtained from the key image $K = I_V(\xi_k)$ in the visual path \mathcal{K} by transforming I_V into \check{I}_V .

When the current key image K is considered reached during the visual path following, the next key image is set as the current one. This process is called **key image update** and it occurs when the value of the cost C (4) satisfies: $\|\check{\mathbf{I}}(\xi) - \check{\mathbf{I}}(\xi_k)\|^2 < \epsilon_u$, where ϵ_u is a threshold. The complete process of visual path following is displayed in Fig. 4.

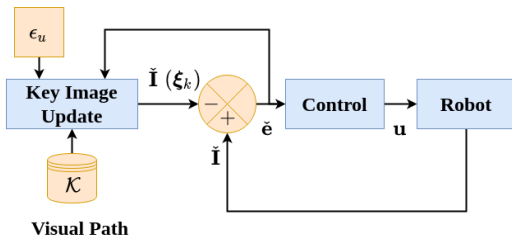


Figure 4: Visual Path Follower

6 Experimentation

The performances of the Photometric-Planner and Visual Path Follower are evaluated by **navigation tasks**. Moreover, since comparing virtual images I_V with onboard images I in a visual servoing scheme using an UAV is a novelty, the performance of the Enhanced PVS is evaluated by **positioning tasks**. Both types of tasks are carried out **indoor** and **outdoor** using the experimental setup which is composed by: an HP ZBook 15 computer as ground station, a Parrot Bebop 2 as testbed UAV, the Optitrack motion capture system (indoor) to track the pose ξ of the UAV, a printed tarpaulin (indoor), a Nikon camera to record external video and three textured meshes \mathcal{M} . In all the indoor tasks, the same satellite image of 2952×3052 pixels is printed on the $4.8\text{m} \times 4.9\text{m}$ tarpaulin and used to generate the textured mesh \mathcal{M} . This image is provided by Google Earth [17] and displays a neighborhood of $1146\text{m} \times 1170\text{m}$ Amiens, France.

The Bebop 2 is connected to the computer, that runs Ubuntu 16.04, through Wi-Fi using the bebop_autonomy package¹ developed for the Robot Operating System [18]. The Photometric-Planner runs offline on the computer using MATLAB and the corresponding textured mesh \mathcal{M} for each navigation task. Indeed, the positioning tasks are an instantiation of the navigation tasks where the visual path \mathcal{K} contains only one key image K . The size of all the images is 240×428 pixels. The Visual Path Follower runs

in the computer and sends the control input \mathbf{u} to the UAV. Generally speaking, the onboard images I are acquired by the camera of the Bebop 2 and sent to the ZBook that computes the control input $\mathbf{u} = \dot{\xi}$ to be applied to the robot. However, the Bebop 2 does not obey velocities in the X and Y -axes (so, v_x and v_y) but angular positions around the Y and X -axes (so, Φ and Θ). Fig. 5. displays the robot and world reference frames ${}^R\mathfrak{F}$ and ${}^W\mathfrak{F}$, respectively.

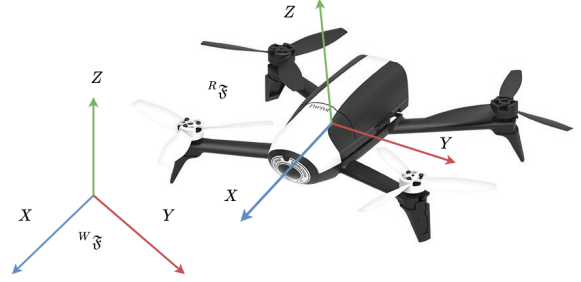


Figure 5: Reference frames of the robot and the world.

Therefore, a proportional controller is proposed to adapt the output of the control law (8) to the input expected by the Bebop 2. Such controller was computed from a velocity characterization of second order and using the dynamic model of the quadrotor [19] that considers air resistance:

$$\begin{bmatrix} {}^W v_x(t) \\ {}^W v_y(t) \end{bmatrix} = \frac{g}{m} \begin{bmatrix} c_0 & 0 \\ 0 & c_1 \end{bmatrix} \begin{bmatrix} \tan({}^W\Phi)(1 - e^{-t\frac{c_0}{m}}) \\ \tan({}^W\Theta)(1 - e^{-t\frac{c_1}{m}}) \end{bmatrix}, \quad (9)$$

where ${}^W v_x$ and ${}^W v_y$ are the linear velocities in the XY plane, ${}^W\Phi$ is the pitch and ${}^W\Theta$ is the roll in the world reference frame ${}^W\mathfrak{F}$, the acceleration caused by the gravity is $g = 9.8\text{m} \cdot \text{s}^{-2}$, the coefficients of the air in both axes are $c_0 = c_1 = 0.35$, $t \in \mathbb{R}$ is the time's variable, and the mass of the quadrotor $m = 0.497\text{kg}$. Nevertheless, such controller is only valid at low speeds. A Proportional-Integral-Derivative controller [20] would be preferred over a Proportional controller. However, it was not possible to design since the readings of the Inertial Measurement Unit (IMU) are retrieved at a lower rate (5Hz) than the control input (30Hz) at **servoing mode**. This issue is related with the update of the Robot Operating System topic that contains the velocity of the Bebop 2 rather than a poor performance of the IMU. This is the main reason to consider the Gaussian filtering in the visual feature.

Furthermore, an image latency, i.e. the time elapsed between an event in the real world and its corresponding change in the image on the laptop side, was measured with a mean of 183.9ms and a standard deviation of 11.26ms. The latency was measured using a mbed² microcontroller that triggers a LED and starts a chronometer at the same time. Then, the chronometer stops when a dark pixel, that corresponds to the center of the LED, increments its intensity up to 200 (with a maximum level of 255). In order to

¹<https://bebop-autonomy.readthedocs.io/>

²Mbed website <https://os.mbed.com/>

mitigate the impact of the latency in the experiments, the robot switches from servoing mode to **hovering mode**, i.e. to apply a null control input $\mathbf{u} = \mathbf{0}$, during 20 iterations at 30Hz when a key image K is reached according to the **key image update** (Section 5). Lastly, the tasks are considered failed when the value of the cost C is larger than a threshold ϵ_f . The thresholds for the navigation tasks are $\epsilon_u = 1.5 \times 10^5$ for the indoor experiment, $\epsilon_u = 5.5 \times 10^4$ for the outdoor experiment and $\epsilon_f = 2.75 \times 10^5$ for both. All the thresholds are experimentally determined. The rest of the Section covers the positioning and navigation tasks.

6.1 Positioning Tasks

Three positioning tasks evaluate the performance of the Enhanced PVS (Section 3). The goal of the robot is to servo a desired virtual image I_V^* , using (8) and real images I acquired with its onboard camera.

Indoor Nominal Positioning. This experiment consists in servoing a desired virtual image I_V^* (Fig. 6) without any external perturbation and at constant illumination. It is important to notice that the native hovering mode drifts and the robot eventually diverges since it is based on visual odometry, which is an open-loop system [21].

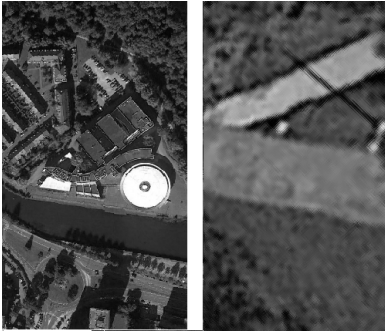


Figure 6: Desired virtual images used for indoor nominal and perturbed positioning (left) and outdoor positioning (right).

While applying the control law (8), the robot minimizes the cost C (4). Then, the experiment finished when the memory that stores the data of the experiment was filled up in about 5 minutes. The evolution of the cost C and control input \mathbf{u} are displayed in Fig. 7.

Indoor Perturbed Positioning. This experiment is similar to the previous one except that a fan is actuated in oscillation mode 2m behind the robot to cause external perturbation.

If the native hovering mode is permanently active, the robot diverges in 90s. However, the Enhanced PVS task lasted 163s when a gust of wind pushed the robot strong enough to diverge.

Outdoor Positioning. This experiment took place on a crossroad (Fig. 8a) during the day but with presence

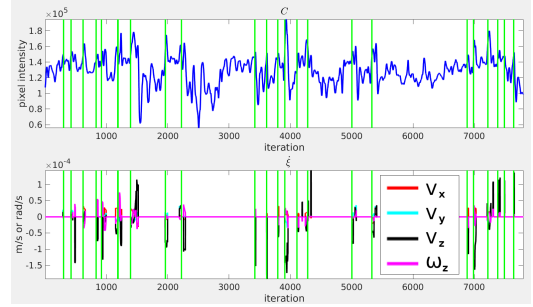


Figure 7: Cost and control input evolution from the indoor nominal positioning. Vertical lines indicate the switch from hovering mode to servoing mode.

of clouds. The desired virtual image I_V^* (Fig. 6) is also generated from the same textured mesh \mathcal{M} as the previous two experiments.

The robot stayed in place, at 10m above the crossroad, without any important perturbation for the entire task with a duration of 240s. Indeed, the robot hovered for the most part of the experiment except at the beginning, when servoing for the first time, and at $t = 100$ s, when a pedestrian caused an increase in the value of the cost C and triggered the servoing mode.



(a) Positioning.

(b) Navigation.

Figure 8: Outdoor experiments.

This group of experiments demonstrated that the servoing is performed at scale (concerning image variation and camera displacement) even if Z is not determined from this scheme. Therefore, perturbations that occur at lower altitude had a greater impact in these tasks since they triggered the servoing mode several times whereas perturbations in the image at higher altitude had less impact. The perturbation caused by the pedestrian showed that the Enhanced PVS is robust to partial occlusions. This experiment was conducted with no Global Navigation Satellite System information for guidance and navigation from ground station. Therefore, the quantitative evaluation of this experiment validates the interest of not relying only on Global Navigation Satellite System for robot positioning and navigation.

6.2 Navigation Tasks

The performance of the Photometric-Planner and Visual Path Follower is evaluated by two navigation tasks. Although any set of straight paths could be considered, e.g. an L-path, the motion capture system field of measure constrains the indoor experiment on a single straight path. A straight path is considered outdoor too for consistency. Therefore, to make the path following harder, these tasks are defined as a **round trip** on a straight line in order to validate the visual path \mathcal{K} in forward and backward directions even if the latter is generated only in backward direction. A video ³ is provided to present the navigation tasks.

Indoor Navigation. In this experiment (Fig. 9), the UAV is intended to travel a straight path of 55cm. The Photometric-Planner generated a visual path \mathcal{K} that contains 11 key images K (Fig. 10).

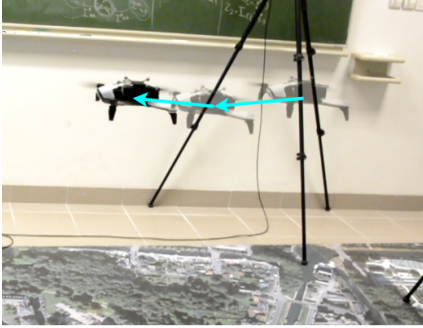


Figure 9: Indoor navigation experiment. The arrows indicate the trajectory done by the robot at the beginning of the second round trip.

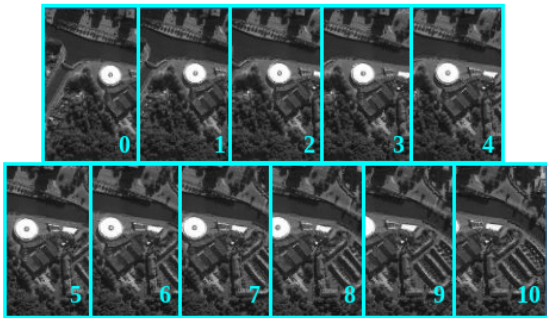


Figure 10: Visual path of the indoor navigation experiment.

Before diverging, the robot traveled approx. 2.7m reaching 54 K : $\{K_0, K_1, \dots, K_{10}\}$, $\{K_{10}, K_9, \dots, K_0\}$, $\{K_0, K_1, \dots, K_{10}\}$, $\{K_{10}, K_9, \dots, K_0\}$, and $\{K_0, K_1, \dots, K_9\}$. Fig. 11 shows the trajectory of the last round trip and Fig. 12 displays the cost C and the control input $\mathbf{u} = \dot{\xi}$ evolution of this experiment. Fig. 13

³http://mis.u-picardie.fr/~g-caron/videos/2020_IIEEESensors.mp4

displays an excerpt of the images of difference between the onboard image I and the key image K .

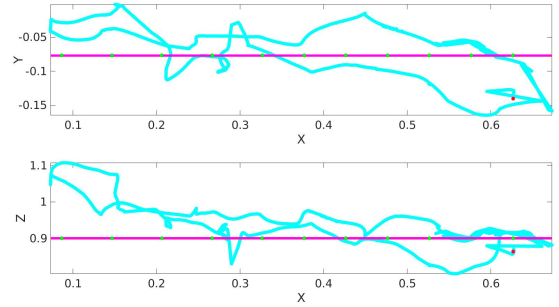


Figure 11: Last round trip (cyan) of the indoor navigation. Green dots indicate the locations of the key poses over the straight path (pink) and red dots where the system failed. Distances appear in meters.

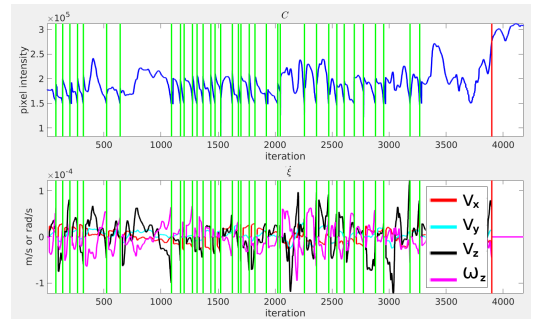


Figure 12: Cost and control input evolution from the indoor navigation. Vertical lines indicate the key image update and red line when the system failed.

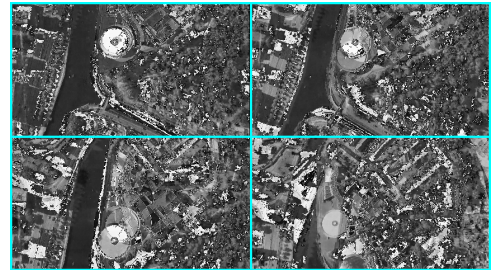


Figure 13: Excerpt of four images of difference of the indoor navigation experiment.

Outdoor Navigation. This experiment took place in the car park of the University of Picardie Jules Verne Campus, Amiens, France (Fig. 8b). The textured mesh \mathcal{M} (Fig. 14) is obtained from a snapshot that the UAV took when flying at 30m above the ground. However, the visual path \mathcal{K} is generated so the robot navigates at a height of 13m. The distance between the start and final poses ξ_s and ξ_f , respectively, is set to 8m. The Photometric-Planner generated a visual path \mathcal{K} composed by 5 key images K . Then, the

robot successfully traveled the round trip, i.e. 16m, using the visual path \mathcal{K} but it diverged when it tried to restart from the first key image K . Fig. 16 shows an excerpt of the images related to this experiment and Fig. 15 displays the evolution of the cost C and the control input $\mathbf{u} = \dot{\xi}$ from this task.



Figure 14: Textured mesh used for the outdoor navigation.

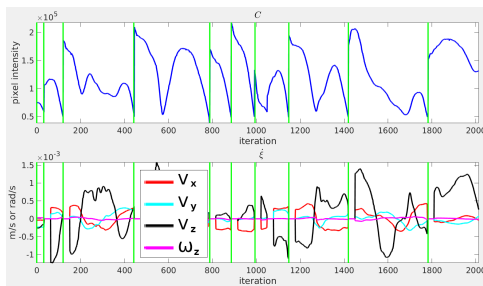


Figure 15: Cost and control input evolution from the outdoor navigation. Vertical lines indicate the key image update.



Figure 16: Excerpt of some images from the outdoor navigation. Desired virtual images appear at the top row, onboard images appear at the central row and images of difference appear at the bottom row.

From this group of experiments, the proposed system completed the navigation tasks in round trip but they diverged when an important perturbation in the image or in the robot occurred.

7 Conclusions

This paper presented a memory-based navigation system that generates its visual path from a textured mesh rather than from a previous taught path. Therefore, the visual path-planning process is carried out *ex situ* and does not require the robot if the textured mesh was obtained *a priori*. Moreover, such mesh can be instantly generated from a snapshot model [22] as in the last experiment.

The Photometric-Planner generated navigable visual paths validated in both directions and the Visual Path Follower controlled an UAV through a total distance of 16m outdoor using only its camera to navigate. In such experiment, navigation based on the Global Navigation Satellite System could not be possible since the signal of the latter was unavailable. Additionally, the Enhanced PVS proved to be able to compare virtual and onboard images while serving a commercial UAV. Nevertheless, the Photometric-Planner only validates the navigability of a visual path defined in straight line and does not explore other visual paths. Future works will consider a topological approach to generate paths not strictly straight. Additionally, they will consider another controller, e.g. Model Predictive Control [23], or UAV so the control input be correctly applied and the Visual Path Follower be evaluated in more challenging situations. As a perspective for indoor navigation, the scene model might include obstacles mapped from a group of robots [24] *prior* to the process of the Photometric-Planner.

Acknowledge

This research was partially supported by the National Council of Science and Technology (CONACYT) of Mexico.

References

- [1] Lu Yuncheng, Xue Zhucun, Xia Gui-Song Xia and Liangpei Zhang, A survey on vision-based UAV navigation, *Geo-spatial Information Science*, vol. 21, no. 1, pp. 21–32, 2018.
- [2] D. Scaramuzza and F. Fraundorfer, Visual Odometry [Tutorial], *IEEE Robotics Automation Magazine*, vol. 18, no. 4, pp. 80–92, 2011.
- [3] G. Bresson and Z. Alsayed and L. Yu and S. Glaser, Simultaneous Localization and Mapping: A Survey of Current Trends in Autonomous Driving, *IEEE Trans. on Intelligent Vehicles*, vol. 2, no. 3, pp. 194–220, 2017,
- [4] P. B. Sujit, S. Saripalli and J. B. Sousa, Unmanned Aerial Vehicle Path Following: A Survey and Analysis of Algorithms for Fixed-Wing Unmanned Aerial Vehicles, *IEEE Control Systems*, vol. 34, no. 1, pp. 42–59, Feb. 2014.
- [5] Raj, Suman and Giordano, Paolo Robuffo and Chaumette, François, Appearance-based indoor navi-

- gation by IBVS using mutual information, *2016 Int. Conf. on Control, Automation, Robotics and Vision*,
- [6] J. Engel and V. Koltun and D. Cremers, Direct Sparse Odometry, *IEEE Trans. on Pattern Anal. Mach. Intell.*, vol. 40. no. 3, pp. 611-625, 2018.
- [7] G. Caron, E. Marchand and E. Mouaddib, Photometric visual servoing for omnidirectional cameras, *Auton Robot* vol. 35, no. 2, 2013, pp. 177-193.
- [8] C. Collewet and E. Marchand, Photometric visual servoing, *IEEE Trans. on Robotics*, vol. 27, no. 4, pp. 828-834, Mar. 2011.
- [9] J. Courbon, Y. Mezouar, N. Guénard and P. Martinet, Vision-based navigation of unmanned aerial vehicles, *Control Engineering Practice*, Vol. 18, no. 7, 2010, pp. 789-799.
- [10] A. Remazeilles, and F. Chaumette, Image-based robot navigation from an image memory, *Robot. Auton. Syst.*, Vol. 55, No. 4, 2007, pp. 345-356.
- [11] Gasteratos, Antonios, et al. "PRONTO: a system for mobile robot navigation via CAD-model guidance." *Microprocessors and Microsystems 26.1* (2002): 17-26.
- [12] K. Otsu, A. A. Agha-Mohammadi and M. Paton, "Where to Look? Predictive Perception With Applications to Planetary Exploration", *IEEE Robot. Autom. Lett.*, vol. 3, no. 2, 2018, pp. 635-642.
- [13] H. Inoue, M. Ono, S. Tamaki and S. Adachi, Active localization for planetary rovers, in *IEEE Aerospace Conf. Big Sky*, MT, 2016, pp. 1-7.
- [14] S. A. Sadat, K. Chutskoff, D. Jungic, J. Wawerla and R. Vaughan, Feature-rich path planning for robust navigation of MAVs with Mono-SLAM, in *IEEE Int. Conf. on Robotics and Automation*, Hong Kong, 2014, pp. 3870-3875.
- [15] G. Costante, J. Delmerico, M. Werlberger, P. Valigi, and D. Scaramuzza. Exploiting Photometric Information for Planning Under Uncertainty, in *Robotics Research. Springer Proceedings in Advanced Robotics vol. 2*.
- [16] F. Chaumette and S. Hutchinson, Visual servo control. Part I. Basic approaches, in *IEEE Robotics Automation Magazine* vol. 13, no. 4, pp. 82-90, 2006.
- [17] Gorelick N., Hancher M., Dixon M., Ilyushchenko S., Thau D. and Moore R., Google Earth Engine: Planetary-scale geospatial analysis for everyone, in *Remote sensing of Environment 2017*. 1;202:18-27.
- [18] Quigley M., Conley K., Gerkey B., Faust J., Foote T., Leibs J., Wheeler R., Ng AY., ROS: an open-source Robot Operating System, in *ICRA workshop on open source software 2009*. 12 (Vol. 3, No. 3.2, p. 5).
- [19] Luukkonen T., Modelling and control of quadcopter, in *Independent research project in applied mathematics*, Espoo. 2011 22;22.
- [20] Salih AL., Moghavvemi M., Mohamed HA., Gaeid KS., Flight PID controller design for a UAV quadrotor, in *Scientific research and essays 2010*. 4;5(23):3660-7.
- [21] Zhang, Ji, and Sanjiv Singh. Visual-lidar odometry and mapping: Low-drift, robust, and fast, in *IEEE Int. Conf. on Robotics and Automation, 2015*.
- [22] Salichs MA, Moreno L., Navigation of mobile robots: open questions. *Robotica*. 2000; 18(3): pp. 227-34.
- [23] Mercorelli, Paolo, et al. A model predictive control in robotino and its implementation using ros system, in *IEEE Int. Conf. on Electrical Systems for Aircraft, Railway, Ship Propulsion and Road Vehicles, 2016*.
- [24] Sergiyenko, O. Yu, et al. Data transferring model determination in robotic group, in *Robotics and Autonomous Systems 2016*.