



HAL
open science

Photometric Path Planning for Vision-Based Navigation

Eder Rodriguez, Guillaume Caron, Claude Pegard, David Lara Alabazares

► **To cite this version:**

Eder Rodriguez, Guillaume Caron, Claude Pegard, David Lara Alabazares. Photometric Path Planning for Vision-Based Navigation. IEEE International Conference on Robotics and Automation, May 2020, Paris, France. hal-02492459

HAL Id: hal-02492459

<https://hal.science/hal-02492459v1>

Submitted on 16 Mar 2020

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.

Photometric Path Planning for Vision-Based Navigation

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

Abstract— We present a vision-based navigation system that uses a visual memory to navigate. Such memory corresponds to a topological map of key images created from moving a virtual camera over a model of the real scene. The advantage of our approach is that it provides a useful insight into the navigability of a visual path without relying on a traditional learning stage. During the navigation stage, the robot is controlled by sequentially comparing the images stored in the memory with the images acquired by the onboard camera.

The evaluation is conducted on a robotic arm equipped with a camera and the model of the environment corresponds to a top view image of an urban scene.

I. INTRODUCTION

Vision-based autonomous navigation is the process of making the robot able to reach an end pose, which is out of sensing range from an initial pose, using vision only. Some representative works are described in [1]–[4].

A vision-based navigation system (VBNS) is able to plan a path and navigate through the scene. Therefore, the representation of the environment plays an important role in this type of system.

Depending on how a VBNS describes the environment, it is classified as a *model-based* or *appearance-based* navigation system [2]. The following paragraphs briefly recall some of the most relevant works regarding these classifications.

In the model-based approach, the environment is described by a model. Generally speaking, this type of system compares the global model (offline) with a local measurement (online) either to localize itself or to navigate.

The work presented in [5] adapts the *Visual Teach & Repeat* (VT&R) methodology [6] to an autonomous *safety-return* aerial navigation system. In the teaching stage, the system performs Visual Odometry (VO) by inserting visual observations into a map of relative poses and scene structure. By comparing the map with the environment, the Unmanned Aerial Vehicle (UAV) localizes itself and reproduces the learnt trajectory in the opposite way. The system presented in [7] updates a map during the exploration stage. Its planner leverages on the geometric and photometric information of the map in order to propose a path that reduces the pose uncertainty while heading towards the goal. The well-known Simultaneous Localization And Mapping (SLAM) [8] technique, merges different types of sensor data to build a map and can also be an input to previously mentioned

methods. When the map is obtained by other means, the exploration stage can be avoided. Such is the case in [9] and [10], where the system compares the visual information acquired during navigation, with a Computer-Aided Design (CAD) model.

The second type is appearance-based navigation which describes the environment in a *topological approach*. The environment is represented by a graph, where the vertices correspond to a measurement of the environment and the edges to the navigability of their corresponding vertices. Two main strategies are used to create such a graph. With the first strategy, an image database completely describes the environment. Then a group of images is selected to form a connected graph [11]–[13]. A localization can be performed relative to the closest vertex on the graph [11]. This method also allows path planning over the graph. With the second strategy, an expert proposes a navigable path and operates a robot through it [14]–[18]. Meanwhile, a sequence of images is collected from which a group is selected to compose the visual memory (\mathcal{K}) of the path, *i.e.* a *visual path*. Then, at the navigation stage, the robot trajectory is controlled through the *key images* ($K \in \mathcal{K}$) on the sensor space, by comparing each K with current visual measurements.

The following statements can be formulated about the aforementioned approaches. The shortest path can only be found when the map fully describes the environment [19]. When the complete model is created during an exploratory stage, it requires the presence of the robot and it is time-consuming. Conversely, when a geometrical model is *a priori* available, features cannot easily nor reliably be matched in some areas [10] and it may lead to navigation failure. Additionally, relying on a geometrical model during the navigation stage may cause error propagation when estimating the pose [13]. Instead, formulating the planning stage as a sequence of Image-Based Visual Servoing (IBVS) tasks avoids such situations. However, the generation of \mathcal{K} that ensures autonomous navigation depends on the skills of the operator at the teaching stage [15]; thus, it is far from being autonomous.

A. System Overview

This paper proposes a VBNS called Photometric-A* (or P-A*, for short) that solves the previously stated problems by combining the advantages of model-based and appearance-based approaches to generate a navigable *visual path*, using only the appearance-based approach to navigate.

Roughly speaking, P-A* focuses on comparing visual information of virtual images rendered from a preobtained

Eder Alejandro Rodríguez Martínez, Guillaume Caron, Claude Pégard are with the MIS laboratory, Université de Picardie Jules Verne, 33 Rue Saint Leu, 80039 Amiens Cedex, France. Emails: {alejandro.rodriguez, guillaume.caron, claude.pegard}@u-picardie.fr.

David Lara Alabazares is with Technological Superior Institute of Misanla, Veracruz, Mexico. Email: dlaraa@itsm.edu.mx

model of the scene with acquired images. Such a virtual/real *hybrid system* features the following characteristics:

- 1) The generation of the visual path does not depend on the skills of the operator but on a topological approach.
- 2) The navigability of the visual path is evaluated using a preobtained model of the scene.
- 3) The navigation mission is defined by a visual path \mathcal{K} made of virtual images.
- 4) During the navigation mission, the images stored in \mathcal{K} are compared with those acquired *in situ*.

The novelty of P-A* lies in generation of the visual memory \mathcal{K} from the preobtained model of the scene, which is our main contribution. Such model is an ortho-image corresponding to a top view image of the scene at scale. From it, virtual images $I_V(V\xi)$ (where $V\xi \in \mathbb{R}^3 \times SO(3)$ is the pose of a virtual camera) are rendered using the inverse perspective camera transformation and the parameters of the camera [20]. The memory is generated in a visual path planning stage by simulating a group of navigation missions in a topological approach (Sec. III). At the navigation stage, the robot is controlled from the start pose (${}^W\xi_s \in \mathbb{R}^3 \times SO(3)$, expressed in world coordinates), to the final pose (${}^W\xi_f$) using \mathcal{K} and the onboard camera (Sec. II-C). As a secondary contribution, the navigation is a concatenation of *hybrid* positioning tasks (Sec. II-C) realized through Photometric Visual Servoing (PVS) [21]. The novelty here is comparing the onboard image $I({}^W\xi)$ with the desired virtual image $I_V(V\xi^*)$ (where $V\xi^*$ is the desired virtual camera pose), rather than using a desired onboard image $I({}^W\xi^*)$ (where ${}^W\xi^*$ is the desired onboard camera pose) as desired image.

B. Related works

The closest VBNS to the P-A* was presented in [13] where a weighted graph is created by matching point features detected on an image database that describes the environment. In their approach, the weight of the edges is determined by the number of matched features between two images in the database, so the graph-search algorithm can propose a more reliable path. By defining the start virtual pose (${}^V\xi_s$) and the final virtual pose (${}^V\xi_f$) at the visual path planning stage (Sec. II-C), P-A* uses Euclidean distances for the weight of the edges so the graph-search algorithm finds the shortest path, as in [22]. Another major difference, with respect to [13], is that P-A* does not need to acquire the image set with the camera used at the navigation stage.

Regarding systems that describe scenes using a textured model, the work proposed in [7] compares the pixel intensities of a *dense surface model* with the onboard images to compute their poses while updating the map. Instead of estimating the camera pose, P-A* computes the camera velocity (${}^W\dot{\xi}$) in order to follow the visual path. The work presented in [23] renders desired depth images from a 3D model of the scene for a positioning task using an RGB-D camera. Nevertheless, its *sensory map* is a sequence of depth maps acquired during manual teaching. Thus, the robot is required

to perform the teaching stage in the navigation scene, contrary to P-A*.

C. Structure of the paper

The remainder of the paper is organized as follows. Section II covers the topics related to the vision-based control exploited by P-A*. Section III describes the generation of the visual path \mathcal{K} . The experiments and their results are presented in Section IV. Finally, Section V discusses the performance of P-A*, its advantages, limitations and future work.

II. EXPLOITING PIXEL INTENSITIES FOR VISION-BASED NAVIGATION

This Section focuses on the motivation, adaptation and implementation of the PVS to perform vision-based navigation in the context of P-A*.

A. Visual Information used in P-A*

Concerning visual information considered within navigation algorithms, feature points have been used in [13] when comparing images acquired by a camera. Nevertheless, this method fails when the desired virtual image $I_V(V\xi^*)$ is used instead of a desired onboard image $I({}^W\xi^*)$. Fig. 1 shows one of the many cases that we tested where an inaccurate matching, between $I_V(V\xi^*)$ and the onboard image $I({}^W\xi)$, occurs using ORB [24] features, even with Random Sample Consensus (RANSAC) [25]. Clearly, the locations of the matched ORB features on both images are imprecise. ORB fails because $I_V(V\xi^*)$ and $I({}^W\xi)$ do not share the same light source; therefore, the performance is degraded [26]. This issue causes an erroneous computation of the relative pose between $V\xi^*$ and ${}^W\xi$; thus, such a method is not suitable within the hybrid context of P-A*.

Therefore, two *dense* and *direct* visual criteria were considered suitable for comparing $I_V(V\xi^*)$ and $I({}^W\xi)$: the Sum of Squared Differences (SSD) between pixel intensities and Mutual Information (MI) [27]. Regarding the virtual/real hybrid context, SSD exhibits three important advantages over MI, when a *Zero-mean Normalization* (ZN) is applied to pixel intensities of the images to be compared; thus leading to a ZNSSD.

The first reason is related to the shape of the cost function (C) of ZNSSD of pixel intensities. This cost function is more

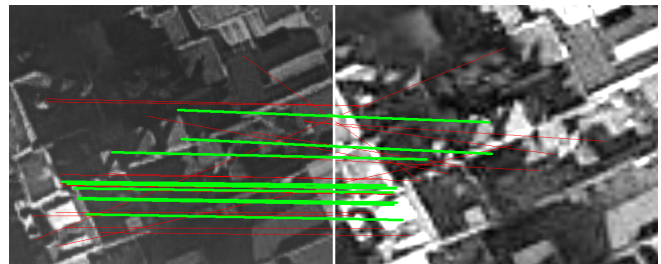


Fig. 1. Imprecise matching. The image on the right is the onboard image whereas the image on the left is the virtual image (contrast enhanced for display). Relative point locations do not correspond between the two images.

convex and its minimum is more pronounced than the corresponding maximum of MI [28]. These characteristics cause a more rapid convergence to the optimum while applying the optimization process; thus reducing the computational cost at the visual path planning stage (Sec. III).

The second reason is that the domain of convergence (\mathcal{C}), of the ZNSSD is larger than the corresponding one of MI. Such a statement is supported by the experiments carried out in [29] between MI and the Zero-mean Normalized Cross Correlation (ZNCC), and the equivalence between ZNCC and ZNSSD highlighted by [30]. The wider the convergence domain, the weaker the number of vertices in the topological graph of poses for visual path planning (Sec. III-B).

Thirdly, it is true that considering the ZNSSD rather than the SSD in the control law for visual servoing [21] slightly increases the computational cost, due to the zero-mean normalization of pixel intensities before computing the Extended Photometric Visual Servoing (Extended PVS) control law (Sec. 5(d)). However, the computational cost remains far lower than in MI-based visual servoing.

Finally, to conclude this subsection, by aligning every corresponding pixel of images, the Extended PVS is far more precise at convergence than an ORB-based visual servoing would be, with its rough matching in the hybrid context of this work (Fig. 1).

B. Extended Photometric Visual Servoing

This scheme considers the vector $\mathbf{I}(\xi)$ of image intensities $I(\mathbf{x})$, acquired at the pose ξ , for all $\mathbf{x} = (x, y)$ belonging to the image domain:

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

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

The Extended PVS is formulated as an optimization process where its goal is to minimize:

$$C(\xi) = \frac{1}{2} \|\tilde{\mathbf{I}}(\xi) - \tilde{\mathbf{I}}(\xi^*)\|^2, \quad (2)$$

where $\tilde{\mathbf{I}}(\xi)$ and $\tilde{\mathbf{I}}(\xi^*)$, are the Zero-mean Normalized current and, respectively, desired image intensity vectors and ξ^* is the desired pose. Using the Gauss-Newton optimization method, the control law of the Extended PVS is:

$$\dot{\xi} = [v_X, v_Y, v_Z, \omega_X, \omega_Y, \omega_Z]^\top = -\lambda \hat{\mathbf{L}}_{\tilde{\mathbf{I}}(\xi^*)}^+ (\tilde{\mathbf{I}}(\xi) - \tilde{\mathbf{I}}(\xi^*)), \quad (3)$$

where λ is a positive scalar, $\hat{\mathbf{L}}_{\tilde{\mathbf{I}}(\xi^*)}^+$ is the Moore-Penrose pseudoinverse of the approximation of the interaction matrix $\mathbf{L}_{\tilde{\mathbf{I}}(\xi^*)} = -\nabla \tilde{\mathbf{I}}(\xi^*)^\top \mathbf{L}_x$ related to $\tilde{\mathbf{I}}(\xi^*)$. $\nabla \tilde{\mathbf{I}}(\xi^*)$ is the gradient of $\tilde{\mathbf{I}}(\xi^*)$ at location \mathbf{x} and:

$$\mathbf{L}_x = \begin{bmatrix} -1/Z & 0 & x/Z & xy & -(1+x^2) & y \\ 0 & -1/Z & y/Z & 1+y^2 & -xy & -x \end{bmatrix}, \quad (4)$$

is the interaction matrix that relates the variations of \mathbf{x} to the velocity of the camera. The interested reader may refer

to [21], [31] and [32] for a more detailed mathematical development.

For P-A*, the Extended PVS control law (Eq. 3) is either applied to current and desired virtual images (Section III) or to current onboard and desired virtual images when following the visual path. Such a path is defined by the key virtual images generated by P-A*.

C. Hybrid Visual Path Following

The robot must follow the ordered set of key images \mathcal{K} describing virtual camera poses ${}^V\xi_k \in \Xi_{\mathcal{K}}$. Thus, to reach the k -th key image $K_k = I_V({}^V\xi_k)$, $\tilde{\mathbf{I}}_V({}^V\xi_k)$ and $\tilde{\mathbf{I}}({}^W\xi)$ are considered in the Extended PVS control law (Eq. 3), instead of $\tilde{\mathbf{I}}(\xi^*)$ and $\tilde{\mathbf{I}}(\xi)$, respectively. The navigation is initialized by roughly placing the camera in such a way that $I({}^W\xi) \approx K_0$. At the navigation stage, P-A* is considered to have converged to K_k when $\|{}^W\dot{\xi}\| < \epsilon_s$, where ϵ_s is a threshold of the norm of the speed and ${}^W\dot{\xi}$ is the output of the hybrid control law (Eq. 3). When the robot reaches the last ${}^V\xi_{|\mathcal{K}|-1}$, the navigation is completed, i.e. the robot has reached ${}^W\xi_f$.

III. VISUAL PATH PLANNING

The goal of the visual path planning stage of P-A* is to find a navigable visual path, i.e. a set of poses to pass through, each of them defined by an image stored in \mathcal{K} . The visual path \mathcal{K} is generated by means of a group of simulated navigation missions which are executed in order to validate the visual path meant to be followed at the navigation stage. A simulated navigation mission, proposed by the planning algorithm, is a concatenation of virtual positioning tasks (Sec. III-D) whose goal is to successively reach each virtual pose (${}^V\xi_p$) in an ordered set of proposed virtual poses (Ξ_P). If Ξ_P allows the simulated navigation mission to be completed, then $\Xi_K = \Xi_P$ and virtual images are rendered at each pose of Ξ_K to make \mathcal{K} .

Ξ_P is selected by a graph-search algorithm (Sec. III-C) from a set Ξ_S of virtual poses tessellating the *virtual search space*. For the sake of simplicity, the latter space is defined in \mathbb{R}^2 for the rest of the article, but considering control laws exploiting either 2, 4 or 6 Degrees of Freedom (DoF) for generality.

The initial tessellation density depends on the extent of the *hybrid* domain of convergence (${}^H\mathcal{C}$). The latter is determined from *hybrid* positioning tasks where Zero-mean Normalized intensities of the desired virtual image $\tilde{\mathbf{I}}_V({}^V\xi^*)$ are used instead of $\mathbf{I}(\xi^*)$ in the Extended PVS control law (Eq. 3).

The rest of the Section formally describes the determination of the Extended PVS convergence domain, the creation of the graph \mathcal{G} of poses representing the search space, the graph-search algorithm, the execution of the virtual positioning tasks and, finally the algorithm that generates \mathcal{K} .

A. Borders of the Hybrid Domain of Convergence

Analogously to the determination of the borders of the domain of convergence \mathcal{C} of any kernel-based method [33],

the borders of $^H\mathcal{C}$ are experimentally determined from *hybrid* positioning tasks. In these tasks, the onboard camera velocity $^W\xi$ is computed from Eq. 3 with $\tilde{\mathbf{I}}(\xi) = \tilde{\mathbf{I}}(^W\xi)$ and $\tilde{\mathbf{I}}(\xi^*) = \tilde{\mathbf{I}}_V(^V\xi_c)$, where $^V\xi_c = (^V\xi_s + ^V\xi_f)/2$ is the central virtual camera pose. The latter $^V\xi_c$ corresponds to the central onboard camera pose $^W\xi_c = (^W\xi_s + ^W\xi_f)/2$.

Then, for each direction $i \in \llbracket 0, N_i - 1 \rrbracket$ regularly surrounding $^W\xi_c$, hybrid positioning tasks (called hybrid positioning test) are executed using the same DoF as those used at the navigation stage. Such a test consists of running Extended PVS from initial poses ($^W\xi_i$) farther and farther from $^W\xi_c$, with step $\Delta \in \mathbb{R}_+^*$, along direction i , until it fails to converge. Let $^W\mathbf{X}(t) = (^WX(t), ^WY(t), ^WZ(t))^T$ be the location and $^W\Theta(t) = (^W\Theta(t), ^W\Phi(t), ^W\Psi(t))^T$ the orientation of $^W\xi$ at the time step t . $^W\mathbf{X}_c$ and $^W\Theta_c$ are equivalently defined for $^W\xi_c$. Then, the goal of the task is achieved if $\| ^W\mathbf{X}(t) - ^W\mathbf{X}_c \| < \epsilon_d$, $| ^W\Theta(t) - ^W\Theta_c | < \epsilon_o$, $| ^W\Psi(t) - ^W\Psi_c | < \epsilon_o$, $| ^W\Phi(t) - ^W\Phi_c | < \epsilon_o$ and $t < \epsilon_k$, where ϵ_d , ϵ_o and ϵ_k are thresholds on the distance, on the orientation difference and on the number of time steps, respectively. These conditions determine if $^W\xi(t)$ is close enough to $^W\xi_c$ before a certain number of time steps.

Poses $^W\xi_i$ furthest from $^W\xi_c$, in every direction, determine the borders of $^H\mathcal{C}$. Distances Hd_i , from $^W\xi_c$ to the borders of $^H\mathcal{C}$, are computed for every direction i by $^Hd_i = \| ^W\mathbf{X}_i - ^W\mathbf{X}_c \|$, $^W\mathbf{X}_i$ being the position of the furthest pose $^W\xi_i$ from $^W\xi_c$. Gathering distances Hd_i in set $^H\mathcal{D}$, $^Hd_{min}$ is defined as the minimum element of $^H\mathcal{D}$.

$^Hd_{min}$ is then considered to define the density of the graph of poses used for planning hereafter.

B. Graph of Poses Construction

The virtual search space (\mathbb{R}^2 in this paper) is uniformly tessellated into the set of poses Ξ_S so that the distance between each neighboring pair of poses is $^Hd_{min}$, at most. Ξ_S is represented as a weighted kinggraph \mathcal{G} [34] of which each vertex is connected to its eight nearest neighbors (leading to $N_i = 8$, Sec. III-A), except for those at the boundaries. \mathcal{G} is also a lattice graph of which the vertices set \mathcal{V} depends on $^Hd_{min}$. Considering $^V\mathbf{X}_f$ and $^V\mathbf{X}_s$ are the locations of $^V\xi_f$ and $^V\xi_s$, we define:

$$n = \left\lceil \frac{\| ^V\mathbf{X}_f - ^V\mathbf{X}_s \|}{^Hd_{min}} \right\rceil, \quad (5)$$

leading to $|\mathcal{V}| = (n+1)^2$.

As a 2D lattice graph, each pose $^V\xi_v$ associated with vertex $v \in \mathcal{V}$ belongs to Ξ_S . Start $^V\xi_s$ and final $^V\xi_f$ poses are associated with, respectively, vertices $v_s = v_0$ and $v_f = v_{(n+1)^2-1}$. Fig. 2 shows \mathcal{G} initialized with $|\mathcal{V}| = 9$. Considering vertices v and v' , $v \neq v'$, associated with locations $^V\mathbf{X}_v$ and $^V\mathbf{X}_{v'}$, the weight function $\gamma(\cdot)$ of their edge $e_{v,v'}$ is defined as the Euclidean distance: $\gamma(e_{v,v'}) = \| ^V\mathbf{X}_v - ^V\mathbf{X}_{v'} \|$.

C. Path Search

The graph-search algorithms A* [35] and Dijkstra [36], have been frequently implemented on VBNS such as [11],

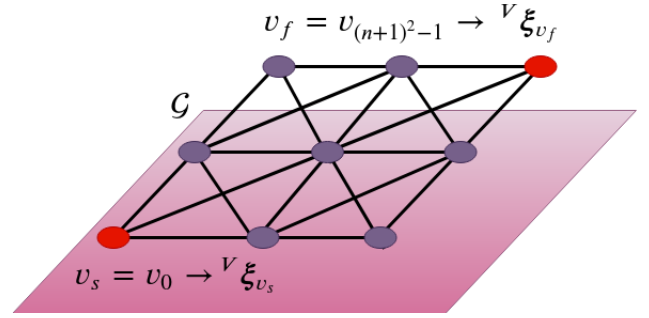


Fig. 2. Graph initialization. Each vertex of the graph is related to a virtual camera pose.

[12] and [17]. A* is preferred for P-A* due to the drastic reduction of the vertex expansion at the graph-search, related to the predefinition of v_s and v_f in \mathcal{G} , as stated in [22].

Then, A* selects the set \mathcal{P}^* of *proposed* vertices $v_p \in \mathcal{P}^*$ in graph \mathcal{G} using the weight function as the *heuristic*. In \mathcal{P}^* , $v_{p_0} = v_s$, $v_{p_{m-1}} = v_f$ with $|\mathcal{P}^*| = m$. Since a pose is associated with each vertex of \mathcal{G} , each proposed \mathcal{P}^* directly leads to an ordered set $\Xi_{\mathcal{P}^*}$ of poses defining a proposed path to be followed.

D. Virtual Positioning Task

In order to evaluate the navigability of a proposed path $\Xi_{\mathcal{P}^*}$, a group of successive virtual positioning tasks is applied for virtual visual path following (similar to Sec. II-C).

A virtual positioning task consists of reaching $^V\xi_{p+1} \in \Xi_{\mathcal{P}^*}$, from an initial $^V\xi(0) = ^V\xi_p \in \Xi_{\mathcal{P}^*}$, by updating $^V\xi$ with the *virtual* control law:

$$^V\dot{\xi} = -\lambda \hat{\mathbf{L}}_{\tilde{\mathbf{I}}_V(^V\xi_{p+1})}^+ (\tilde{\mathbf{I}}_V(^V\xi) - \tilde{\mathbf{I}}_V(^V\xi_{p+1})). \quad (6)$$

Convergence success is evaluated as in Section III-A, *i.e.* by precision tolerance and iteration number thresholds.

E. Visual Memory Generation

The generation of \mathcal{K} is summed up below:

- 1) Create \mathcal{G} as a weighted kinggraph of poses with $|\mathcal{V}| = (n+1)^2$ (Sec. III-B).
- 2) Check if any chain in \mathcal{G} connects v_s with v_f . If so, proceed to the next step. Otherwise, increment n by one and return to step 1.
- 3) Run A* in \mathcal{G} (Sec. III-C). Execute virtual positioning tasks (Sec. III-D) between successive pairs of poses $\{^V\xi_p, ^V\xi_{p+1}\} \subset \Xi_{\mathcal{P}^*}$. If one of the tasks fails, disconnect the corresponding edge $e_{v_p, v_{p+1}}$ in \mathcal{G} and return to step 2. Otherwise, continue to the next step.
- 4) At this step, $\Xi_{\mathcal{P}^*}$ had produced a successful simulated navigation mission. Thus, $\Xi_{\mathcal{K}} = \Xi_{\mathcal{P}^*}$ and render each $K_k = \tilde{\mathbf{I}}_V(^V\xi_k)$, $^V\xi_k \in \Xi_{\mathcal{K}}$. Finally, enqueue every K_k in \mathcal{K} , the visual path made of virtual images to be followed by the real camera at the navigation stage.

IV. EXPERIMENTATIONS

The evaluation of P-A* is conducted on a Stäubli TX2-60 robotic arm equipped with an uEye camera at the end-effector. The robotic arm is considered as a testbed to obtain precise ground truth (poses and trajectories) for a relevant analysis. The robot and the camera (512×640 pixels) are connected to an HP ZBook laptop by TCP/IP and USB, respectively. The distance between ${}^W\mathbf{X}_s$ and ${}^W\mathbf{X}_f$, positions of start and final poses ${}^W\xi_s$ and, respectively, ${}^W\xi_f$, over the scene is 495 mm. This length is bounded by the range of the robot. Hence, the camera is set close to the scene, *i.e.* at 209 mm, and uses a lens with a focal length of 13 mm, so that the camera field of views at ${}^W\mathbf{X}_s$ and ${}^W\mathbf{X}_f$ are far from sharing any part of the scene. The visual path planning and the navigation algorithms are developed in C++. An aerial image, provided by Google Earth, was printed on a tarpaulin and used as the model of the scene. Fig. 3(a) shows the experimentation setup.

In this Section convergence thresholds (Sec. III-A) are experimentally set once for every convergence domain and planning experiment: $\epsilon_d = 0.5$ mm, $\epsilon_k = 500$ iterations, $\epsilon_o = 1.74 \times 10^{-6}$ rad.

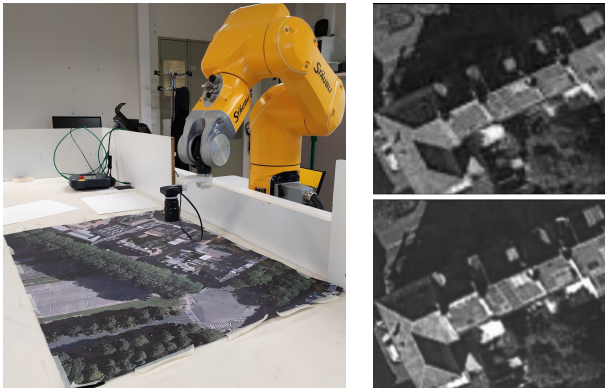
In the following, convergence domains are first determined experimentally, for several configurations and sets of DoF, in order to be able to tessellate the search space and build the graph of poses. Considered sets of DoF are v_X, v_Y (2 DoF), v_X, v_Y, v_Z and ω_Z (4 DoF) and the full camera 6 DoF (ξ , Eq. 3). Then, eight experiments of P-A* are reported in order to evaluate its behavior.

A. Experimental Determination of Convergence Domain

Three configurations are considered for determination and comparison of the Extended PVS convergence domains:

1) *Real Configuration*: The velocity of the onboard camera ${}^W\dot{\xi}$ is computed from Eq. 3 with $\tilde{\mathbf{I}}(\xi^*) = \tilde{\mathbf{I}}({}^W\xi_c)$.

2) *Hybrid Configuration*: This is the hybrid extended PVS (as in Sec. II-C).



(a) Environment set up.

(b) Central images.

Fig. 3. (a) The scene, the robotic arm and the camera. (b) The virtual (top) and the onboard (bottom) images.

3) *Virtual Configuration*: Obviously, this configuration does not need to be performed either by the robot or the camera. The control law is computed from Eq. 6 with $\tilde{\mathbf{I}}_V({}^V\xi_c)$ instead of $\tilde{\mathbf{I}}_V({}^V\xi_{p+1})$.

Fig. 3(b) displays the central pose images used as desired image to determine the convergence domain. Table I shows the distances from the center location (\mathbf{X}_c) to the borders of each domain of convergence \mathcal{C} in eight directions spanning the 360° surrounding \mathbf{X}_c and with increment $\Delta = 2$ mm (see Sec. III-A).

The borders of \mathcal{C} , in each direction, are labeled counter-clockwise beginning from $i = 0$ at the middle-right location of the central pose (ξ_c) (Fig. 4).

TABLE I
CONVERGENCE DOMAIN LIMITS EVALUATION (UNIT: MM)

Direction	0	1	2	3	4	5	6	7	min
Real 2 DoF	16	14	22	46	20	18	24	30	14
Hybrid 2 DoF	16	14	18	42	24	20	28	36	14
Virtual 2 DoF	26	18	28	48	22	18	26	32	18
Virtual 4 DoF	16	18	22	16	16	18	22	16	16
Virtual 6 DoF	18	20	22	16	16	20	24	18	16

B. Navigation Tasks

The eight navigation tasks consider the same convergence threshold (Sec. II-C) $\epsilon_s = 10^{-8}$, experimentally determined.

1) *Virtual versus Hybrid convergence domain limits*: The last column of Table I shows the shortest extents of the Extended PVS convergence domain in each real, hybrid and virtual configuration. Theoretically, ${}^H d_{min}$ must be considered to tessellate the search space and build the graph of poses (Sec. III-B) for P-A*. To limit the experimental time, ${}^H d_{min}$ for 4 DoF and 6 DoF, noted ${}^H_4 d_{min}$ and ${}^H_6 d_{min}$, are deduced from both ${}^V_4 d_{min} = 16$ mm, respectively ${}^V_6 d_{min} = 16$ mm, the equivalent for the virtual configuration, and the ratio ${}^H_2 d_{min}/{}^V_2 d_{min} = 14/18$, made of hybrid and virtual d_{min} with 2 DoF, *i.e.*:

$$\begin{cases} {}^H_4 d_{min} \approx {}^V_4 d_{min} \frac{{}^H_2 d_{min}}{{}^V_2 d_{min}} \approx 12.44 \text{ mm} \\ {}^H_6 d_{min} \approx {}^V_6 d_{min} \frac{{}^H_2 d_{min}}{{}^V_2 d_{min}} \approx 12.44 \text{ mm} \end{cases} \quad (7)$$

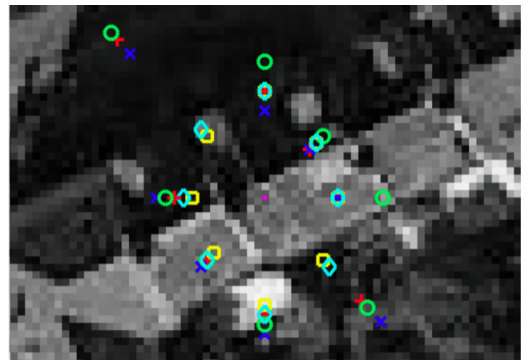


Fig. 4. Domain of convergence around the center (●) of the model. Borders of the domain of convergence for the real (+), hybrid (x), both with 2 DoF, and those of the virtual case with 2 (o), 4 (□) and 6 DoF (◇).

Then, P-A* is executed six times between the same ${}^W\xi_s$ and ${}^W\xi_f$, i.e. two times per number of DoF, one considering ${}^Vd_{min}$ and another one considering ${}^Hd_{min}$. On the one hand, as expected, only the consideration of ${}^Hd_{min}$ and ${}^Hd_{min}$ leads to a planned visual path successfully followed with 4 and 6 DoF, respectively. It means the deduction of ${}^Hd_{min}$ for both 4 and 6 DoF thanks to Eq. 7 is relevant, although it has no value of generality. On the other hand, both ${}^Vd_{min}$ and ${}^Hd_{min}$ lead to a planned visual path successfully followed with 2 DoF. However, the visual path planned considering ${}^Vd_{min}$ is not straight (Fig. 5(a)) whereas it is when considering ${}^Hd_{min}$ (similar to the 4 DoF case of Fig. 5(d)). Although both P-A* experiments with ${}^Hd_{min}$ and ${}^Hd_{min}$ are not reported in the paper, they are shown in the video accompanying the paper¹.

2) *Robustness to scene and model contents*: This last set of experiments aims at showing the power and limits of adaptability of P-A* with respect to the scene content. The camera 2 DoF result obtained with ${}^Vd_{min}$ above in *nominal* conditions serves as reference with respect to which two additional P-A* experiments are compared. Both experiments are challenging in the sense that the first one involves a *textureless zone*, of more than 50% of both the scene and its model, and the second one an *outdated model* made 10 years before looking at today’s actual scene.

The three experiments are reported and compared below.

a) *Nominal Case*: At the visual path planning stage, \mathcal{G} was initialized with $|\mathcal{V}| = 841$ in 2.3 s. The shortest visual path \mathcal{K} , in the sense of A*, was produced in 2286.5 s. \mathcal{K} is composed of 35 images. The robot successfully reaches each $K \in \mathcal{K}$ (see its trajectory superimposed on the planned path in Fig. 5(a)).

b) *Textureless Zone*: In this experiment, a textureless area is set at the center of the scene and its model. The interest here is to study the case where no visual path along the straight line between ${}^W\xi_s$ and ${}^W\xi_f$ is fully navigable (no visual information to control the camera DoF in some area of the scene). At the visual path planning stage, P-A* automatically determines that some visual paths, far from the straight line, are navigable. The shortest path, in the sense of A*, was found from \mathcal{G} with $|\mathcal{V}| = 841$ (initialized in 2.3 s as well) in 2245.8 s. The visual path \mathcal{K} is made up of 35 images. At the navigation stage, the camera successfully reaches the final pose ${}^W\xi_f$ despite some difficulties when reaching the textureless zone (Fig. 5(b)).

c) *Outdated Model*: The image on this model was acquired at a different time of day and ten years before the image printed on the tarpaulin. Consequently, its content and the content of today’s scene vary in two main aspects: the orientation and the size of the shadows, and some remodeled buildings. Notice that this experiment is similar to the case when the preobtained model of the scene had not been recently updated on a real navigation task performed by an aerial robot, for instance. The shortest \mathcal{K} , in the A* sense,

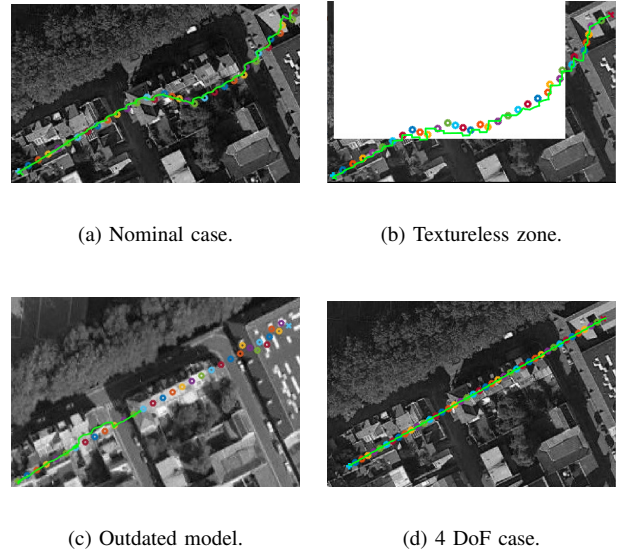


Fig. 5. Planned paths (multicolor o) and executed trajectories (green lines, + for start and x for final poses) with (a)-(c) 2 DoF, considering ${}^Vd_{min}$; (d) 4 DoF, considering ${}^Hd_{min}$.

was found in 1831.7 seconds from \mathcal{G} with $|\mathcal{V}| = 841$ (also initialized in 2.3 seconds). \mathcal{K} includes 34 images. Despite the dissimilarities between the images in \mathcal{K} and the scene, the camera traveled a distance of 225 mm over 495 mm towards its goal (Fig. 5(c)) before diverging, which was expected since nothing is done at the image comparison level (Eq. 2) to be robust to such differences of content between the actual scene and its model.

In all of the experiments presented in this Section, the discontinuous velocities produced by the memory update do not cause enough disruptions in the robot to prevent the mission from being achieved. This issue must be addressed for robots that navigate at higher speed.

V. CONCLUSION

In this paper we introduced the Photometric-A*, P-A*, a Vision-Based Navigation System that exploits the characteristics of photometric visual servoing. P-A* combines the advantages of relying on a preobtained model of the scene and navigating using a visual memory. Given start and final poses over the scene, the planner of P-A* outputs a short and navigable visual path by leveraging on the preobtained model. The robot, equipped with a camera, is successfully controlled with 2, 4 and 6 degrees of freedom through the scene following such a visual path. Future works will be dedicated to exploring different models (e.g. dense surface model or 3D model), to search paths on graphs in higher dimensions than \mathbb{R}^2 as well as other sensors.

ACKNOWLEDGMENT

The Mexican National Council for Science and Technology (CONACYT) funded this research.

¹available at http://mis.u-picardie.fr/~g-caron/videos/2020_ICRA.mp4

REFERENCES

- [1] B. Rahmani, A. E. Putra, A. Harjoko, and T. K. Priyambodo, "Review of vision-based robot navigation method," *IAES Int. Journal of Robotics and Automation*, vol. 4, no. 4, 2015.
- [2] G. N. DeSouza and A. C. Kak, "Vision for mobile robot navigation: A survey," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 24, no. 2, pp. 237–267, 2002.
- [3] F. Bonin-Font, A. Ortiz, and G. Oliver, "Visual navigation for mobile robots: A survey," *Journal of intelligent and robotic systems*, vol. 53, no. 3, pp. 263–296, 2008.
- [4] A. Chatterjee, A. Rakshit, and N. Singh, *Vision based autonomous robot navigation: algorithms and implementations*. Springer, 2012, vol. 455.
- [5] M. Warren, M. Greeff, B. Patel, J. Collier, A. P. Schoellig, and T. D. Barfoot, "There's no place like home: Visual teach and repeat for emergency return of multirotor uavs during gps failure," *IEEE Robotics and Automation Letters*, vol. 4, no. 1, pp. 161–168, 2018.
- [6] P. Furgale and T. D. Barfoot, "Visual teach and repeat for long-range rover autonomy," *Journal of Field Robotics*, vol. 27, no. 5, pp. 534–560, 2010.
- [7] G. Costante, J. Delmerico, M. Werlberger, P. Valigi, and D. Scaramuzza, "Exploiting photometric information for planning under uncertainty," *Robotics Research*, pp. 107–124, 2018.
- [8] G. Bresson, Z. Alsayed, L. Yu, and S. Glaser, "Simultaneous localization and mapping: A survey of current trends in autonomous driving," *IEEE Trans. on Intelligent Vehicles*, vol. 20, pp. 1–1, 2017.
- [9] A. Kosaka and J. Pan, "Purdue experiments in model-based vision for hallway navigation," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, vol. 95, 1995, pp. 87–96.
- [10] A. Gasteratos, C. Beltran, G. Metta, and G. Sandini, "Pronto: a system for mobile robot navigation via cad-model guidance," *Microprocessors and Microsystems*, vol. 26, no. 1, pp. 17–26, 2002.
- [11] T. Goedemé, M. Nuttin, T. Tuytelaars, and L. Van Gool, "Omnidirectional vision based topological navigation," *Int. Journal of Computer Vision*, vol. 74, no. 3, pp. 219–236, 2007.
- [12] D. Fontanelli, A. Danesi, F. A. Belo, P. Salaris, and A. Bicchi, "Visual servoing in the large," *Int. Journal of Robotics Research*, vol. 28, no. 6, pp. 802–814, 2009.
- [13] A. Remazeilles and F. Chaumette, "Image-based robot navigation from an image memory," *Robotics and Autonomous Systems*, vol. 55, no. 4, pp. 345–356, 2007.
- [14] J. Courbon, Y. Mezouar, and P. Martinet, "Indoor navigation of a non-holonomic mobile robot using a visual memory," *Autonomous Robots*, vol. 25, no. 3, pp. 253–266, 2008.
- [15] J. Courbon, Y. Mezouar, N. Guénard, and P. Martinet, "Vision-based navigation of unmanned aerial vehicles," *Control Engineering Practice*, vol. 18, no. 7, pp. 789–799, 2010.
- [16] T. Nguyen, G. K. Mann, and R. G. Gosine, "Vision-based qualitative path-following control of quadrotor aerial vehicle," in *IEEE Int. Conf. on Unmanned Aircraft Systems*, 2014, pp. 412–417.
- [17] D. Santosh, S. Achar, and C. Jawahar, "Autonomous image-based exploration for mobile robot navigation," in *IEEE Int. Conf. on Robotics and Automation*, 2008, pp. 2717–2722.
- [18] A. Diosi, S. Segvic, A. Remazeilles, and F. Chaumette, "Experimental evaluation of autonomous driving based on visual memory and image-based visual servoing," *IEEE Trans. on Intelligent Transportation Systems*, vol. 12, no. 3, pp. 870–883, 2011.
- [19] K. D. Sharma, A. Chatterjee, and A. Rakshit, "Experimental study II: Vision-based navigation of mobile robots," in *Intelligent Control*, 2018, pp. 243–280.
- [20] P. Corke, *Robotics, Vision and Control: Fundamental Algorithms In MATLAB® Second, Completely Revised*. Springer, 2017.
- [21] C. Collewet and E. Marchand, "Photometric visual servoing," *IEEE Trans. on Robotics*, vol. 27, no. 4, pp. 828–834, 2011.
- [22] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, *Autonomous mobile robots*. MIT press, 2011.
- [23] C. Teulière and E. Marchand, "A dense and direct approach to visual servoing using depth maps," *IEEE Trans. on Robotics*, vol. 30, no. 5, pp. 1242–1249, 2014.
- [24] E. Rublee, V. Rabaud, K. Konolige, and G. R. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *CVF/IEEE Int. Conf. on Computer Vision*, vol. 11, no. 1, 2011, p. 2.
- [25] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.
- [26] K. Wu, X. Li, R. Ranasinghe, G. Dissanayake, and Y. Liu, "RISAS: a novel rotation, illumination, scale invariant appearance and shape feature," in *IEEE Int. Conf. on Robotics and Automation*, 2017, pp. 4008–4015.
- [27] A. Dame and E. Marchand, "Mutual information-based visual servoing," *IEEE Trans. on Robotics*, vol. 27, no. 5, pp. 958–969, 2011.
- [28] N. Crombez, G. Caron, and E. Mouaddib, "3D point cloud model colorization by dense registration of digital images," *Int. Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 40, no. 5, p. 123, 2015.
- [29] S. Park, T. Schöps, and M. Pollefeys, "Illumination change robustness in direct visual slam," in *IEEE Int. Conf. on Robotics and Automation*, 2017, pp. 4523–4530.
- [30] B. Pan, "Recent progress in digital image correlation," *Experimental Mechanics*, vol. 51, no. 7, pp. 1223–1235, 2011.
- [31] E. Malis, "Improving vision-based control using efficient second-order minimization techniques," in *IEEE Int. Conf. on Robotics and Automation*, vol. 2, 2004, pp. 1843–1848.
- [32] C. Collewet, E. Marchand, and F. Chaumette, "Visual servoing set free from image processing," in *IEEE Int. Conf. on Robotics and Automation*, 2008, pp. 81–86.
- [33] G. Chesi and K. Hashimoto, *Visual servoing via advanced numerical methods*. Springer, 2010.
- [34] V. Chepoi, F. Dragan, and Y. Vaxès, "Center and diameter problems in plane triangulations and quadrangulations," in *Symposium on Discrete Algorithms*, vol. 2, 2002, pp. 346–355.
- [35] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Trans. on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.
- [36] E. Dijkstra, "A note on two problems in connection with graphs," *Numerische Mathematik*, vol. 1, pp. 269–271, 1959.