Defocus-based Direct Visual Servoing

Guillaume Caron

Defocus-based Direct Visual Servoing

Guillaume Caron

***Abstract*— Direct Visual Servoing (DVS) considers pixel brightness directly as input of robot control. Recent DVS variants consider image processing as smoothing or frequency domain transforms, resulting in large convergence domains.**

This paper proposes to consider defocus to optically smooth images without processing. The resulting Defocus-based DVS shows convergence domains competing with the state-of-the-art, larger in some challenging cases, for lower complexity.

I. INTRODUCTION

A. Motivation

Visual Servoing (VS) is the feedback control of robot motion with images of a vision sensor [1]. This fundamental framework designs the control law so that information carried by the acquired image reach a desired value. The control law may rely on the *interaction matrix*, linking the camera velocity to the variation of visual information.

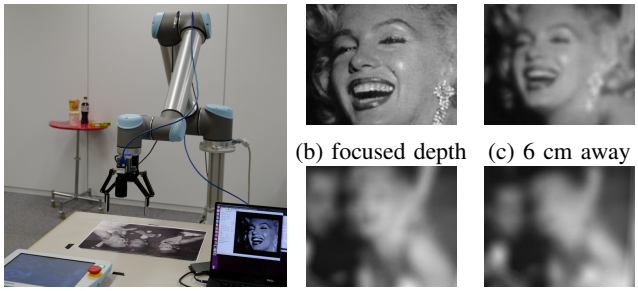
VS considers indirect or direct information. Indirect VS (IVS) requires the extraction of geometric features from images whereas Direct VS (DVS) does not. Instead, DVS original version, *i.e.* Photometric DVS (PVS), directly considers pixel brightness of the whole image as input of the control law. This allows higher positioning precision for PVS than any IVS but within a tighter convergence domain [2].

To enlarge the convergence domain of PVS, several *transforms* of images were introduced : kernels [3], gradients [4], [5], Photometric Gaussian Mixtures (PGM) [6], subspaces [7], [8], [9], photometric moments [10], frequency domain [11], [12]. Such image transforms-based DVS (t-DVS) show a significantly larger convergence domain than PVS, *e.g.* +25% in translation on real robot for PGM-based DVS (PGM VS) thanks to an adaptive smoothing of images [6]. Intensity-based VS [13], in-between DVS and IVS, reaches even larger convergence domains but requires the direct tracking of an image region, as input of the control law.

t-DVS' transforms increase the algorithmic complexity. Indeed, considering an image of $N \times M$ pixels, the algorithmic complexity in computing the interaction matrix of PGM VS is $O((NM)^2)$ against $O(NM)$ for the seminal PVS. So, this paper investigates a new DVS using smooth images directly acquired thanks to defocus (Fig. 1). The goal is to reach a large convergence domain as PGM VS but with an algorithmic complexity as low as the one of PVS.

B. Related works

Defocus properties, also known as the "Bokeh effect", are well known since decades by photographers of art. They have been exploited both in computer graphics for realistic rendering [14], [15], [16] and in computer vision for dense depth computation [17], [18], [19]. In short, the amount of defocus blur in the image depends on the depth difference between scene points and the plane in focus (which depth is set by the camera lens).



(a) Visual servoing setup (b) focused depth (c) 6 cm away (d) 26 cm away (e) 36 cm away

Fig. 1: Defocus with respect to depth.

To date, the robotics research mainly considered defocus by integrating computer vision works of depth estimation, *e.g.* for mobile robot navigation [20], Simultaneous Localization And Mapping [21] or micro-positionning [22], [23], [5]. Micro-positionning works are closer to robot control than others but exploit defocus to estimate the focus depth [22] or virtually extend the depth-of-field to track a micropart [23]. Then, the outputs of the latter processes feed a control law of one [22] or three [23] Degrees of Freedom (DoF).

Among micro-positionning works, only [5] belongs to DVS, in a hybrid scheme. Indeed, five over six DoF are controlled from image brightness with PVS [2]. The translation along the optical axis is controlled from image gradients assuming their defocus vary proportionally to the variation of the camera pose along its optical axis. The proportionality assumption is made for small variations of that pose.

C. Contributions

This paper introduces the Defocus-based DVS (DDVS). Its core idea is to model the closest link between brightness of defocused images and six DoF robot control within the DVS framework. DDVS builds on the full non-linear relationship between defocus and camera pose with the goal of making the defocus blur benefiting to the convergence domain of the control law as PGM [6] did with Gaussian blurred images. Main differences are related to blur characteristics and processing times. Indeed, PGM features an isotropic Gaussian blur whereas the blur related to defocus is anisotropic. Then, as the defocused image is directly acquired and not the result of processing a sharp one, processing times are shrinked.

Guillaume Caron is with CNRS-AIST JRL (Joint Robotics Laboratory), IRL, AIST, Tsukuba, Japan and with Université de Picardie Jules Verne, MIS laboratory, Amiens, France guillaume.caron@u-picardie.fr

The main contributions are:

- explicit formulation of the DDVS interaction matrix
- clear impact of defocus on the convergence domain of the DDVS control law thanks to simulations
- extensive evaluation with respect to the state-of-the-art on positionning tasks of a 6 DoF real robot arm
- description of the drawbacks of the new method.

D. Outline

Section II recalls related DVS, *i.e* PVS and PGM VS. Section III models defocus in order to detail the interaction matrix of DDVS (Sec. IV). Finally, Section V shows experimental evaluation of DDVS, with a 6 DoF robot arm and several scenes, with respect to baseline works.

II. DIRECT VISUAL SERVOING

A. Photometric DVS: PVS

PVS is the seminal Photometric Visual Servoing [2] where all pixel brightness are used as input of the control law to drive the camera from an initial pose to a desired pose. Here is a recall of the control law together with notations.

Considering image I of size $N \times M$ pixels, its definition domain is $\mathcal{U} = [\![0, N-1]\!] \times [\![0, M-1]\!]$, such that $I : \mathbf{u} \in \mathcal{U} \longmapsto I(\mathbf{u}) \in [\![0, 255]\!]$. Then, the input of the PVS is the stacking of brightness[1] $I(\mathbf{u})$ of all pixels of I, acquired at the current location $\mathbf{p} \in \mathbb{R}^6$. The latter stacking leads to vector $\mathbf{I}(\mathbf{p}) \in [\![0, 255]\!]^{|\mathcal{U}|\times 1}$:

$$\mathbf{I}(\mathbf{p}) = [\mathbf{I}_{1\bullet}, \mathbf{I}_{2\bullet}, \dots, \mathbf{I}_{N\bullet}]^\top, \tag{1}$$

where $\mathbf{I}_{i\bullet} \in [\![0, 255]\!]^{1\times M}$ is the i-th line of I. Vector $\mathbf{p} = [\mathbf{t}^\top, \theta\mathbf{w}^\top]^\top$ represents the camera pose, *i.e.* $\mathbf{t} = [t_X, t_Y, t_Z]^\top \in \mathbb{R}^3$ is the 3D translation and the 3D rotation is represented as axis-angle with axis $\mathbf{w} = [w_X, w_Y, w_Z]^\top \in \mathbb{R}^3 : ||\mathbf{w}|| = 1$ and angle $\theta \in \mathbb{R}$.

Brightness vector $\mathbf{I}^*$ is built at desired pose $\mathbf{p}^*$ from the desired image I^*, as $\mathbf{I}(\mathbf{p})$ is built with (1) from I. PVS is designed to minimize the Sum of Squared Differences (SSD):

$$\mathscr{C}(\mathbf{p}) = \frac{1}{2}||\mathbf{I}(\mathbf{p}) - \mathbf{I}^*||^2, \tag{2}$$

with the below control law [2]:

$$\mathbf{v} = -\mu {\mathbf{L}_\mathbf{I}}^+(\mathbf{I}(\mathbf{p}) - \mathbf{I}^*). \tag{3}$$

In (3), $\mathbf{v} \in \mathbb{R}^n$ is the n DoF camera velocity ($\mathbf{v} \approx \dot{\mathbf{p}}$, if $n = 6$), $\mu \in \mathbb{R}_+^*$ is a gain, and ${\mathbf{L}_\mathbf{I}}^+$ is the pseudo-inverse of the interaction matrix $\mathbf{L}_\mathbf{I} \in \mathbb{R}^{|\mathcal{U}|\times n}$. $\mathbf{L}_\mathbf{I}$ links the variations of image brightness $\mathbf{I}$ to the n considered camera pose DoF, thanks to the *Optical Flow Constraint Equation* (OFCE), valid in Lambertian scenes [24]. $\mathbf{L}_\mathbf{I}$ is the stacking of interaction matrices $\mathbf{L}_{I(\mathbf{u})}$ evaluated for all $\mathbf{u} \in \mathcal{U}$, each composing image gradients $\nabla I_\mathbf{u} \in [\![-255, 255]\!]^2$ to the geometric interaction matrix $\mathbf{L}_\mathbf{u} \in \mathbb{R}^{2\times n}$:

$$\mathbf{L}_{I(\mathbf{u})} = -\nabla I_\mathbf{u}^\top \mathbf{L}_\mathbf{u}. \tag{4}$$

[1]Pixel brightness coded with 8 bits are used, since very common, but any other quantization could be considered.

The reader may refer to [1] for the details of $\mathbf{L}_\mathbf{u}$ involving the pinhole camera model and the motion of a 3D point.

PVS shows high precision at convergence, *i.e.* below one tenth of millimeter [2], while avoiding feature detection and matching. But its main drawback is its narrow convergence domain compared to feature-based VS [12].

B. Photometric Gaussian Mixtures-based DVS: PGM VS

PGM VS [6] is the t-DVS reaching the widest convergence domain of related works. To reach a larger convergence domain than PVS, PGM VS [6] no longer considers directly I as input. Instead, its input is G, the transform of image I as a Photometric Gaussian Mixture. Before detailing G, let us note that G mixes Photometric Gaussians (PG) of every pixel $I(\mathbf{u})$. Thus, a single PG is characterized by its *center* $\mathbf{u}$ and its *spread* $\lambda \in \mathbb{R}_+^*$, *i.e.* its mean and standard deviation in a statistical phrasing. Hence a PG is defined as $g : \mathbf{u}_g \in \mathcal{U}_g \longmapsto g(\mathbf{u}_g, I, \mathbf{u}, \lambda)$:

$$g(\mathbf{u}_g, I, \mathbf{u}, \lambda) = I(\mathbf{u}) \exp\left(-||\mathbf{u}_g - \mathbf{u}||^2/(2\lambda^2)\right). \tag{5}$$

g models the power of attraction of the pixel at location $\mathbf{u}$ in image I [6]. If the definition domain of the PG is the same as the one of the image, then $\mathcal{U}_g = \mathcal{U}$.

After that, the PGM G is defined considering a single λ:

$$G(\mathbf{u}_g, I, \lambda) = \sum_{\mathbf{u}\in\mathcal{U}} g(\mathbf{u}_g, I, \mathbf{u}, \lambda). \tag{6}$$

As $\mathcal{U}$ is a discrete set, the exponential in the expression (5) of g is evaluated at discrete locations $\mathbf{u} \in \mathcal{U}$, thus sampled. Hence, if λ tends to 0, then $G(\mathbf{u}_g)$ (in short) shrinks to $I(\mathbf{u})$.

The control law of PGM VS is designed to minimize the SSD cost between current G, *i.e.* the PGM transform of I with λ, and G^*, *i.e.* the PGM transform of I^* with $\lambda^* \in \mathbb{R}_+^*$. PGM VS considers λ as an additionnal DoF to those of camera pose $\mathbf{p}$ [6]. λ^* is set constant so it can be different than λ, both at initialization and while the control loop is running. The evolution of λ while the camera is moving is one of the PGM VS's key of both wide convergence domain (large λ) and precision at convergence (small λ).

To express the PGM SSD cost, elements of G are stacked as vector $\mathbf{G} \in \mathbb{R}^{|\mathcal{U}_g|\times 1}$ and those of G^* as $\mathbf{G}^* \in \mathbb{R}^{|\mathcal{U}_g|\times 1}$:

$$\mathscr{C}_{PGM}(\mathbf{p}, \lambda) = \frac{1}{2}||\mathbf{G}(\mathbf{p}, \lambda) - \mathbf{G}^*||^2. \tag{7}$$

The control law minimizing $\mathscr{C}_{PGM}$ is then expressed as:

$$\begin{bmatrix}\mathbf{v}\\ \dot{\lambda}\end{bmatrix} = -\mu {\mathbf{J}_\mathbf{G}}^+(\mathbf{G}(\mathbf{p}, \lambda) - \mathbf{G}^*), \tag{8}$$

where $\mathbf{J}_\mathbf{G} \in \mathbb{R}^{|\mathcal{U}_g|\times n+1}$ juxtaposes $\mathbf{L}_\mathbf{G} \in \mathbb{R}^{|\mathcal{U}_g|\times n}$, *i.e.* the interaction matrix related to a PGM sample, and $\mathbf{J}_\lambda \in \mathbb{R}^{|\mathcal{U}_g|\times 1}$, *i.e.* the Jacobian of the PGM sample with respect to λ. One may refer to [6] for their detailed expressions but may note that $\mathbf{L}_\mathbf{G}$ is function of geometric interaction matrices involving the pinhole camera model, as $\mathbf{L}_\mathbf{I}$ (3), (4).

λ and λ^* are critical for the ideal behavior of PGM VS, *i.e.* large convergence domain and precision at convergence. Thus, [6] reports a sequence of PGM VS: Step 1 with a

large λ^* (exact value depends on experiments) and $\lambda = 2\lambda^*$ at initialization; Step 2 with constant $\lambda = \lambda^* = 1$. Step 1 allows the large convergence domain and Step 2 allows precision.

III. MODEL OF DEFOCUS

A. Overview

Defocus occurs in an acquired image when the scene depth is not included in the depth-of-field, which is the volume in the interval of scene depths that appear sharp in the image [25]. The depth-of-field optically depends on the camera aperture: narrow aperture leads to large depth-of-field while large aperture leads to shallow depth-of-field. Defocus pixels appear blurred. The defocus blur can be approximated by a normal Gaussian kernel [26]. The amount of defocus blur increases with the distance of scene depth to the volume in focus. Thus, moving a camera of constant large aperture, even in a static environment, makes the apparent blur vary depending on the camera pose (Fig. 1).

Thinking about a planar scene fronto-parallel to the image at the desired camera pose $\mathbf{p}^*$, the defocus behavior is intuitively similar to PGM VS's control of λ, for forward/backward motion along the camera optical axis (Fig. 1b to 1e). Indeed, PGM VS's control of λ is done to adapt the smoothness of current G simultaneously to driving the camera toward $\mathbf{p}^*$, where G is expected as sharp as G^* (Sec. II-B). However, smoothness characteristics of defocus are different of those of PGM.

Indeed, in the PGM transform, (5) can be interpreted more classically as the convolution of a Gaussian kernel g with the image I, where g is isotropic (same λ for every pixel of I, (6)). In general non-planar scenes, the normal Gaussian kernel approximating the defocus is anisotropic as the amount of blur depends on the distance of the scene to the focused volume. Section III-C formally expresses the anisotropic Gaussian kernel related to defocus. It relies on the thin-lens camera model, shortly recalled in Section III-B.

B. Thin lens camera model

The pinhole camera model describes the ideal stenope [27], to which a conventional camera gets close only for pixels in focus or when its aperture is narrow. The thin lens camera model [27] allows describing any aperture setting, including large open, thus defocus characteristics.

A camera following the pinhole model images a 3D point $\mathbf{X} = [X, Y, Z]^\top \in \mathbb{R}^3$ as a single 2D point on the actual image plane. Instead, a camera following the thin lens model maps $\mathbf{X}$ to a surface named the *Circle of Confusion* (CoC). The diameter of the CoC tends to zero when $\mathbf{X}$ is in focus and increases as $\mathbf{X}$ gets away the focused depth. The CoC also depends on several optical parameters of the camera: focal length $f \in \mathbb{R}^*_+$, aperture diameter $D \in \mathbb{R}^*_+$, focus depth $Z_f \in \mathbb{R}^*_+$. Then, the CoC diameter $d(Z) \in \mathbb{R}^*_+$ of scene point $\mathbf{X}$, expressed in the camera lens frame, is expressed as [14]:

$$d(Z) = \frac{Df}{Z_f - f}\left(1 - \frac{Z_f}{Z}\right). \tag{9}$$

If $d(Z)$ is lower than the side $k_\mathbf{u} \in \mathbb{R}^*_+$ of a photodiode of the camera sensor, then the corresponding pixel is sharp. Otherwise the image locally features defocus blur.

Recall that the aperture diameter D is usually described as the unitless F-Number quantity for most camera lenses. The F-Number is commonly written "F-ϕ", with $\phi \in \mathbb{R}^*_+$, to indicate how closed the aperture is. Then, D is expressed from the F-Number value ϕ and the focal length f as [27]:

$$D = f/\phi. \tag{10}$$

C. Anisotropic Gaussian kernel related to CoC

Defocus blur is approximated by anisotropic filtering with normal Gaussian kernels depending on the CoC of the thin lens camera model (Sec. III-B). Every CoC depends on the same camera parameters but each CoC depends on the Z coordinate of the 3D point $\mathbf{X}$ it images (9). Hence, the normal Gaussian kernel related to the CoC of $\mathbf{X}$ in image I possibly has its own unique spread $\lambda(Z) \in \mathbb{R}^*_+$.

In order to match the CoC with a normal Gaussian kernel, we express the extension $\lambda(Z)$ of the latter depending on the diameter $d(Z)$ of the former (9). Assuming 99.7% of the normal Gaussian is included in the CoC, we get:

$$\lambda(Z) = d(Z)/(6\, k_\mathbf{u}), \tag{11}$$

where $k_\mathbf{u}$, the physical size of a pixel, converts actual image plane units, *i.e.* meter, to the digital image ones.

Then, we make the approximation that $\mathbf{X}$ projects at the center of the CoC following the pinhole camera model[2] $pr : \mathbf{X} \in \mathbb{R}^3 \longmapsto \mathbf{x} \in [0, N-1] \times [0, M-1]$:

$$\mathbf{x} = \begin{bmatrix} f/k_\mathbf{u} & 0 & u_0 \\ 0 & f/k_\mathbf{u} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X/Z \\ Y/Z \\ 1 \end{bmatrix} = pr(\mathbf{X}), \tag{12}$$

where $\mathbf{X}$ is expressed in the camera frame and $u_0 \in \mathbb{R}$ and $v_0 \in \mathbb{R}$ are the coordinates of the principal point in the digital image. $\mathbf{x}$ is expressed in the digital image plane as $\mathbf{u}$ but with real coordinates. Hence, the intrinsic parameters of the thin-lens camera model are: $\gamma = \{f, k_\mathbf{u}, u_0, v_0, \phi, Z_f\}$.

Using (11) and (12), the anisotropic normal Gaussian kernel $\tilde{g}$ related to defocus is expressed as:

$$\tilde{g}(\mathbf{u}, \mathbf{X}) = \frac{1}{2\pi\lambda(Z)^2} \exp\left(-\frac{||\mathbf{u} - pr(\mathbf{X})||^2}{2\,\lambda(Z)^2}\right). \tag{13}$$

If $\lambda(Z)$ tends to 0, then $\tilde{g}$ tends to a Dirac impulse.

Finally, the scene radiance $L(\mathbf{X}) \in \mathbb{R}_+$ of $\mathbf{X}$ is assumed equally mapped to brightness $I(pr(\mathbf{X}))$, when $\mathbf{X}$ is in focus[3], *i.e.* $I(pr(\mathbf{X})) = L(\mathbf{X})$. Then, considering the scene is a continuous set $\mathcal{X}$ of 3D points $\mathbf{X} \in \mathcal{X}$, the defocus image I_d writes as:

$$I_d(\mathbf{u}) = \int_\mathcal{X} I(pr(\mathbf{X}))\, \tilde{g}(\mathbf{u}, \mathbf{X})\, d\mathbf{X}. \tag{14}$$

[2]As the CoC is geometrically the intersection of the cone of light emanating from $\mathbf{X}$ with the actual image plane, $\mathbf{X}$ exactly projects at the center of its CoC, if and only if $\mathbf{X}$ is aligned with the optical axis.

[3]This equal mapping is voluntarily a huge simplification of the imaging process to make clear expressions. Rigorously, other transformations as vignetting, camera response function, quantization can be considered [28].

A first look at (14) shows high similarity with the PGM transform (5), (6). Actually, in the particular case of a scene fronto-parallel to the camera and only t_Z motion is considered, the anisotropic filtering shrinks to isotropic as PGM does (though the spread of the Gaussian kernel is set differently). More deeply, there are fundamental differences. First, the most obvious, $I_d(\mathbf{u})$ may directly be acquired blurred in (14) contrary to $G(\mathbf{u}_g, I, \lambda)$ in (6), thus saving digital processing time. Second, the kernel spread $\lambda(Z)$ of (13) is strongly linked to the camera pose $\mathbf{p}$, contrary to the kernel spread λ of the PGM (5). The main consequence is: one no longer needs to find an evolution law of the kernel spread during the VS. Thus, DDVS exploiting (14) needs to consider only the camera pose DoF, as Section IV shows.

IV. DEFOCUS-BASED DIRECT VISUAL SERVOING

As PVS (Sec. II-A) and PGM VS (Sec. II-B), we design the control law of DDVS to minimize the SSD of the current I_d and desired I_d^* images, both acquired with the camera of constant large aperture. Hence, pixel brightness of acquired images are stacked as current $\mathbf{I_d}(\mathbf{p}) \in [\![0, 255]\!]^{|\mathcal{U}|\times 1}$ and desired $\mathbf{I_d}^* \in [\![0, 255]\!]^{|\mathcal{U}|\times 1}$ brightness vectors.

Then, the defocus SSD cost is expressed as:

$$\mathscr{C}_d(\mathbf{p}) = \frac{1}{2}||\mathbf{I_d}(\mathbf{p}) - \mathbf{I_d}^*||^2, \tag{15}$$

to be minimized by the control law:

$$\mathbf{v} = -\mu \mathbf{L_{I_d}}^+ (\mathbf{I_d}(\mathbf{p}) - \mathbf{I_d}^*), \tag{16}$$

very similar to (3) and (8), since $\mathbf{I_d}^*$ is constant as previously recalled DVS (Sec. II). In (16), the interaction matrix $\mathbf{L_{I_d}} \in \mathbb{R}^{|\mathcal{U}|\times n}$ is nothing but the stacking of interaction matrices $\mathbf{L}_{I_d}(\mathbf{u}) \in \mathbb{R}^{1\times n}$ computed for each $I_d(\mathbf{u}), \forall \mathbf{u} \in \mathcal{U}$.

A. DDVS interaction matrix

As the defocus of I_d may evolve with respect to $\mathbf{p}$, the expression of $\mathbf{L}_{I_d}(\mathbf{u})$ must rely on the *focal flow constraint equation* [18] (FFCE). It is more general than the OFCE [24] since the FFCE models the brightness *inconsistency* due to defocus on time $t \in \mathbb{R}_+$ and $\delta t \in \mathbb{R}_+$ later as:

$$I_d(\mathbf{u} + \boldsymbol{\delta}\mathbf{x}, t + \delta t) = \tilde{g}(\mathbf{u}, \boldsymbol{\delta}\mathbf{X}) * I_d(\mathbf{u}, t), \tag{17}$$

where $\boldsymbol{\delta}\mathbf{x} \in \mathbb{R}^2$ is the motion in the image plane, $\boldsymbol{\delta}\mathbf{X} = [0, 0, \delta Z]^\top \in \mathbb{R}^3$ represents the change of camera Z coordinate during δt and the operator $*$ denotes the convolution product. In (17), if $\delta Z = 0$ or if the CoC tends to a point (9), (17) falls back to the brightness consistency equation exploited for PVS.

A first order Taylor expansion of (17) leads to the FFCE:

$$\nabla_{\mathbf{u}} I_d^\top \dot{\mathbf{u}} + \frac{\mathrm{d}\, I_d(\mathbf{u}, t)}{\mathrm{d}\, t} \approx -\Delta_{\mathbf{u}} I_d \left(\frac{\lambda(Z)}{Z} + \frac{\partial \lambda(Z)}{\partial Z} \right) \dot{Z}, \tag{18}$$

where the equality is approximate since high order terms of the expansion are dropped and $\Delta_{\mathbf{u}} I_d \in [\![-512, 512]\!]$ is the Laplacian of I_d. If there is no motion along the camera optical axis, *i.e.* $\dot{Z}$ is null, or if the CoC tends to a point, *i.e.* $\lambda(Z)$ tends to zero and its derivative with respect to Z:

$$\frac{\partial \lambda(Z)}{\partial Z} = \frac{D Z_f f}{6 k_u (Z_f - f) Z^2}, \tag{19}$$

too, then the right side of (18) tends to zero and the FFCE falls back to the OFCE.

Finally, as $\dot{\mathbf{u}}$ and $\dot{Z}$ can be expressed in terms of their partial derivatives with respect to the camera pose [1], one deduces from (18) the expression of the interaction matrix $\mathbf{L}_{I_d}(\mathbf{u})$ related to brightness with defocus as:

$$\mathbf{L}_{I_d}(\mathbf{u}) = \begin{bmatrix} -\nabla_{\mathbf{u}} I_d^\top \\ -\Delta_{\mathbf{u}} I_d \end{bmatrix} \begin{bmatrix} \mathbf{L_u} \\ \frac{Df}{6k_u(Z_f - f)Z} \mathbf{L}_Z \end{bmatrix}, \tag{20}$$

with, as a recall [1]:

$$\mathbf{L}_Z = \begin{bmatrix} 0 & 0 & -1 & -Y & X & 0 \end{bmatrix}. \tag{21}$$

As most previous DVS, since $|\mathcal{U}| >> n$, only local stability can be ensured. The convergence domain can be rather large in practice [1], which is evaluated in the rest of the article.

B. DDVS simulation

In order to clearly highlight the contribution of considering defocus for DVS, simulation results are reported. Simulations consider a synthetic scene with a single bright 3D point $\mathbf{X} = [0, 0, 0]^\top$ in order to understand the DDVS behavior regarding defocus only. The radiance of $\mathbf{X}$ is set to $L(\mathbf{X}) = 1$. Then, we consider a linear camera of intrinsic parameters γ (Sec. III-C) which pose is restricted to 2 DoF: t_Z and t_X.

1) Realistic camera aperture: First, we consider camera aperture F-ϕ settings with $\phi = 8$ and $\phi = 0.95$. The latter ϕ denotes wide aperture, actually the widest on the market of compact machine vision camera lenses of C/CS mount (Note that the IB-E Optics company provides a lens with $\phi = 0.85$ but for photographer cameras with E, X, SL and M mounts). With $\phi = 8$, we simulate a large depth-of-field, in which the pinhole assumption of PVS is met, still being capable of acquiring 30 images per second without motion blur (at low speed) on most machine vision cameras.

Intrinsic parameters are set as in experiments of Section V, *i.e.* $\gamma = \{17$ mm, 5.3 μm, 320 pixels, 256 pixels, ϕ, 25 cm$\}$, in order to anticipate practical behaviors of DDVS. $Z_f = 25$ cm is the smallest possible focus depth of the lens used in experiments (Sec. V). This setting allows observing variations of defocus even for small variations of t_Z.

The desired camera pose is $\mathbf{p}^* = [t_X^*, 0, t_Z^*, 0, 0, 0]^\top$. $t_X^* = -t_X(0)/2$ such that $t_X(0)$ is the horizontal translation in space that VS must correct. By doing so, the initial and desired $pr(\mathbf{X})$ (see (12)) are symmetric with respect to the image center of coordinate u_0. Then, t_Z^* is set such that the desired brightness $I_d^*(\mathbf{u})$ is equal to 1, following the image formation model (14). Finally, initial t_Z is set to $t_Z(0) = t_Z^*$.

With a single non null pixel at both initial and desired poses, PVS is expected to only control t_X. Conversely, as DDVS considers explicitely defocus, it is expected to control t_Z as well. Table I reports DDVS parameters and maximum initial distances that allow convergence, both in space with

TABLE I: Simulation results of DDVS: maximum errors in space and in pixels corrected for various apertures.

ϕ	t_Z^* (mm)	$t_X(0)$ (mm)	$\boldsymbol{\delta}\mathbf{x}$ (pixel)	iter.
8	289.0000	0.4	2.1	36
0.95	253.6000	5	31.7	71
0.1	250.04841	75	480.1	98

$t_X(0)$ and in the image (distance $\boldsymbol{\delta}\mathbf{x}$ in pixel unit), for various apertures. μ, the control law gain (16), is set the highest without making oscillations around the optimum, whatever the ϕ considered, *i.e.* $\mu = 0.08$. Finally, iterations are counted until convergence of the control law, *i.e.* when the $\mathscr{C}_d$ cost (15) falls below 0.01.

With aperture $\phi = 8$, DDVS only corrects an error of a few pixels as PVS (not shown for conciseness and well known). Indeed, the large depth-of-field due to $\phi = 8$ prevents observations of defocus variations for small variations of t_Z.

The large aperture $\phi = 0.95$ leads DDVS to correct much larger $\boldsymbol{\delta}\mathbf{x}$ errors than when $\phi = 8$: *13 times more*. DDVS first controls the camera backward (Fig. 2) to increase defocus such that image signals at current and desired poses overlap enough. Then, DDVS moves the camera to correct t_X and t_Z to get closer to $\mathbf{X}$ such that the $\mathscr{C}_d$ cost (15) is minimized.

Surprisingly, when PVS considering t_Z and t_X is basically applied to defocused images, it moves the camera backward very similarly to first iterations of DDVS, even if defocus is not taken into account in the control law (3). But once reached a depth big enough to make current and desired images overlap, the motion is much slower than DDVS, then not stable near the optimum. Thus, final $\boldsymbol{\delta}\mathbf{x}$ does not fall below 2.9 pixels (4 mm in space) whereas, DDVS features a final $\boldsymbol{\delta}\mathbf{x}$ of 0.1 pixel (0.5 mm in space, Fig. 2).

All above mentioned results are with known Z whereas, in practice, no depth measure is available with a color or grayscale camera. Then, simulations with the same parameters than those reported in Table I are led, but with constant $Z = Z^*$. Same observations as when Z is known are made. The main difference for DDVS is an increase of iterations to converge: +15 with F-8, +17 with F-0.95, +20 with F-0.1.

2) Very large camera aperture: The bottom of Table I reports simulations with the very large camera aperture $\phi = 0.1$. With this aperture, DDVS shows a huge convergence domain: 75% of the entire field-of-view (5% when $\phi = 0.95$). In that case, t_X^* and $t_X(0)$ are set as far as possible. Figure 3 shows the behavior of DDVS with the

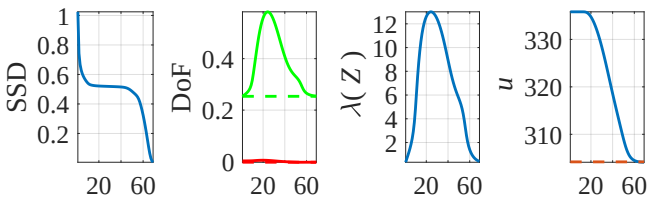


Fig. 2: DDVS simulation: 2 DoF, F-0.95. (a) From left to right: evolution along iterations of the cost, DoF in m (t_Z in green with t_Z^* dashed green; t_X in red with t_X^* dashed red), defocus spread (11), location (blue) in the image (pixel units) of current $pr(\mathbf{X})$ (see (12)) and the desired one (dashed red).

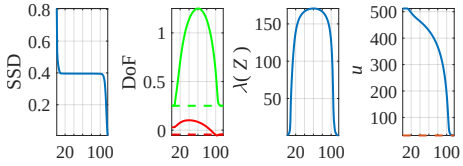


Fig. 3: DDVS maximum initial error allowing convergence with F-0.1 (same legends of plots as for Figure 2).

latter settings and with constant $Z = Z^*$ (DDVS behaves very similarly when knowing Z but with 20% less iterations).

Although such $\phi = 0.1$ aperture is not yet available among lenses of machine vision cameras, DDVS results with such setting may open future investigations on the digital processing side, *e.g.* a new PGM VS with λ depending on $\mathbf{p}$. This idea is left as future works.

To conclude the simulations, one must note DDVS achieves significant motion along the Z axis in order to increase enough the defocus to converge. This is interesting to allow convergence where it was not possible with PVS but the resulting camera trajectory is not straight. PGM VS is expected to achieve straighter trajectories, however at the price of well setting the initial value of the extra DoF λ (Sec. II-B) as real experiments show (Sec. V).

V. EXPERIMENTAL RESULTS

Experimental results of DDVS, PVS and the state-of-the-art PGM VS [6] with a camera embedded on a robot arm (Fig. 1a) are reported. Four, then six, DoF are controlled in planar and non-static 3D scenes (background change).

A. Experimental setup

Experiments use a 6 DoF Universal Robot 10 arm with a Flir FL-U3-13E4C camera on its end-effector (Fig. 1a). A Yakumo lens (17 mm focal length; maximum aperture F-0.95) equips the camera, set to acquire 30 images per second. It is connected to a laptop (Intel Core i7-7700HQ Central Processing Unit, CPU). The Graphics Processing Unit (GPU) is not used. Velocities computed with control laws (3), (8), (16) are sent to the robot through wired network. Camera intrinsic parameters are got from datasheets: $\gamma = \{17$ mm, 5.3 μm, 320 pixels, 256 pixels, ϕ, $Z_f\}$.

Four scenes are considered, two with the photograph of Bacall, Monroe and Bogart (Fig. 1a, 1b and 8a) used in every DVS paper. The third is textureless, only featuring a black hex key on a uniform clear background (Fig. 4a). Finally, the fourth scene is 3D and made of food boxes and drinks (Fig. 1a on the red table: Fig. 5a) at various depths.

PVS implementation is the one of the ViSP (`https://visp.inria.fr`) C++ library. The same is applied whatever the camera aperture setting.

Then, we reimplemented PGM VS on CPU only and reached similar processing time performances than [6] did on GPU (about 10 Hz with 100×100 pixels images) thanks to the most precomputations that we could do, requiring λ to be constant within PGM VS' Step 1. As acquired images are of 512×640 pixels, they are resized by a factor $\alpha \in \mathbb{N}_+$

used to update camera intrinsic parameters $k_{\mathbf{u}}$, u_0 and v_0. Contrary to [6], the switch from Step 1 to Step 2 (Sec. II-B) occurs when the residual is stable enough. In all experiments considering 4 DoF, the residual stability threshold is set to 10^{-3}, and $5 \cdot 10^{-4}$ when 6 DoF are controlled. These slight changes with respect to the original implementation of PGM VS allow converging from farther poses than those reported in [6] (see Sec. V-C and V-E).

Finally, DDVS is implemented extending the *ViSP luminance feature* of PVS with image Laplacian and rewriting the interaction matrix as (20).

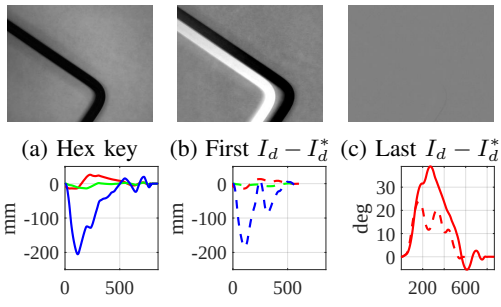
B. Lateral initial error

This set of experiments controls 4 DoF of the robot arm in order to highlight the behavior of DDVS and PVS (basically applied to defocused images) without DoF coupling issues (discussed in Sec. V-E). Similarly to simulations (Sec. IV-B), only the initial $t_X(0)$ is different from the desired pose $\mathbf{p}^*$.

We report experiments with aperture $\phi = 0.95$, focus distance is $Z_f = 25$ cm and $Z = Z^* = Z_f$ constantly, *i.e.* the desired image is acquired fronto-parallel to the almost flat scene featuring a hex key (Fig. 4a) and focused. Setting $t_X(0) = 10$ mm leads to a lateral shift of about 40 pixels in the image (Fig. 4b). Gains μ of control laws (3), (16) are set the highest avoiding oscillations at convergence.

First, contrary to simulations, PVS converges (Fig. 4c) precisely as DDVS (below 0.1 mm of 3D residual error). This is obviously due to the fact that simulations considered a scene of a single 3D point whereas real scenes are dense.

DDVS converges faster (609 iterations) than PVS (809 iterations) and, comparing the evolution of their DoF (Fig. 4d), it is clear PVS deviates more than DDVS on every DoF. Quantitatively, intervals of camera position and orientation are [40.1 mm, 17.1 mm, 225.0 mm, 44.4 degrees] for PVS and [27.8 mm, 8.4 mm, 192.8 mm, 24.6 degrees] for DDVS. Thus, DDVS deviation is about half the PVS one, except along the Z axis as sufficient motion must be done on that axis in order to defocus enough images to converge. But DDVS, by explicitly considering the defocus term in its interaction matrix, needs less backward motion (-183.5 mm) than PVS (-205.2 mm) to get in the convergence domain.



(a) Hex key (b) First $I_d - I_d^*$ (c) Last $I_d - I_d^*$

(d) Positions (Left: PVS; center: DDVS): red, X; green, Y; blue, Z. Right: DDVS (dashed) and PVS orientations.

Fig. 4: (a) Desired image. (b) initial and (c) final images of differences surrounding (d) the DoF evolution over iterations.

C. Convergence domain extents

With same settings as in Section V-B, but looking at Monroe's face (Fig. 1b) PVS with $\phi = 8$ and $\phi = 0.95$, PGM VS with $\phi = 8$ and DDVS with $\phi = 0.95$ are run from a set of initial $t_Z(0)$ ranging from 3 cm to 45 cm. Control law gains μ are set to avoid oscillations around the optimum. μ is set once per VS method from $t_Z(0) = 3$ cm. Then, for other $t_Z(0)$, μ is unchanged. This is the only setting for PVS and DDVS, whereas PGM VS had to be run several times to find the ideal value of λ for the Step 1 of PGM VS (Sec. II-B).

Table II shows the maximum $t_Z(0)$ for which each considered VS converged to the desired pose as well as the number of iterations, *i.e.* the number of images acquired until convergence. In a few words, thanks to its two steps, PGM VS performs the best in terms of maximum initial error and absolute number of iterations, at the price of longer setting time to find the ideal $\lambda = \lambda^* = 12$. Such value requires an image resize parameter $\alpha = 8$ (so 80×64 pixels are considered) to keep the control loop rate above 10 Hz.

Then, as expected from the litterature, PVS features the weakest convergence domain for a lot of iterations (12 minutes to correct 27 cm). Interestingly, its rough application to defocused images increases 22% its convergence domain to reach 78.5% of the PGM VS one (in similar durations).

Finally, DDVS is ranked second, PGM VS convergence domain still being 15% larger. This second place must be balanced by the fact that no extra parameter needs to be set for DDVS, contrary to the λ of PGM VS.

D. Robustness to non-static 3D scene

Still controlling 4 DoF, the scene of 3D objects is considered (Fig. 1a, background: Fig. 5a). As Section V-C shows PGM VS and DDVS are the two best DVS among those evaluated, this section considers them only.

This set of experiments shows that some content of a 3D scene impacts more PGM VS than DDVS. If the background is dark, both PGM VS and DDVS converge from $t_Z(0) = 10$ cm (the initial values of the 5 other DoF are not changed with respect to the desired ones). DDVS takes 155 iterations to converge while PGM VS takes 120 ($\lambda = 2$ for Step 1). However, with a bright background, PGM VS converges from $t_Z(0) = 5$ cm at most, for $\lambda = 2$ or other values, whereas DDVS still converges from $t_Z(0) = 10$ cm. Figure 5 shows the desired and initial images for $t_Z(0) = 10$ cm along with initial PGM error and the one when PGM VS is stuck in a local minimum (about 45 degrees of rotation error).

This surprising poor behavior of PGM VS regarding bright backgrounds is confirmed when the background is changed from dark to bright, between the acquisition of desired and

TABLE II: Comparison of the 4 DoF convergence domain extents with only $t_Z \neq t_Z^*$ for the considered DVS.

ϕ	DVS	μ	$t_Z(0)$ max	iterations
8	PVS	4	27 cm	21711
8	PGM VS	1	42 cm	759
0.95	PVS	4	33 cm	809
0.95	DDVS (ours)	4	36 cm	1096

(a) Initial I (b) I^* (c) First $G-G^*$ (d) Last $G-G^*$

Fig. 5: PGM VS in a scene of white background. (d) means the PGM error when PGM VS is stuck in a local minimum.

current images (Fig. 6). Indeed, PGM VS (Fig. 6a to 6d) stops in a local minimum. Inversely, DDVS (Fig. 6e to 6h) reaches the global minimum. Figure 6h allows seeing the background differences (left and top right). Recall the image of differences almost uniformly features the median gray level when the scene does not change (*e.g.* Fig. 4c). Changing the background from bright to dark is better dealt by PGM VS but still perturbs its precision at convergence: 10 mm of translation residual error (slightly less than 1 mm for DDVS).

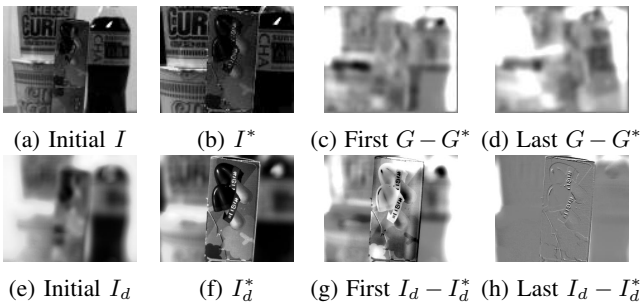
Other DDVS experiments were done from $t_Z(0) = 10$ cm, changing the bottle and one food box between acquisitions of desired and current images. Trajectories are slightly different but precision at convergence are as with constant scene, confirming DDVS' robustness to scene content changes.

E. Experiments with 6 DoF

The last set of experiments considers the control of the robot arm 6 DoF. Two desired poses are considered in front of the same planar scene: 1) $Z^* = 25$ cm to frame Monroe's face only (Fig. 1b); 2) $Z^* = 50$ cm (Fig. 8a).

In either case, PGM VS and DDVS first show a much tighter convergence domain than with 4 DoF. Indeed, even if initial and desired poses are solely separated by $t_Z \neq 0$, starting beyond $t_Z(0) = 2.5$ cm makes PGM VS to fail and DDVS fails when $t_Z(0) > 1.5$ cm. With $t_Z(0) \leq 1.5$ cm, trajectories of DDVS are very far from the straight line while those of PGM VS are much closer, but less precise at convergence (1.2 mm of final error) than DDVS (0.2 mm). With $t_Z(0) = 2.5$ cm, PGM VS trajectory keeps almost straight (t_X and t_Y stay in a square of 3 mm side) with a final error of 1.2 mm.

It turns out that these DVS are badly conditioned as the conditioning of interaction matrices, when $t_Z(0) = 1.5$ cm, varies from 137.9 to 146.7 for PGM VS and from 2162.5 to 2308.7 for DDVS. As this appears when activating rotations



(a) Initial I (b) I^* (c) First $G-G^*$ (d) Last $G-G^*$

(e) Initial I_d (f) I_d^* (g) First $I_d - I_d^*$ (h) Last $I_d - I_d^*$

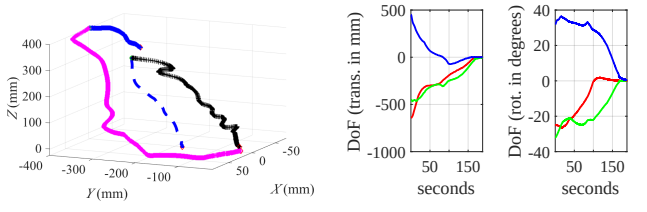
Fig. 6: Moving a chair changes the background from dark, in the desired image, to white, in current images. PGM VS: local minimum (d). DDVS: global minimum (h).

around X and Y axes of the camera, it is obvious that it is related to the well known strong coupling between rotation around X and translation along Y and inversely. To our knowledge, such poor conditioning was never mentioned in previous DVS works. One of the reasons this work observes it might be the use of a rather long focal length. Unfortunately [2], [6] do not provide such detail to confirm.

The long focal length leads to a narrow field of view. It prevents distinguishing coupled camera motions in the image [29, Ch. 15]. This problem is linked to motion perceptibility [30] that, when poor, makes the condition number of the interaction matrix large. To solve this issue, inspiration comes from the camera motion estimation from image points in two views [31, Ch. 4.4.4]. To pre-condition the estimation, one first translates image points to put their centroid at the origin of the image frame. Then, a scaling makes "the average distance of an image point from the origin equal to $\sqrt{2}$" [31, p. 107]. As a DVS considers every pixel, their centroid is already the image center. Scaling the coordinates of every pixel without extra processing can be done by artificially scaling f (focal length).

By doing so with a factor of 0.1, experimentally found but unique, mean condition numbers are divided by 15 for PGM VS and 97 for DDVS. Maximum $t_Z(0)$ allowing convergence become 31 cm for PGM VS and 21 cm for DDVS. From $t_Z(0) = 21$ cm, trajectories deviate from the straight line by 15 mm for PGM VS and 75 mm for DDVS, on average.

Then, the largest initial errors with coupled DoF were looked for, considering $Z^* = Z_f = 50$ cm (Fig. 8a). Figure 7a shows trajectories where PGM VS took 33 s to converge from the initial error $\boldsymbol{\delta}\mathbf{p}_G$ =[0 cm, -30 cm, 30 cm, -20 degrees, 0 degrees, 0 degrees] (visual error in Fig. 8g), setting its new record in translation with 42.4 cm (compared to 30.3 cm in [6]). On its side, DDVS succeeded with even larger initial errors such as $\boldsymbol{\delta}\mathbf{p}_D$ =[0 cm, -40 cm, 40 cm, 27 degrees, 0 degrees, 0 degrees] (in 88 s, initial visual error in Fig. 8h), $\boldsymbol{\delta}\mathbf{p}'_D$ =[-64 cm, -45 cm, 45 cm, 32 degrees, -32 degrees, -25 degrees] (in 184 s, largest initial translation magnitude coupled with full 3D rotation: 90.2 cm) and $\boldsymbol{\delta}\mathbf{p}''_D$ =[-64 cm, -45 cm, 25 cm, 37 degrees, -37 degrees, -25 degrees] (in 202 s, largest initial rotation magnitude coupled with full 3D translation: 58.0 degrees). DDVS results from $\boldsymbol{\delta}\mathbf{p}'_D$ (initial image on Fig. 8c and DoF evolution on Fig. 7b) and $\boldsymbol{\delta}\mathbf{p}''_D$ (initial image on Fig. 8d)



(a) Initial $\boldsymbol{\delta}\mathbf{p}_D$ for DDVS (pink) and PVS (blue), $\boldsymbol{\delta}\mathbf{p}_G$ for PGM VS (black) and PVS (dashed).

(b) Left: t_X, red; t_Y, green; t_Z, blue. Right: θw_X, blue; θw_Y, green; θw_Z, red.

Fig. 7: (a) Trajectories (desired pose: $\mathbf{0}$), (b) 6 DoF over time for DDVS with initial error $\boldsymbol{\delta}\mathbf{p}'_D$.

(a) $I_d^* = I^*$ (b) $I_d(\boldsymbol{\delta}\mathbf{p}_D)$ (c) $I_d(\boldsymbol{\delta}\mathbf{p}'_D)$ (d) $I_d(\boldsymbol{\delta}\mathbf{p}''_D)$

(e) $I(\boldsymbol{\delta}\mathbf{p}_G)$ (f) $I(\boldsymbol{\delta}\mathbf{p}_G) - I^*$ (g) $G(\boldsymbol{\delta}\mathbf{p}_G) - G^*$ (h) $I_d(\boldsymbol{\delta}\mathbf{p}_D) - I_d^*$

Fig. 8: Desired (a) and initial images for which (b-d) only DDVS succeeds, (e) DDVS and PGM VS succeed with some initial differences for PVS (f), PGM VS (g), DDVS (h).

are shown in the accompanying video[4]. PGM VS diverges from $\boldsymbol{\delta}\mathbf{p}_D$ and beyond where the initial image captures the bright scene around the photograph of the actors (*e.g.* Fig. 8b, top). This, and an extensive study [32] with large, randomly drawn, initial errors on all 6 DoF confirm the trend about the sensitivity of PGM VS to a bright background (Sec. V-D). PVS (pre-conditioned) was applied from both $\boldsymbol{\delta}\mathbf{p}_G$ and $\boldsymbol{\delta}\mathbf{p}_D$ but stops on local minima, far from desired poses (Fig. 7a).

DDVS converges from larger initial errors than those reported for best DVS on this criterion [10], [6], [12]: up to 3 times the translation magnitude of [6] and 1.7 times the rotation magnitude of [12]. DDVS converges also from 1.7 times larger initial translation errors than intensity-based VS [13], for a slightly larger rotation magnitude (58 degrees versus 57 [13]). However, DDVS trajectories are curvier and its time duration to converge is longer than those of [13].

Finally, as in this last experiment $Z^* = Z_f = 50$ cm, it also shows that exploiting defocus extends a lot the DVS convergence domains for other focus depths than the shortest.

VI. CONCLUSION AND FUTURE WORKS

This paper introduced the new Direct Visual Servoing exploiting defocus. As Photometric Visual Servoing, submillimetric precision is reached. Its convergence domains are competing and can significantly overperform those of the state-of-the-art Photometric Gaussian Mixtures-based Visual Servoing. This is obtained for a much lower processing time, *i.e.* 70 ms per image of 80×64 pixels for the latter versus 52 ms per image of 320×256 pixels for the new DVS. However, as a counterpart, trajectories are curvier.

Future works will target the control of focus depth in order to perform straight trajectories, faster. Investigations will also strike a general solution to DVS pre-conditioning.

REFERENCES

[1] F. Chaumette and S. Hutchinson, "Visual servo control. I. Basic approaches," *IEEE Rob. & Autom. Mag.*, vol. 13, pp. 82–90, 2006.

[2] C. Collewet and E. Marchand, "Photometric visual servoing," *IEEE T. on Rob.*, vol. 27, no. 4, pp. 828–834, 2011.

[3] V. Kallem, M. Dewan, J. P. Swensen, G. D. Hager, and N. J. Cowan, "Kernel-based visual servoing," in *IEEE/RSJ Int. C. on Intel. Robots and Systems*, 2007, pp. 1975–1980.

[4] E. Marchand and C. Collewet, "Using image gradient as a visual feature for visual servoing," in *IEEE/RSJ Int. C. on Intel. Robots and Systems*, 2010, pp. 5687–5692.

[4] http://mis.u-picardie.fr/~g-caron/videos/ds.mp4

[5] L. Cui, E. Marchand, S. Haliyo, and S. Régnier, "Hybrid automatic visual servoing scheme using defocus information for 6-DoF micropositioning," in *IEEE Int. C. on Rob. and Autom.*, 2015, pp. 6025–6030.

[6] N. Crombez, E. Mouaddib, G. Caron, and F. Chaumette, "Visual servoing with photometric gaussian mixtures as dense features," *IEEE T. on Rob.*, vol. 35, no. 1, pp. 49–63, 2019.

[7] S. K. Nayar, S. A. Nene, and H. Murase, "Subspace methods for robot vision," *IEEE T. on Rob. and Autom.*, vol. 12, no. 5, pp. 750–758, 1996.

[8] K. Deguchi, "A direct interpretation of dynamic images and camera motion for vision guided rob." in *IEEE/SICE/RSJ Int. C. on Multisensor Fusion and Integr. for Intel. Systems*, 1996, pp. 313–320.

[9] E. Marchand, "Subspace-based direct visual servoing," *IEEE Rob. and Autom. Letters*, vol. 4, no. 3, pp. 2699–2706, 2019.

[10] M. Bakthavatchalam, O. Tahri, and F. Chaumette, "A Direct Dense Visual Servoing Approach using Photometric Moments," *IEEE T. on Rob.*, vol. 34, no. 5, pp. 1226–1239, 2018.

[11] L.-A. Duflot, R. Reisenhofer, B. Tamadazte, N. Andreff, and A. Krupa, "Wavelet and Shearlet-based Image Representations for Visual Servoing," *The Int. J. of Rob. Research*, vol. 38, no. 4, pp. 422–450, 2019.

[12] E. Marchand, "Direct visual servoing in the frequency domain," *IEEE Rob. and Autom. Letters*, vol. 5, no. 2, pp. 620–627, 2020.

[13] G. Silveira, L. Mirisola, and P. Morin, "Decoupled intensity-based nonmetric visual servo control," *IEEE T. on Control Syst. Tech.*, vol. 28, no. 2, pp. 566–573, 2020.

[14] H.-Y. Lin and K.-D. Gu, "Photo-realistic depth-of-field effects synthesis based on real camera parameters," in *Advances in Visual Computing*, 2007, pp. 298–309.

[15] T. Hach, J. Steurer, A. Amruth, and A. Pappenheim, "Cinematic bokeh rendering for real scenes," in *ACM Eur. C. on Visual Media Prod.*, 2015, pp. 1–10.

[16] B. Zhang, B. Sheng, P. Li, and T. Lee, "Depth of field rendering using multilayer-neighborhood optimization," *IEEE T. on Visualization and Computer Graphics*, vol. 26, no. 8, pp. 2546–2559, 2020.

[17] M. Subbarao and G. Surya, "Depth from defocus: A spatial domain approach," *Int J Comput Vision*, vol. 13, pp. 271–294, 1994.

[18] E. Alexander, Q. Guo, S. Koppal, S. Gortler, and T. Zickler, "Focal flow: Velocity and depth from differential defocus through motion," *Int. J. of Computer Vision*, vol. 126, pp. 1062–1083, 2018.

[19] M. Maximov, K. Galim, and L. Leal-Taixe, "Focus on defocus: Bridging the synthetic to real domain gap for depth estimation," in *IEEE/CVF C. on Computer Vision and Pattern Recogn.*, 2020.

[20] I. R. Nourbakhsh, D. Andre, C. Tomasi, and M. R. Genesereth, "Mobile robot obstacle avoidance via depth from focus," *Rob. and Autonomous Systems*, vol. 22, no. 2, pp. 151 – 158, 1997.

[21] T. Shiozaki and G. Dissanayake, "Eliminating scale drift in monocular slam using depth from defocus," *IEEE Rob. and Autom. Letters*, vol. 3, no. 1, pp. 581–587, 2018.

[22] M. Wang, X. Lv, and X. Huang, "Self-optimizing visual servoing control for microassembly robotic depth motion," in *Int. C. on Information Acquisition*, 2007, pp. 482–486.

[23] D. Hong, F. Janabi-Sharifi, and H. Cho, "An adaptive depth of field imaging system for visual servoing," in *IFAC World Congress*, vol. 41, no. 2, 2008, pp. 5405–5410.

[24] B. K. Horn and B. G. Schunck, "Determining optical flow," *Artif. Intell.*, vol. 17, no. 1, pp. 185 – 203, 1981.

[25] N. Salvaggio, *Basic Photographic Materials and Processes, 3rd Edition*, L. Stroebel and R. Zakia, Eds. Focal Press, 2013.

[26] T. Iijima, "Theory of pattern recognition," *Electronics and Communications in Japan*, pp. 123–134, Nov. 1963.

[27] D. Forsyth and J. Ponce, *Computer Vision: A Modern Approach, 2nd Edition*. Pearson, 2012.

[28] P. Bergmann, R. Wang, and D. Cremers, "Online photometric calibration of auto exposure video for realtime visual odometry and SLAM," *IEEE Rob. and Autom. Letters*, vol. 3, pp. 627–634, 2018.

[29] P. Corke, *Robotics, Vision and Control: Fundamental Algorithms In MATLAB*, 2nd ed. Springer, 2017.

[30] R. Sharma and S. Hutchinson, "Motion perceptibility and its application to active vision-based servo control," *IEEE T. on Robotics and Automation*, vol. 13, no. 4, pp. 607–617, 1997.

[31] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, 2004.

[32] G. Caron, "Defocus-based direct visual servoing - addendum," https://hal.archives-ouvertes.fr/hal-03161692/document, Mar. 2021.