

# Direct 3D model-based tracking in omnidirectional images robust to large inter-frame motion

Seif-Eddine Guerbas, Nathan Crombez, Guillaume Caron, El Mustapha Mouaddib

▶ **To cite this version:**

Seif-Eddine Guerbas, Nathan Crombez, Guillaume Caron, El Mustapha Mouaddib. Direct 3D model-based tracking in omnidirectional images robust to large inter-frame motion. IEEE International Conference on Advanced Robotics, Dec 2021, Ljubjana, Slovenia. 10.1109/ICAR53236.2021.9659324 . hal-03371352

**HAL Id: hal-03371352**
**https://hal.science/hal-03371352**

Submitted on 29 Oct 2021

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Direct 3D model-based tracking in omnidirectional images robust to large inter-frame motion

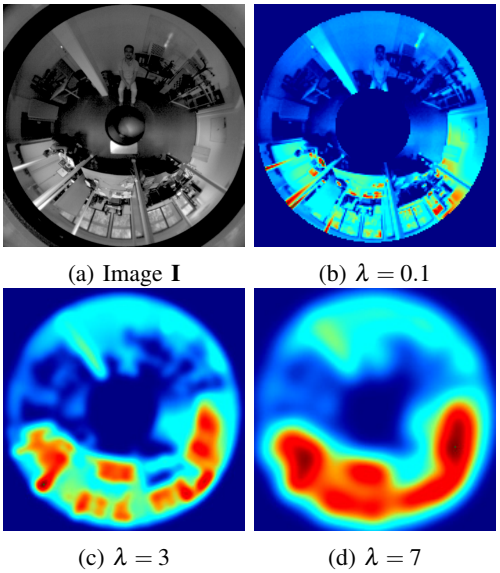
Seif Eddine Guerbas[1], Nathan Crombez[2], Guillaume Caron[1,3] and El Mustapha Mouaddib[1]

***Abstract*— This paper tackles direct 3D model-based pose tracking. It considers the Photometric Gaussian Mixtures (PGM) transform of omnidirectional images as direct features. The contributions include an adaptation of the pose optimization to omnidirectional cameras and a rethink of the initialization and optimization rules of the PGM extent. These enhancements produce a giant leap in the convergence domain width. Application to images acquired onboard a mobile robot within an urban environment described by a large 3D colored point cloud shows significant robustness to large inter-frame motion, compared to approaches that directly use pixel brightness as direct features.**

## I. Introduction

Direct image alignment [1] and direct visual servoing [2] (DVS) have significantly progressed during the last decade in their respective communities, namely computer vision and robotics. The best known direct approaches concern visual odometry [3] and visual Simultaneous Localization And Mapping [4] (SLAM). For a while, direct approaches were known to save time by avoiding features processing and to be of high accuracy whereas suffering of a narrow convergence domain [5]. While usually overcome by encapsulation in a pyramidal scheme [6], the latter narrowness was recently enlarged intrinsically by direct approaches relying on transforms of images: scale space [7], frequency domain [8], photometric moments [9] or Photometric Gaussian Mixtures [10] (PGM). The latter optimizes the Gaussian extent of PGMs (Fig. 1 shows its impact on PGM smoothness) together with camera pose degrees-of-freedom. This allows to significantly enlarge the convergence domain of DVS with conventional camera.

This paper investigates the application of PGM to omnidirectional (panoramic) vision. The motivation comes from the fact that the wider field of view of an omnidirectional camera compared to a conventional camera (perspective-like) allows more reliable 3D motion estimation [11] and higher localization rates [12]. This is due to the possibility to capture information better spread around the viewpoint. This paper not only considers the PGM of omnidirectional images but its use in a direct approach for camera and robot localization with respect to a 3D model of the environment. Interestingly, the bridge from Visual Servoing (VS) to the full scale alignment of an image on 3D model is well established as Virtual Visual Servoing [13]. It was applied to direct 3D model-based pose tracking in omnidirectional images from pixel brightness [14], [15], making full benefit of environment 3D point cloud with colors for robot localization, though sensitive to the amount of inter-frame motion.



(a) Image $\mathbf{I}$ (b) $\lambda = 0.1$ (c) $\lambda = 3$ (d) $\lambda = 7$

Fig. 1: (a) An acquired image $\mathbf{I}$ and (b-d) PGMs $\mathbf{G}(\mathbf{I}, \lambda)$ for various extents $\lambda$.

Since it is hard to find more recent works of omnidirectional image direct alignment with a 3D model of an environment, [14], [15] are considered as baselines. Despite the variety of contributions in the field of neural networks, only conventional or rectified images are considered as input of pose detection [16] approaches. One could generate conventional images from omnidirectional ones [17] to feed the latter methods but, in this paper, we focus on using acquired images directly without geometric pre-transformation. This way, one can think about considering large scale direct visual SLAM with omnidirectional images [18] but none implement yet localization within a pre-built map. The latter functionality is handled by handcrafted feature-based approaches. But

Note: Part of this work was presented orally at the 2021 IEEE CVPR workshop 3DVR without proceedings.

This work is carried out as part of the Interreg VA FCE ADAPT project "Assistive Devices for empowering disAbled People through robotic Technologies" (adapt-project.com). The Interreg FCE Programme is a European Territorial Cooperation programme that aims to fund high quality cooperation projects in the Channel border region between France and England. The Programme is funded by the European Regional Development Fund (ERDF).

The authors wish to thank A. Rabhi for the experiments with the vehicle.

[1]Université de Picardie Jules Verne, MIS laboratory, 33 rue Saint Leu, 80039 Amiens Cedex 1, France {seif-eddine.guerbas, guillaume.caron, mouaddib}@u-picardie.fr

[2]Université Bourgogne Franche-Comté, UTBM, CIAD Laboratory, 90010 Belfort, France nathan.crombez@utbm.fr

[3] CNRS/AIST JRL, Tsukuba Central 1, 1-1-1 Umezono, Tsukuba, Ibaraki 305-8560, Japan

even the state-of-the-art ORB-based visual SLAM [19] fails in localizing acquired omnidirectional images in a map that has been pre-built from another camera, thus making hard to share the map and requiring additional sensors to estimate the full scale 3D pose of the camera.

The contributions of this paper benefit from both the use of a pre-built 3D model of an environment and the properties of omnidirectional vision and are summarized as:

- PGM adaptation to the omnidirectional camera;
- a new solution of initialization and optimization of the PGM extent, enlarging the convergence domain;
- a 3D model-based visual tracking approach robust to very large inter-frame motion.

The rest of the paper shortly describes the PGM-based omnidirectional visual servoing, focusing on the contribution regarding the new rules of initialization and optimization of the Gaussian extent of PGMs. Then, Section III reports their evaluation in a virtual environment made of 3D scans of streets. Finally, Section IV presents results of direct 3D model-based tracking in omnidirectional images transformed as PGMs, acquired aboard a mobile robot and a vehicle driven in streets of a city for hundreds of meters, before conclusion (Sec. V).

## II. PGM for Omnidirectional Visual Servoing

### A. Photometric Gaussian Mixture

From an omnidirectional image $\mathbf{I}$ with $M \in \mathbb{N}$ pixels, we express its PGM in the exact same way as [10] did for conventional images, *i.e.*, as a mixture of $M$ Gaussians, sharing a unique Gaussian extent parameter $\lambda \in \mathbb{R}_+^*$, weighted by brightness $I(\mathbf{u})$, for the $M$ pixels of coordinates $\mathbf{u} = (u,v) \in \Omega \subset \mathbb{N}^2$ of the image. To distinguish image coordinates from PGM coordinates, we write the latter $\mathbf{u_g} = (u_g, v_g) \in \Omega$, leading to express a PGM sample as:

$$G(\mathbf{u_g}, \mathbf{I}, \lambda) = \sum_{\mathbf{u}} I(\mathbf{u}) \exp\left(-\frac{(u_g-u)^2+(v_g-v)^2}{2\lambda^2}\right). \quad (1)$$

The PGM of an image $\mathbf{I}$ is written $G(\mathbf{I}, \lambda)$, for compactness. Figure 1 shows the impact of $\lambda$ on the PGM.

### B. PGM-based omnidirectional visual servoing

Visual servoing is similar to a Gauss-Newton optimization that computes camera pose increments $\mathbf{v} \in \mathbb{R}^6$, minimizing the error between a reference (desired) image $\mathbf{I}^*$ and the one to align, namely the current image $\mathbf{I}$. $\mathbf{I}$ is acquired at pose $\mathbf{r} = (t_X, t_Y, t_Z, \theta w_X, \theta w_Y, \theta w_Z) \in \mathbb{R}^6$, with $||[w_X, w_Y, w_Z]|| = 1$ and $\theta \in [-\pi, \pi]$ representing the rotation as axis-angle. $\mathbf{I}^*$ is acquired at pose $\mathbf{r}^* \in \mathbb{R}^6$. Then, highlighting the dependence of PGMs to the camera pose as $G(\mathbf{r}, \mathbf{I}, \lambda)$ for $\mathbf{I}$ and $G(\mathbf{r}^*, \mathbf{I}^*, \lambda^*)$ for $\mathbf{I}^*$, and by stacking all their samples (1) as, respectively, $\mathbf{G}(\mathbf{r}, \lambda) \in \mathbb{R}^M$ and $\mathbf{G}^* \in \mathbb{R}^M$, we express the error vector to regulate to zero:

$$\mathbf{e}(\mathbf{r}, \lambda) = \mathbf{G}(\mathbf{r}, \lambda) - \mathbf{G}^* \in \mathbb{R}^M. \quad (2)$$

In (2), the Gaussian extent $\lambda$ is variable while $\lambda^*$ of $\mathbf{G}^*$ is not [10], thus they are possibly different. $\lambda$ is not constant because it is optimized in addition to $\mathbf{r}$ by a Gauss-Newton method computing iteratively increments as [10]:

$$[\mathbf{v}^T, \dot{\lambda}]^T = -\mu \, [\mathbf{L_G} \;\; \mathbf{J}_\lambda]^+ \mathbf{e}(\mathbf{r}, \lambda), \quad (3)$$

where $[\,]^+$ is the pseudo-inverse operator, $\mathbf{L_G} \in \mathbb{R}^{M \times 6}$ is the interaction matrix related to $G(\mathbf{r}, \mathbf{I}, \lambda)$ at pose $\mathbf{r}$ and $\mathbf{J}_\lambda \in \mathbb{R}^{M \times 1}$ is the Jacobian of $G(\mathbf{r}, \mathbf{I}, \lambda)$ with respect to $\lambda$. $\mathbf{J}_\lambda$ is the same as in [10]. However, $\mathbf{L_G}$ is now expressed for the unified central camera projection model (UCM) [20] instead of the perspective one used in [10]. Considering intrinsic parameters $\alpha_u \in \mathbb{R}^*$, $\alpha_v \in \mathbb{R}^*$ as the generalized focal length, $u_0 \in \mathbb{R}$, $v_0 \in \mathbb{R}$ as the principal point coordinates and $\xi \in \mathbb{R}$ as the mirror shape parameter, the UCM relates 3D points $\mathbf{X} = [X, Y, Z]^T \in \mathbb{R}^3$ to digital image points $\mathbf{u_g}$ as:

$$u_g = \alpha_u x_g + u_0 \text{ and } v_g = \alpha_v y_g + v_0, \quad (4)$$

with $x_g = X/(Z + \xi\rho)$, $y_g = Y/(Z + \xi\rho)$ and $\rho = \sqrt{X^2+Y^2+Z^2}$. Then, each line $\mathbf{L}_G$ of $\mathbf{L_G}$ is expressed as:

$$\mathbf{L}_G = \left(\sum_{\mathbf{u}} \nabla_{\mathbf{u_g}} G\right) \begin{bmatrix} \alpha_u & 0 \\ 0 & \alpha_v \end{bmatrix} \mathbf{L}_{\mathbf{x_g}}, \quad (5)$$

where:

$$\mathbf{L}_{\mathbf{x_g}} = \begin{bmatrix} -\frac{1+x_g^2(1-\xi(\gamma+\xi))+y_g^2}{\rho(\gamma+\xi)} & \frac{\xi x_g y_g}{\rho} \\ \frac{\xi x_g y_g}{\rho} & -\frac{1+y_g^2(1-\xi(\gamma+\xi))+x_g^2}{\rho(\gamma+\xi)} \\ \frac{\gamma x_g}{\rho} & \frac{\gamma y_g}{\rho} \\ x_g y_g & \frac{(1+y_g^2)\gamma - \xi x_g^2}{\gamma+\xi} \\ -\frac{(1+x_g^2)\gamma - \xi y_g^2}{\gamma+\xi} & -x_g y_g \\ y_g & -x_g \end{bmatrix}^T, \quad (6)$$

with $\gamma = \sqrt{1+(1-\xi^2)(x_g^2+y_g^2)}$ [14]. This is the key difference in the computation of $\mathbf{L_G}$ compared to [10].

### C. The two stages strategy: new rules

In (3), $\lambda$ is optimized for the ideal behavior of PGM VS [10], *i.e.*, a large convergence domain (large $\lambda$) and a high precision at convergence ($\lambda$ tends to $\lambda^*$, set small). To achieve this ideal behavior, [10] reports a sequence of two PGM VS, that we name here *Rule0*: Step 1 with a large $\lambda^*$ (exact value depends on experiments) and $\lambda = \alpha\lambda^*$, with $\alpha = 2$, at initialization; Step 2 with constant $\lambda = \lambda^* = 1$.

In our experimental convergence study (Sec. III), we observed that setting $\lambda = 2\lambda^*$ at the initialization of Step 1 may lead to unexpected divergence. As such setting can lead to a current and a desired PGM of very different orders of magnitude, we assume it is the cause of the problem.

To simplify, we propose to remove factor $\alpha$ and set $\lambda = \lambda^*$ large at the initialization of Step 1. We name this new rule *Rule1*. Then, we define *Rule2* with Step 2 initialized with $\lambda = \lambda^* = 1$ (or smaller) and $\lambda$ optimized for more coherence with respect to Step 1, more freedom and to use the exact same control law (3). For all rules, the switch from Step 1 to Step 2 occurs after a fixed number of iterations $n_{it} \in \mathbb{N}^*$ to ease the comparisons. In the rest of this paper, $n_{it} = 120$ and Step 2 runs for 130 iterations. These values

are chosen to ensure every compared method converged, for fair comparison and should not be considered as ideal values, generally speaking. The following Section III validates the adaptation of the PGM to omnidirectional cameras and evaluates the latter introduced rules.

## III. Evaluation in a virtual environment

The virtual environment is a point cloud of four streets in the city of Amiens, France. The point cloud is a registration of thirteen 3D scans with Red-Green-Blue photographic colors, all acquired with a Lidar scanner Faro Focus 3D. The virtual camera simulates the UCM (Sec. II-B) implemented with a vertex shader in the Unity 3D software (http://unity.com) bridged to our C++ implementation. Camera intrinsic parameters match those of a real catadioptric camera (Sec. IV) calibrated classically by observing known chessboards [21].

This evaluation compares the PGM omnidirectional VS (PGMoVS) with *Rule0*, *Rule1* and *Rule2*, and the seminal Photometric omnidirectional VS [14] (PoVS). Only virtual images are considered for fair quantitative comparison as previous works of visual odometry evaluations did [11]. Finally, camera pose increments $\mathbf{v}$ are used with the exponential map of se(3) to update the virtual camera pose at each iteration of the VS.

### A. Protocol

In order to evaluate the convergence domain, 64 initial poses $\mathbf{r}$ are generated around various desired poses $\mathbf{r}^*$ with combinations of transformations $t_X = \{-8\text{m}, 8\text{m}\}$, $t_Y = \{-2\text{m}, 2\text{m}\}$, $t_Z = \{-1.5\text{m}, 1.5\text{m}\}$, $\theta w_X = \{-10°, 10°\}$, $\theta w_Y = \{-10°, 10°\}$ and $\theta w_Z = \{-15°, 15°\}$. The initial positions form together a volume included in streets of about 12m width (Fig. 2a). For variety, we consider 7 desired poses $\mathbf{r}^*$ spread within the 3D model (Fig. 2b-2h).

For all three rules of PGMoVS, $\lambda = 15$ in order to have sufficient overlapping area between $\mathbf{G}$ and $\mathbf{G}^*$. The control gain $\mu$ in (3) is set to 0.2 for every VS.

### B. Results

Figure 3 shows the cumulative distribution of the 448 final position errors for each of the four VS compared. Setting a convergence threshold equal to 2cm, *Rule2* achieves a 97% success rate while *Rule1* and *Rule0* respectively achieve 78% and 70%. For *Rule1* and *Rule0*, Step 2 may converge to a local minimum, whereas for *Rule2* most of the local minima are removed, allowing to reach the global minimum. In contrast, all the PoVS diverge, which is not surprising as initial errors allowing PoVS to converge are known to be below 1.3m [14] whereas initial errors are greater than 8m in this evaluation.

These results show that the new *Rule2* significantly outperforms the previous *Rule0* and is hence, considered for the next experiment of 3D tracking (Sec. IV).

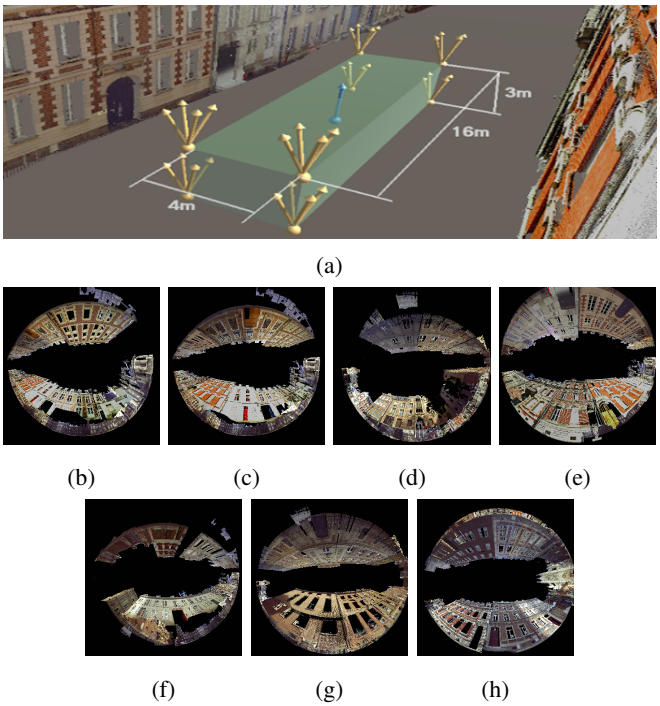



Fig. 2: Convergence domain evaluation: (a) Visualization of a parallelepiped formed by 64 initial camera poses (gold arrows) around a desired one (blue arrow); (b-h) Images rendered at the seven desired poses.

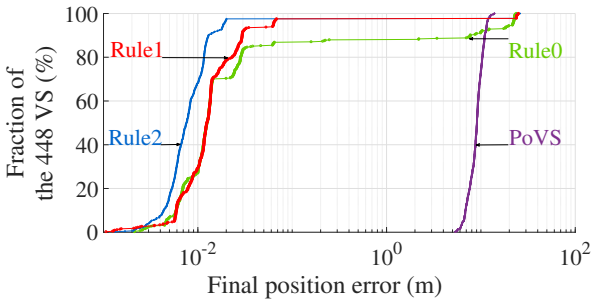
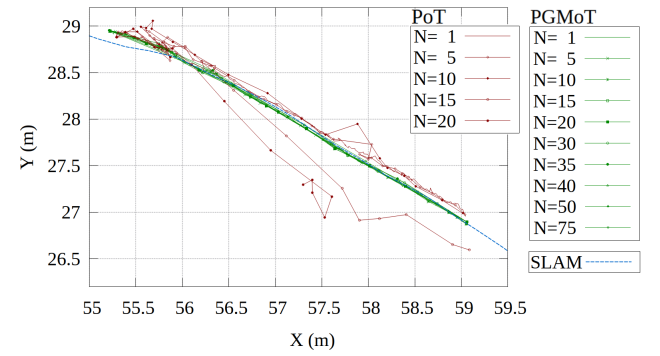


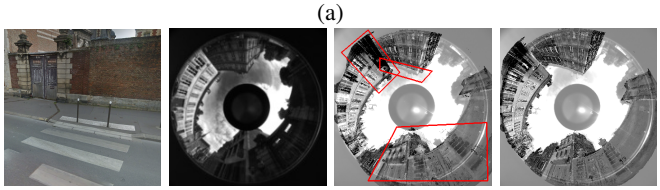

Fig. 3: Cumulative distribution of position errors.

## IV. Experiment: tracking by alignments

To perform direct 3D model-based tracking using omnidirectional images transformed as PGMs, we compute the desired PGM ($\mathbf{G}^*$ in (3)) from an image captured by an actual camera. The current PGM, $\mathbf{G}$, is computed from an omnidirectional image rendered as in Section (III). Brightness of both images are centered and normalized to improve their consistency [15]. Tracking in a sequence of acquired omnidirectional images is done by successive executions of the virtual control law (3) with *Rule2*. The camera pose for the current acquired image is initialized with the optimal pose of the previous one in the sequence and updated at each iteration of the alignment with the camera pose increments $\mathbf{v}$, thanks to the exponential map of se(3) (see Sec. III). The initial pose for the first image of the sequence is set manually. This tracking by direct alignment from PGMs (PGMoT) is




(a)



(b) (c) (d) (e)

Fig. 4: Direct 3D model-based tracking with PGMoT and PoT using 1 frame every $N$ frame(s): (a) Estimated trajectories on (b) challenging uneven ground where images as (c) are acquired; (d-e) Superimposition of the last acquired and optimal virtual images ($N = 20$): (d) PoT, (e) PGMoT.

below compared to the seminal tracking by direct alignment from brightness (PoT) within two experiments: the first (Sec. IV-A) with the camera aboard a mobile robot moving at human walking speed and the second (Sec. IV-B) aboard a vehicle driven in streets of a city.

### A. Experiment with a mobile robot

The sequence of omnidirectional images is acquired by an IDS UI-1545LE-M-GL camera (30 images per second) equipped with a RemoteReality omnidirectional catadioptric optics (double mirror). The camera was attached to a Mobile Robots Pioneer 3AT, manually piloted at walking speed (3 km/h, on average) in streets[1]. The robot embeds a SICK LMS-200 Lidar (single horizontal measurement plane) used by the robot software for SLAM, considered for qualitative (not synchronized) comparisons (Fig. 4a).

This section focuses on a sequence of about 4.35m. Although short, it is very challenging because the robot leaves a sidewalk to cross a street (Fig. 4b). The uneven ground leads the robot to shake while moving straight forward. Although the 210 images are sharp (Fig. 4c), some inter-frame motions are large due to the shaking movements of the robot.

Tracking by direct alignment with PGMoT and PoT have been compared on this sequence using 1 frame every $N$ frame(s). Various $N$ values were used in order to simulate different robot speeds (e.g., $N = 20$ simulates an average displacement speed of about $45km/h$). Figure 4a shows all the estimated trajectories and the manually registered Lidar SLAM one, in order to evaluate the tracking. Due to the motion between consecutive images PoT struggles to precisely track the robot displacements and thus provides noisy trajectories. PoT is actually strongly influenced by the value of $N$. Indeed, the larger $N$, the more the drift. For $N = 20$, PoT diverges due to too important inter-frame motion for its limited convergence domain. By contrast, PGMoT remains consistent and succeeds to track the robot displacement even for higher $N$ values. The poses estimated with PGMoT are more reliable, precise and thus produce smooth and accurate trajectories qualitatively on par with the Lidar SLAM.

Figure 4d shows the superimposition of the last real and virtual images of the sequence, rendered at the optimal pose computed with PoT for $N = 20$. The poor alignment is particularly visible in the areas highlighted in red whereas PGMoT leads to a much more precise alignment (Fig. 4e).

We also compare these results with ORB-based visual SLAM [19] (OpenVSLAM), extended to the unified camera model (UCM, Sec. II-B). In short, while a sparse map and an erratic trajectory can be estimated from the sequence of acquired omnidirectional images (Fig. 4c), they are up to a scale factor, furthermore variable as there is no loop closure in the sequence of images. No way was found to automatically fix the scale despite the several investigations made. First, the sparse map is too sparse to be registered[2] with the dense point cloud of the 3D model. Second, a sparse map could be obtained from rendered images of the 3D model but not any acquired image could be localized in that map, even when built from images rendered at optimal poses obtained with PGMoT. So there is no way to compare quantitatively OpenVSLAM to PGMoT that reliably succeeds in the full scale estimation of camera, hence robot, poses.

### B. Experiment aboard a vehicle

The vehicle is a Peugeot 308 car on the roof of which a Ricoh Theta S 360 camera is attached thanks to a suction cup. While driving, the dual-fisheye images acquired by the latter camera are streamed to a computer for soft synchronisation of their recording with the GPS data of the car.

Two sequences S1 and S2 of raw dual-fisheye images are acquired at different times on the same path in city streets. The images of the first sequence are transformed to equirectangular spherical images set as input of the photogrammetry software Agisoft Metashape, together with their GPS coordinates as initial guesses, to build offline a dense 3D point cloud. Then, the images of both sequences are transformed to omnidirectional images following the UCM[3] that are later aligned to the 3D point cloud by PGMoT (and PoT for comparison). One may note we could basically take one of the two fisheye views of the Ricoh Theta image instead of generating another one. However, the orientation of the 360 camera on the car roof (imposed for practical reasons) prevents getting a panoramic view of the car surroundings within a single fisheye view.

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

[2] with geometric algorithms of https://www.cloudcompare.org

[3] We share our implementation of image transformation on repository https://github.com/PerceptionRobotique/equi2omni

Gradual difficulty is achieved by applying the tracking on S1, then on S2. Tracking on S1 looks easy as omnidirectional images share the same viewpoint as the equirectangular images used to build the dense 3D point cloud, but the field of view and the non linearity of the resolution is different in both type of images, representing a first challenge. Tracking on S2 is more challenging as, in addition to the difficulties mentioned for the tracking on S1, the car trajectory (both in space and velocity) is not the same and the scene may have changed between the two acquisitions because some pedestrians, cars, trucks or clouds moved.

Figure 5 shows the results of PGMoT (still with *Rule2*) and PoT, both with $N = 1$ as the velocity of the car ($22.5km/h$, on average) is much higher than the one of the mobile robot and the image acquisition framerate is 5 images per second less. Hence, the motion between successively acquired images is sometimes already huge (see Fig. 6). PoT fails as soon as the inter-frame motion increases a little (see, for example, Fig. 6a and 6b) and tracks successfully only 6m (1.6%) of the 366m trajectory of S2 reported on Figure 5b (streets "des Augustins", "Adéodat Lefèvre" and "place Saint Michel" of Amiens, France). Instead, PGMoT succeeds tracking on 318m (86.9%) of the S2 trajectory, even when very large inter-frame motion occurs (see Fig. 6c and 6d). The reason of the failure of PGMoT at the end of the S2 trajectory is the lack of buildings on the right side of the car that lead to slight residual misalignment in successive images that is never overcome.

On S1, both PGMoT and PoT track the 3D point cloud for longer trajectories (Fig. 5a) than on S2: PGMoT successfully tracks on 377m (92.1%, so +5.2% compared to S2) and PoT on 159m (43.4%, so +41.8% compared to S2, though still far below PGMoT capabilities). In addition to the least difficulty on tracking S1 due to the fact that the 3D point cloud is made from images of the same viewpoints, S1 does not show very large inter-frame motion as S2 features (Fig. 6d). But the tracking results on S1 are shown for two reasons: (i) the coherency of the tracked poses with respect to both the streets imprint and the path of camera poses computed by the photogrammetry; (ii) the moderate consistency of the GPS measurements in S1 (on Fig. 5a, the GPS trajectory is sometimes on the wrong side of the street or on sidewalks or even in buildings at the turn) and its inconsistency in S2 where on the West of the turn, the GPS trajectory is several meters in buildings to the right of the street (Fig. 5b) while the actual car obviously not. Hence, such poor consistency of GPS measurements makes irrelevant quantitative evaluations.

Figure 7 shows the superimposition of the last captured and the optimal virtual images for the PoT, for the two cases: convergence (Fig. 7a) and divergence (Fig. 7b), and for the PGMoT for the two cases also: convergence (Fig. 7c) and divergence (Fig. 7d).

Finally, we applied OpenVSLAM in the same way as described in Section IV-A. This time, the localisation was possible (up to scale) for 12.2% of S1, hence 74.7% below PGMoT that outputs the full scale 3D pose of the camera, hence of the vehicle.




(a) S1




(b) S2

Fig. 5: Direct 3D model-based tracking with PGMoT and PoT with the camera aboard a vehicle: Estimated trajectories for (a) the first sequence S1 (GPS: blue; Camera poses output of the photogrammetry process: dark green; PGMoT: orange; PoT: yellow) and (b) the second sequence S2 (GPS: red; PGMoT: pink; PoT: yellow). Trajectories are plotted after conversion to latitude and longitude coordinates exported as kml files to Google My Maps (base map monochrome).

## V. Conclusion

This paper expresses omnidirectional direct visual servoing, representing images as Photometric Gaussian Mixtures. Evaluation in virtual scenes of photographic appearance shows a significant increase of the convergence domain compared to the previous state-of-the-art photometric omnidirectional direct visual servoing. The new rule of Gaussian extent initialization and optimization also shows a significant improvement over the state-of-the-art rules.

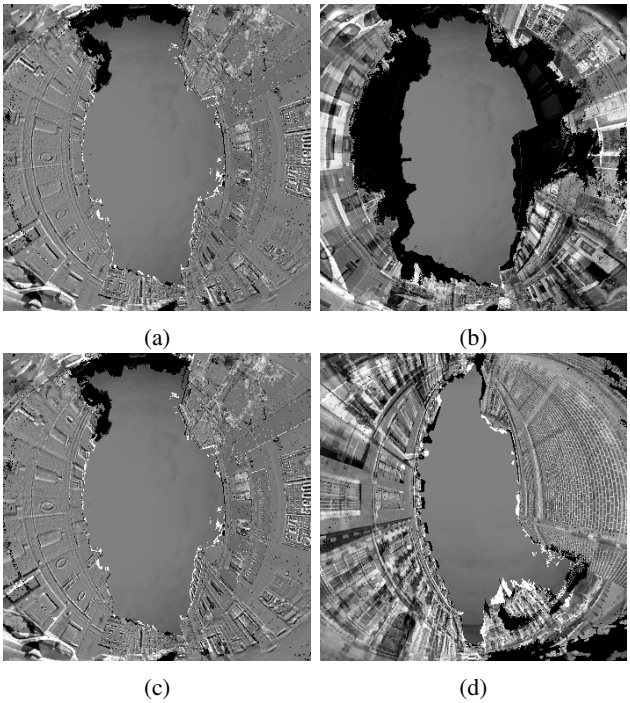
Several experiments of direct 3D model-based tracking of



(a) image 2732 of S2 (b) image 2733 of S2

(c) image 3216 of S2 (d) image 3127 of S2

Fig. 6: Examples of successive images of the S2 sequence with (a-b) medium and (c-d) large inter-frame motion. (a-b) are acquired at the beginning of the trajectory (where PoT fails in Fig. 5b) and (c-d) within the turn.



(a) (b)

(c) (d)

Fig. 7: Superimposition of the last acquired and optimal virtual images ($N = 1$): (a) and (c) correspond to the start of the pink and yellow trajectories of the Fig. 5b respectively, when (b) and (d) correspond to the end of the these trajectories.

two 3D models of a city (obtained by very different means) in omnidirectional images acquired within streets of the same city show the new tracking succeeds where large inter-frame motion prevents the success of the former state-of-the-art one.

Future works will be dedicated on considering the PGM-based visual servoing with an omnidirectional camera attached to a robotic arm, motivated by the large convergence domain that the PGM representation of the omnidirectional image offers. We will also investigate on learning-based methods to predict the initial value of the Gaussian extent.

## REFERENCES


[1] P.-C. Wu, H.-Y. Tseng, M.-H. Yang, and S.-Y. Chien, "Direct pose estimation for planar objects," *Computer Vision and Image Understanding*, vol. 172, pp. 50 – 66, 2018.

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

[3] A. Lakshmi, F. AGJ, and D. Deodhare, "Robust direct visual odometry estimation for a monocular camera under rotations," *IEEE Robotics and Autom. Letters*, vol. 3, no. 1, pp. 367–372, 2018.

[4] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *European C. on Computer Vision*, 2014.

[5] S. Park, T. Schöps, and M. Pollefeys, "Illumination change robustness in direct visual slam," in *IEEE Int. C. on Robotics and Autom.*, 2017, pp. 4523–4530.

[6] J.-Y. Bouguet, "Pyramidal implementation of the Lucas Kanade feature tracker," *Intel, MRL*, 2000.

[7] Y. Ahmine, G. Caron, E. Mouaddib, and F. Chouireb, "Adaptive Lucas-Kanade tracking," *Image and Vision Computing*, vol. 88, Aug. 2019.

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

[9] M. Bakthavatchalam, O. Tahri, and F. Chaumette, "A direct dense visual servoing approach using photometric moments," *IEEE Trans. on Robotics*, vol. 34, no. 5, pp. 1226–1239, 2018.

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

[11] Z. Zhang, H. Rebecq, C. Forster, and D. Scaramuzza, "Benefit of large field-of-view cameras for visual odometry," in *IEEE Int. C. on Robotics and Autom.*, 2016, pp. 801–808.

[12] K. Chappellet, G. Caron, F. Kanehiro, K. Sakurada, and A. Kheddar, "Benchmarking Cameras for OpenVSLAM Indoors," in *Int. C. on Pattern Recognition*, Milan, Italy, Jan. 2021.

[13] E. Marchand and F. Chaumette, "Virtual visual servoing: A framework for real-time augmented reality," in *EUROGRAPHICS C.*, vol. 21(3), Saarebrün, Germany, 2002, pp. 289–298.

[14] G. Caron, E. Marchand, and E. Mouaddib, "Photometric visual servoing for omnidirectional cameras," *Autonomous Robots*, vol. 35, no. 2-3, pp. 177–193, Oct. 2013.

[15] N. Crombez, G. Caron, and E. Mouaddib, "Using dense point clouds as environment model for visual localization of mobile robot," in *IEEE Int. C. on Ubiquit. Robots and Ambient Intell.*, 2015, pp. 40–45.

[16] U. Nadeem, M. Jalwana, M. Bennamoun, R. Togneri, and F. Sohel, "Direct Image to Point Cloud Descriptors Matching for 6-DOF Camera Localization in Dense 3D Point Clouds," in *Int. C. on Neural Information Processing*, 2019, pp. 222–234.

[17] M. Eder, M. Shvets, J. Lim, and J.-M. Frahm, "Tangent images for mitigating spherical distortion," in *IEEE/CVF C. on Computer Vision and Pattern Recognition*, June 2020.

[18] D. Caruso, J. Engel, and D. Cremers, "Large-scale direct slam for omnidirectional cameras," in *IEEE/RSJ Int. C. on Intell. Robots & Syst.*, 2015, pp. 141–148.

[19] S. Sumikura, M. Shibuya, and K. Sakurada, "OpenVSLAM: A Versatile Visual SLAM Framework," in *ACM Int. C. on Multimedia*, 2019, pp. 2292–2295.

[20] J. Barreto, F. Martin, and R. Horaud, "Visual servoing/tracking using central catadioptric images," in *Experimental Robotics VIII*, 2003, pp. 245–254.

[21] G. Caron and D. Eynard, "Multiple camera types simultaneous stereo calibration," in *IEEE Int. C. on Robotics and Autom.*, 2011, pp. 2933–2938.