2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Semantic Equirectangular Visual Tracking in Lightweight 3D Building Reconstructions

Hussein Loubani*, Nathan Crombez, Jocelyn Buisson, and Yassine Ruichek, *Senior Member, IEEE*

***Abstract*— Accurate visual localization often relies on dense, high-fidelity 3D models, which provide rich geometric and photometric detail but are expensive to acquire, heavy to store, and limited in scalability. As an alternative, lightweight city models represent only coarse building volumes, offering compactness, accessibility, and privacy but posing challenges for reliable alignment due to the lack of textures and fine structure. This work addresses these challenges by introducing a semantic equirectangular Gaussian Mixture–based virtual visual servoing approach that aligns real panoramic images with synthetic views rendered from lightweight building models. The method combines semantic building masks with Gaussian Mixtures, a seamless 360° formulation, and frequency-domain computation to overcome the poor gradients of direct photometric binary-mask alignment while maintaining computational efficiency. Experiments on outdoor trajectories show stable tracking under frame skipping and dynamic occlusions through semantic masking. These results indicate that reliable localization is feasible with coarse city models, providing a scalable alternative to high-fidelity reconstructions and opening perspectives for deeper integration of semantic rules into the localization process.**

I. INTRODUCTION

A. Motivations

Visual tracking and localization are central to robotics, computer vision, augmented reality, and autonomous navigation, enabling camera pose estimation through image-to-3D model alignment [1], [2]. High-fidelity 3D models support accurate localization, but are costly to acquire, storage-intensive, and constrained by privacy concerns [3]. A practical alternative is lightweight 3D building models reconstructed from publicly available data. These models preserve essential structures such as footprints and volumes while discarding fine-grained textural or photometric details [4]. Their compactness and scalability make them attractive priors, and similar 3D building or city models have been used for urban localization [5], [6]. However, the simplifications that enable scalability also introduce challenges for reliable alignment. The absence of textures and structural detail impedes visual tracking, particularly at ground level, where viewpoint variations, occlusions, and dynamic objects are prevalent [7]. While most previous research has concentrated on aerial localization using 3D models, ground-level scenarios remain comparatively underexplored.

Recent advances in model-based alignment have shown that Photometric Gaussian Mixtures (PGM) within a Virtual Visual Servoing (VVS) framework can effectively align synthetic views with omnidirectional images derived from dense, colored point clouds [8]. By encoding image appearance into smooth representations, this approach enlarged the convergence domain and improved robustness to large motions. However, its reliance on dense point clouds and photometric cues restricts scalability, while direct binary mask alignment suffers from poor gradients and unstable convergence. Gaussian Mixtures (GM), by contrast, offer smoother gradients that enable more stable optimization with lightweight models.

A complementary study analyzed field-of-view effects in PGM-based VS by comparing perspective and 360° cameras and introduced a trajectory metric contrasting ideal and achieved paths [9]. Equirectangular imagery enlarged the convergence domain, produced straighter/shorter trajectories, and improved robustness to both translational and rotational displacements. However, experiments were limited to controlled robot-arm setups and did not address alignment against coarse urban models, leaving outdoor scalability open. These results motivate exploiting the full 360° FoV for ground-level localization, where occlusions and limited overlap are common, since equirectangular images capture the entire scene in a single frame while preserving continuity across image borders.

Building on these insights, we extend PGM-based VVS beyond dense photometric point clouds to scalable 3D city models by introducing semantic segmentation and encoding building masks as GM. This design choice ensures that only the static architectural structures, which are consistently present, contribute to the alignment. Dynamic or transient objects such as vehicles, pedestrians, or vegetation are not reliably represented in the synthetic environment and can change over time, making them unsuitable for robust alignment. By focusing solely on building geometry within the equirectangular domain, the method avoids misleading cues from such variable elements and enhances the stability and reliability of the alignment process. Combined with the comprehensive coverage of equirectangular imagery, our approach mitigates occlusions, maximizes structural visibility, and supports camera pose tracking in realistic outdoor conditions while relying only on lightweight, publicly available building models.

B. Contributions

This work offers the following key contributions:

- A semantic-based alignment pipeline for visual tracking

*Corresponding author
Hussein Loubani, Nathan Crombez, Jocelyn Buisson, and Yassine Ruichek are with the Université de Technologie de Belfort-Montbéliard (UTBM), CIAD (UR 7533), F-90010 Belfort, France.
E-mail addresses: hussein.loubani@utbm.fr; nathan.crombez@utbm.fr; jocelyn.buisson@utbm.fr; yassine.ruichek@utbm.fr

over coarse 3D building models, leveraging equirectangular acquisitions and semantic information.
- An extension of PGM modeling to semantic masks, addressing the poor gradients of direct binary-mask alignment.
- A frequency-domain calculation of GM to drastically reduce computation times.
- Preprocessing of equirectangular images to take into account the 360° aspect of these acquisitions when calculating GM.

All these contributions are quantitatively and qualitatively evaluated on real-world data, showing consistent visual tracking with coarse 3D building models.

C. Outline

After introducing the motivation, challenges, and objectives in Section I. Section II reviews existing works on pose estimation, visual tracking, localization, and image-to-model alignment. Section III presents our approach, including camera modeling, real semantic mask extraction, synthetic rendering, and Semantic GM-based VVS alignment. Section IV outlines the experimental setup and results. Finally, Section V concludes and discusses future directions.

II. RELATED WORK

Research on image-to-3D alignment has progressed along directions addressing scalability, and complexity. This review highlights contributions and limitations for pose estimation.

A. Skyline-based methods

Early work used skyline cues. Horizon-line matching between equirectangular images and urban models is efficient, reducing localization to horizon alignment [10]. However, it becomes ambiguous when vertical structure is weak or the horizon is cluttered. Building on this idea, omni-skyline matching was later extended for GPS-denied urban canyons, improving localization where traditional satellite-based positioning fails [11]. This study demonstrated improved robustness against partial occlusions, but performance degraded significantly in cluttered or vegetation-rich environments where skyline continuity is disrupted. To expand the representational power of skylines, skyline variability was combined with semantic segmentation to estimate distances to vegetation [12]. This highlighted that skylines encode not only buildings but can also capture additional scene elements. Nevertheless, skyline-only constraints remain fragile under occlusions, high-rises, and limited visibility.

B. Photometric and semantic alignment

To overcome the inherent limitations of skyline geometry, researchers have explored photometric and semantic strategies for image-to-model alignment. Mutual information was proposed to mitigate sensitivity to lighting, but required strong initialization and scaled poorly to city environments due to computational cost [13]. Multi-sensor fusion approaches, such as aligning vehicle-mounted LiDAR with 3D building models, improved accuracy but depended on dense LiDAR, limiting scalability [5]. MeshLoc extended alignment to triangular mesh representations, offering richer geometry and feature-based matching, yet robustness in textureless or large-scale scenes remained problematic [1]. Semantic segmentation can emphasize buildings and vegetation, but robust continuous tracking in outdoor environments remains challenging, leaving a gap to integrate semantics into full alignment pipelines.

C. Localization with low-complex models

A parallel research line has focused on robustness to incomplete, noisy, or low-complex 3D data. Open-source reconstructions can provide scalable priors, but often misalign due to geometric errors and inconsistency [14]. To mitigate these shortcomings, a render-and-compare refinement strategy exploiting pre-trained features was proposed to improve robustness against textureless or inaccurate geometry [15]. This approach demonstrated that feature learning can mitigate some of the weaknesses of coarse reconstructions; however, it still lacked strong geometric consistency across wide baselines and viewpoint changes. Collectively, these works highlight the promise of lightweight priors for large-scale deployment, while also emphasizing the need for new formulations that explicitly account for structural simplifications when aligning images with low-complex building models.

D. LoD-based city models

Structured 3D building models defined at different Levels of Detail (LoD) have become an important reference for urban-scale localization. LoD-2 models enable scalable city-level alignment [6], though robustness degrades under severe occlusions or viewpoint changes. To exploit coarser data, LoD-Loc aligned aerial imagery with LoD models via neural wireframe predictions [16], later extended in LoD-Loc v2 with silhouette alignment for improved robustness [2], but both remained limited to aerial perspectives. Complementary approaches such as BEV-Locator [17], which fused multi-view images into semantic bird's-eye-view maps, and City-Loc [18], which modeled poses as Gaussian distributions conditioned on visual and textual inputs, further highlight the growing role of semantics and multi-modality. However, these methods typically rely on dense or high-LoD city models, restricting scalability in real-world deployments where only lightweight or incomplete data is available.

III. SEMANTIC EQUIRECTANGULAR GM–BASED VVS ALIGNMENT

The proposed scheme estimates the pose of a real 360° camera by aligning synthetic equirectangular images, rendered from a low-complex 3D building representation, with real equirectangular images. As shown in Fig. 1, real images are segmented into building masks, while a virtual camera generates corresponding synthetic masks and depth maps. Both masks are converted into GM, and alignment is achieved by minimizing their discrepancy. This represents a single-frame alignment that, when applied sequentially, extends naturally to visual tracking.

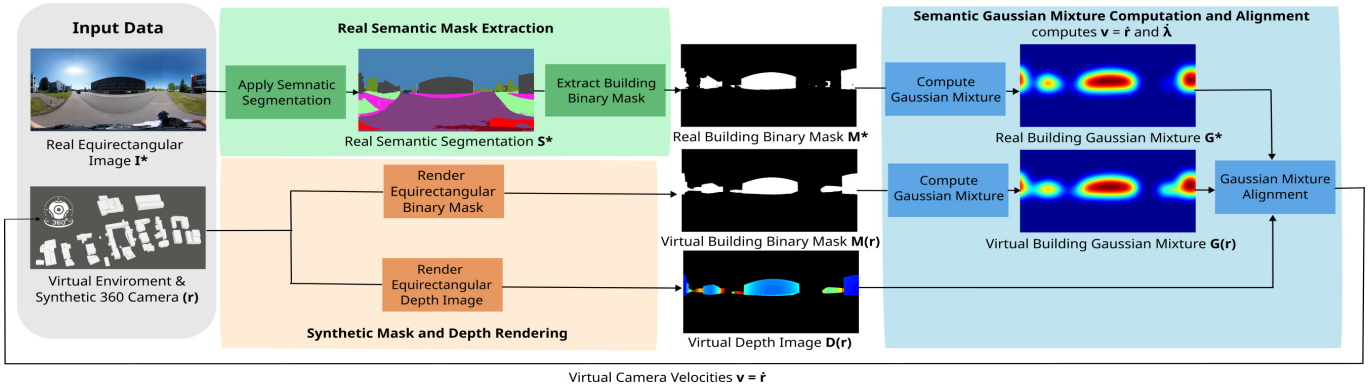


Fig. 1: Proposed scheme for single frame iterative synthetic-to-real alignment: (Green) semantic mask extraction from real images, (Orange) synthetic mask and depth rendering, and (Blue) GM computation and alignment which outputs virtual camera velocities $v = \dot{r}$ that are integrated over iterations to update the pose until convergence.

A. Real and synthetic 360° cameras modeling

Our images are acquired and rendered respectively with a real and a synthetic 360° field-of-view camera. Both cameras follow the same projective geometry, considering a 3D point $\mathbf{X} = (X, Y, Z)^T$ expressed within the camera frame (whether real or synthetic), it is first projected at $\boldsymbol{\theta} = (\theta, \phi)^T$ following the equirectangular projection:

$$\begin{cases} \theta & = arctan(\frac{X}{Z}) \\ \phi & = arctan(\frac{Y}{\sqrt{X^2+Y^2}}) \end{cases} \tag{1}$$

where $\theta \in [-\pi, \pi]$ and $\phi \in [-\frac{\pi}{2}, \frac{\pi}{2}]$ are respectively azimuth and elevation angles. Those angles are then expressed in pixels coordinates $\mathbf{u} = (u, v)^T$ following:

$$\begin{cases} u & = \frac{W}{2\pi}\theta + \frac{W}{2} \\ v & = \frac{H}{\pi}\phi + \frac{H}{2} \end{cases} \tag{2}$$

where $W \in \mathbb{N}$ and $H \in \mathbb{N}$ are respectively the width and height of the acquired or rendered image.

B. Real and synthetic equirectangular building masks

On the one hand, the real equirectangular input image passes through a semantic segmentation network in order to distinguish buildings from other categories such as sky, vegetation, and poles. The resulting segmentation is then binarized to produce the real building mask $\mathbf{M}^*$ that emphasizes the building structures of the environment, providing a reliable reference for the alignment process.

On the other hand, synthetic equirectangular images are rendered by placing a 360° virtual camera inside the lightweight 3D building environment. The virtual camera pose is defined as $\mathbf{r} = [t_x, t_y, t_z, \theta_{ux}, \theta_{uy}, \theta_{uz}]^T$, where the first three components represent translation and the last three describe rotation in axis–angle form. Using a dedicated shader, the virtual camera directly renders binary building masks at pose $\mathbf{r}$ noted as $\mathbf{M}(\mathbf{r})$ that are structurally comparable to the real masks $\mathbf{M}^*$. In parallel, the same rendering process produces a depth image $\mathbf{D}(\mathbf{r})$, which is used in the alignment process (Section III-D).

C. Equirectangular GM Computation

Formally, a GM is defined at coordinates $\mathbf{u}_g = (u_g, v_g)$ as follows:

$$G(\mathbf{u}_g, \mathbf{M}, \lambda) = \sum_{\mathbf{u} \in \mathbf{M}} M(\mathbf{u}) \;\; E_\lambda(\mathbf{u}_g - \mathbf{u}) \tag{3}$$

where $E_\lambda(\mathbf{u}_g - \mathbf{u}) = exp\left(-\frac{(u_g-u)^2+(v_g-v)^2}{2\lambda^2}\right)$ with $\lambda \in \mathbb{R}_+^*$ is the parameter that controls the spatial extent of the Gaussians.

1) Computational complexity: In previous works [19] [8], GM are computed straightforwardly in the spatial domain, leading, as shown by (3), to a computational complexity of order $\mathcal{O}(P^2)$ where $P = W \times H$ is the size of the mask $\mathbf{M}$. Recently, a truncated Gaussian kernel whose size s is relative to the value of the extension parameter λ has been used in [9], leading to a computational complexity of order $\mathcal{O}(P \cdot s^2)$. However, as our experimental evaluation shows (Section IV-A.3), even the latter remains time-consuming, with calculation time increasing dramatically as the size of the mixture or the extension parameter increases. Thus, we propose to calculate the GM in the frequency domain. Indeed, (3) can be written for the computation of the whole mixture as:

$$\mathbf{G} = \mathbf{M} * \mathbf{K}_{E_\lambda} \tag{4}$$

where $*$ denotes the convolution operation and where $\mathbf{K}_{E_\lambda}$ is an unnormalized Gaussian kernel with a standard deviation equal to λ. It is well-known that a convolution operation in the spatial domain is equivalent to a simple product in the frequency domain, which gives us:

$$F(\mathbf{G}) = F(\mathbf{M} * \mathbf{K}_{E_\lambda}) = F(\mathbf{M}) \cdot F(\mathbf{K}_{E_\lambda}) \tag{5}$$

where $F(\mathbf{.})$ denotes the Fourier Transform. The GM is obtained by applying the inverse Fourier Transform on $F(\mathbf{G})$. For each frame, $F(\mathbf{M})$ is computed once and then an element-wise product is applied with $F(\mathbf{K}_{E_\lambda})$, followed by a single inverse transform. The same mechanism applies to the derivatives used later in the control law by replacing $\mathbf{K}_{E_\lambda}$ with $\partial\mathbf{K}_{E_\lambda}/\partial u$, $\partial\mathbf{K}_{E_\lambda}/\partial v$, and $\partial\mathbf{K}_{E_\lambda}/\partial\lambda$, reusing $F(\mathbf{M})$.

As a result, the runtime is dominated by FFTs and scales as $\mathcal{O}(P\ log\ P)$ with weak dependence on λ, which matches the timings reported in Table I.

2) Seamless GM: An equirectangular image maps a full 360° view into a two-dimensional space; this representation has horizontal periodicity and vertical polar singularities. Thus, the image's left and right borders connect seamlessly, and the top and bottom borders collapse into the north and south poles of the sphere, respectively. These continuity properties must be explicitly considered when computing GM; otherwise, discontinuities at the image boundaries lead to border artifacts and fragmented responses.

To address the periodicity and polar singularities of equirectangular images, we introduce a mirror-tiling preprocessing step before GM computation. The image is first divided into upper and lower halves, which are vertically flipped and concatenated above and below the original frame, producing an extended representation of size $2H \times W$. A full-field isotropic Gaussian kernel, together with its spatial derivatives, is then defined over this enlarged domain. The kernel is padded to match the image size and applied through FFT-based circular convolution, which ensures that the Gaussian weighting operates consistently over the entire tiled image, with its effective scale determined solely by λ. After convolution, the central $H \times W$ region is extracted, yielding outputs that are spatially aligned with the original equirectangular input while avoiding polar artifacts and preserving horizontal wrap-around continuity.

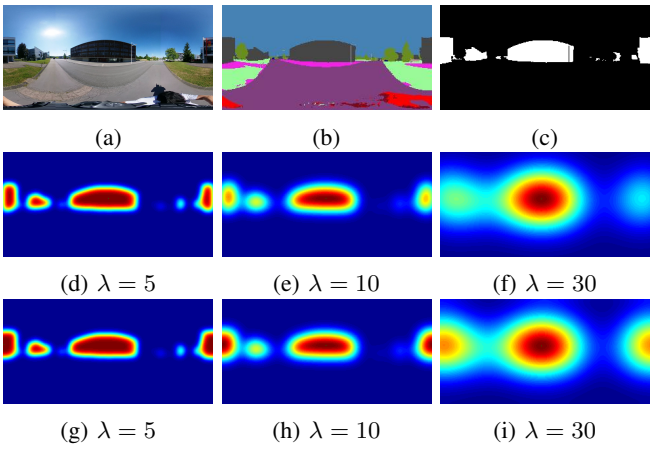
Fig. 2 highlights the differences between GM computed for three different extension parameters ($\lambda = [5.0, 10.0, 30.0]$), without (Fig. 2d-2f) and taking into account the continuity aspect of equirectangular acquisition (Fig. 2g-2i). It can be seen that GMs computed in a bounded image space exhibit border effects. Furthermore, since periodicity is not considered, a building with one part projected onto one border of the image and another part onto another border is spatially disconnected and therefore produces two distinct responses in the mixture instead of a single consistent one. These issues can cause ambiguity or even loss of information, which complicates visual alignment and, by extension, tracking (Section IV). Throughout the paper, a GM computed on the real mask $\mathbf{M}^*$ with an extent λ^* is denoted $\mathbf{G}(\mathbf{M}^*, \lambda^*)$ or simply $\mathbf{G}^*$ for compactness. Similarly, a GM computed on a synthetic mask rendered at pose $\mathbf{r}$ with an extension parameter λ is denoted $\mathbf{G}(\mathbf{M}(\mathbf{r}), \lambda)$ or simply $\mathbf{G}(\mathbf{r})$ for compactness.

D. Equirectangular GM alignment

The estimation of the real 360° camera pose is formulated as a visual alignment problem resolved within a VVS framework. The alignment error that has to be minimized is defined as the difference between a real and a synthetic GM:

$$\begin{aligned}\mathbf{e} &= \bar{\mathbf{G}}(\mathbf{M}(\mathbf{r}), \lambda) - \bar{\mathbf{G}}(\mathbf{M}^*, \lambda^*) \\ &= \bar{\mathbf{G}}(\mathbf{r}) - \bar{\mathbf{G}}^*\end{aligned} \tag{6}$$

where $\bar{\cdot}$ denotes the column vectorization. A classical Gauss-Newton scheme is used to design a GM-based VVS control law that computes both the synthetic 360° camera velocities $\mathbf{v} = \dot{\mathbf{r}}$ and the Gaussian extension increment $\dot{\lambda}$ to regulate to zero the error $\mathbf{e}$:



(a) (b) (c)

(d) $\lambda = 5$ (e) $\lambda = 10$ (f) $\lambda = 30$

(g) $\lambda = 5$ (h) $\lambda = 10$ (i) $\lambda = 30$

Fig. 2: GM computed from a real image (a), its semantic segmentation (b), and binary building mask (c), Results for $\lambda \in \{5, 10, 30\}$ without continuity handling (d–f) and with seamless equirectangular continuity (g–i).

$$[\mathbf{v}\ \dot{\lambda}] = -\mu[\mathbf{L_G}\ \mathbf{J}_\lambda]^+\ \mathbf{e} \tag{7}$$

$$\mathbf{v}_\lambda = -\mu{\mathbf{L_G}_\lambda}^+\ \mathbf{e} \tag{8}$$

where $\cdot^+$ is the Moore–Penrose pseudo-inverse and μ is a fixed control gain. $\mathbf{L_G}$ is the interaction matrix that expresses changes in the synthetic $\mathbf{G}(\mathbf{r})$ with respect to the synthetic 360° camera displacement $\mathbf{v}$. For one sample of the $\mathbf{G}(\mathbf{r})$ evaluated at $\mathbf{u}_g$, the interaction matrix is expressed as a product of Jacobian matrices using the derivative chain rule:

$$\begin{aligned}\mathbf{L}_{G_{cs}}(\mathbf{u}_g) &= \frac{\delta G}{\delta \mathbf{u}_g}\frac{\delta \mathbf{u}_g}{\delta \mathbf{X}}\frac{\delta \mathbf{X}}{\delta \mathbf{r}} \\ &= \frac{\delta G}{\delta \mathbf{u}_g}\frac{\delta \mathbf{u}_g}{\delta \mathbf{X}}\mathbf{L}_\mathbf{X}.\end{aligned} \tag{9}$$

The first Jacobian $\frac{\delta G}{\delta \mathbf{u}_g}$ expresses the analytical Gaussian spatial derivatives:

$$\frac{\delta G}{\delta \mathbf{u}_g} = \sum_{\mathbf{u}\in\mathbf{M}} M(\mathbf{u}) \begin{bmatrix} \frac{-(u_g-u)^2}{\lambda^2} \\ \frac{-(v_g-v)^2}{\lambda^2} \end{bmatrix} E(\mathbf{u}_g - \mathbf{u}). \tag{10}$$

Note that those derivatives can be computed in the frequency domain using the convolution theorem, as described before for the computation of the GM itself (Section III-C.1).
The second Jacobian $\frac{\delta \mathbf{u}_g}{\delta \mathbf{X}}$ expresses the partial derivatives directly related to the camera model, thus in this work, the partial derivatives of the equirectangular projection (Section III-A):

$$\begin{aligned}\frac{\delta \mathbf{u}_g}{\delta \mathbf{X}} &= \frac{\delta \mathbf{u}_g}{\delta \boldsymbol{\theta}}\frac{\delta \boldsymbol{\theta}}{\delta \mathbf{X}} \\ &= \begin{bmatrix} \frac{W}{2\pi} & 0 \\ 0 & \frac{H}{\pi} \end{bmatrix} \begin{bmatrix} \frac{Z}{X^2+Z^2} & 0 & -\frac{X}{X^2+Z^2} \\ -\frac{XY}{\sqrt{X^2+Z^2}\rho^2} & \frac{\sqrt{X^2+Z^2}}{\rho^2} & -\frac{YZ}{\sqrt{X^2+Z^2}\rho^2} \end{bmatrix}\end{aligned} \tag{11}$$

with $\rho^2 = X^2+Y^2+Z^2$. It can be seen that the computation of this Jacobian requires the 3D coordinates of the points

captured by the camera. We use the rendered depth map $\mathbf{D}$ for that purpose, i.e., the 3D coordinates of a point projected at $\mathbf{u}$, are retrieved following the inverse projection:

$$\begin{cases} X & = cos(\phi)sin(\theta)D(\mathbf{u}) \\ Y & = sin(\phi)D(\mathbf{u}) \\ Z & = cos(\phi)cos(\theta)D(\mathbf{u}) \end{cases} \tag{12}$$

Finally, the complete interaction matrix $\mathbf{L}_{\mathbf{G}\lambda}$ contains the Jacobian $\mathbf{J}_\lambda$ that expresses the derivatives of $\mathbf{G}(\mathbf{r})$ with respect to its extension parameter λ. For one sample of the GM evaluated at $\mathbf{u}_g$:

$$\mathbf{J}_\lambda = \frac{dG}{d\lambda} = \sum_{\mathbf{u}\in\mathbf{M}} M(\mathbf{u}) \begin{bmatrix} \frac{-(u_g-u)^2}{\lambda^3} \\ \frac{-(v_g-v)^2}{\lambda^3} \end{bmatrix} E(\mathbf{u}_g - \mathbf{u}). \tag{13}$$

As with the previous derivatives, this term can be efficiently computed in the frequency domain using the convolution theorem, as described before. Unlike the projection-related Jacobian, this term is independent of the projective model of the camera, and specifically captures the sensitivity of the GM to variations of the extent parameter λ. By including J_λ in the extended interaction matrix $L_{G\lambda}$, the control law can jointly update the camera motion $\mathbf{v}$ and the extent λ, ensuring that the optimization not only reduces the alignment error but also regulates the smoothness of the cost function.

In our implementation, we further adopt the extension parameter strategy proposed and evaluated by [8]. This strategy adapts the Gaussian extent λ during the alignment process, starting from large values of λ that smooth low differences and enlarge the convergence domain. λ progressively decreases toward a smaller target value under a low-pass update rule. This continuous coarse-to-fine strategy improves robustness to large inter-frame motions compared to a fixed or manually scheduled λ.

IV. EXPERIMENTAL EVALUATION

A. Experimental Setup

1) Data Acquisition: Real equirectangular images were collected using a RICOH Theta Z1 [20] 360° camera mounted on a tripod and carried by hand while walking along a landmark-guided trajectory to ensure consistent coverage. The acquisition covered $415\,m$ and was traversed at an average walking speed of $7\,km/h$ ($\approx 1.94\,m/s$), capturing diverse building structures, pathways, and vegetation for representative coverage. Fig. 3 shows the acquisition setup, where Fig. 3a depicts the aerial view of the predefined trajectory, and Fig. (3b–3g) illustrate sample equirectangular frames captured at the corresponding locations. To perform quantitative evaluation, pseudo-ground-truth trajectories were estimated with StellaVSLAM, an established open-source VSLAM system with native support for 360° cameras and robust place recognition, closely related to the OpenVSLAM framework [21]. Loop-closure and global optimization are used to mitigate drift. The resulting trajectory was registered to the reconstructed 3D environment through a coarse manual alignment followed by Iterative Closest Point (ICP) refinement, expressing poses in the environment coordinate frame. These poses were then used to compute translation and rotation errors for the proposed GM-based VVS alignment.

2) 3D Building Reconstruction and Simulation: The 3D building models of our environment were reconstructed using City3D [22], a modeling framework that produces lightweight building models. The reconstruction process combines 3D aerial LiDAR point-cloud data and 2D building footprints obtained from the IGN GeoServices catalogue [23], producing coarse geometric building structures suitable for simulation. Fig. 4 illustrates this process, where 4a shows the aerial LiDAR point-cloud data, and 4b shows the reconstructed 3D building models. These latter served as the basis for rendering synthetic equirectangular views using a simulated virtual 360° camera in Unity Simulator [24].

3) Implementation details: The proposed approach is implemented in ROS with C++ and Python nodes. The VVS controller, developed in C++ using ViSP [25], manages alignment, pose initialization, multi-frame triggering, and trajectory logging. Real equirectangular images are published through a dedicated node to ensure synchronized evaluation. Overall, the integration of C++/ViSP for control and GPU-accelerated GM computation enables GM-based VVS on 360° data. Semantic masks are extracted with OneFormer [26], and resized to 320×160, with mean inference time $1.01\,\mathrm{s}$ per frame on an NVIDIA GeForce RTX 3090 GPU, and a Python service computes GM and its derivatives on the GPU using PyTorch with FFT-based convolutions, providing a major speed-up over the original spatial-domain implementation from libPeR[1]. Computation times for different GM resolutions and extension parameters are reported in Table I. The frequency-domain formulation is computationally stable and scales efficiently, unlike the spatial-domain baseline, whose cost grows rapidly with mixture size and extension.

GM size	Spatial domain [1]			Frequency domain (ours)		
	$\lambda = 3$	$\lambda = 6$	$\lambda = 9$	$\lambda = 3$	$\lambda = 6$	$\lambda = 9$
120×60	234.42	777.85	1519.25	72.39	72.01	72.34
240×120	922.66	3320.14	6900.06	72.57	75.55	72.47
480×240	3782.18	13529.20	29287.80	73.96	76.88	71.02
960×480	15069.32	✖	✖	74.04	72.12	72.62
1920×960	✖	✖	✖	77.24	77.34	77.51

TABLE I: Comparison of GM computation times (in ms) calculated in the spatial domain and in the frequency domain for different extension parameter values and different mixture sizes. ✖ indicates that the method required too much memory space and therefore could not be completed.

B. Results, Analysis and Discussion

The alignment process illustrated in Fig. 5 shows the initial and final states of a VVS alignment on a single frame for demonstration purposes. The real equirectangular image $\mathbf{I}^*$ (Fig. 5a) and its corresponding binary building mask $\mathbf{M}^*$

[1]https://github.com/PerceptionRobotique/libPeR_base

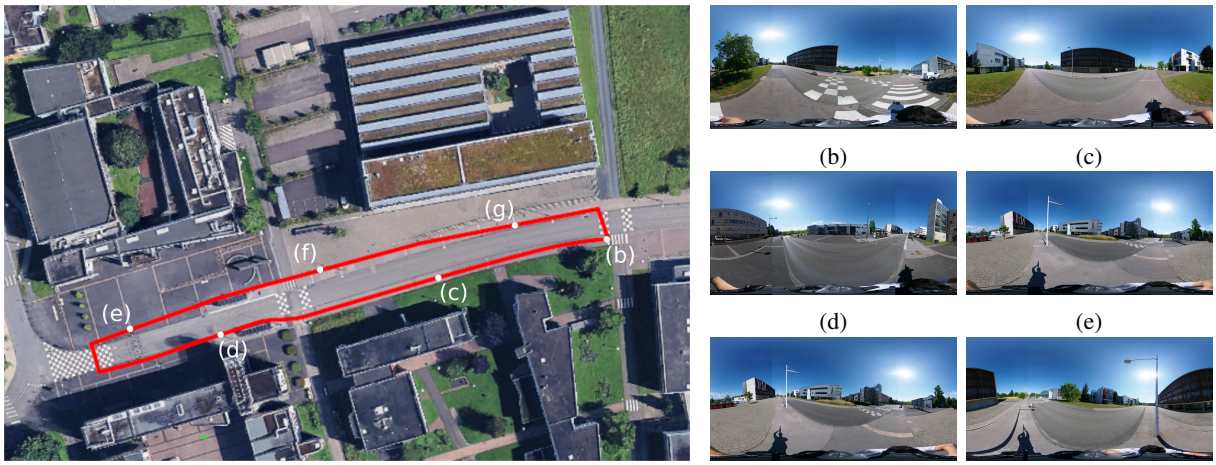


Fig. 3: Acquisition overview: (a) aerial view of the predefined trajectory; (b–g) sample panoramic frames captured along the path.

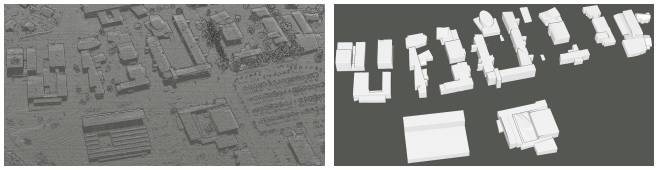


Fig. 4: Study area: (a) aerial LiDAR point cloud, (b) 3D building models reconstructed with City3D [22]

(Fig. 5b), the GM of the real mask $\mathbf{G}^*_{init}$ (Fig. 5c) serves as a target in the beginning and $\mathbf{G}^*_{final}$ at the end (as detailed hereafter). In parallel, the synthetic rendered image $\mathbf{I}(\mathbf{r})$ (Fig. 5e) and its binary mask $\mathbf{M}(\mathbf{r})$ (Fig. 5f) rendered at the initial virtual pose provide the initial synthetic mixture $\mathbf{G}(\mathbf{r})$. The alignment error $\mathbf{E}$ is visualized as images of difference before (Fig. 5g) and after (Fig. 5h) alignment, where non-gray regions highlight discrepancies between the two masks. The final image of difference is not entirely gray because residual differences remain due to coarse 3D geometry, imperfect semantic segmentation, and transient occlusions (e.g., vegetation, poles) present only in the real images. However, even with those variations and even if the initial displacement is large, the virtual camera successfully converges to a correct pose, as the final visual alignment shows. This is explained by the fact that GMs increase the power of attraction of the features, thereby enlarging the convergence domain and allowing convergence even from far initial poses.

Beyond single-frame alignment, we evaluate the capability of the proposed VVS framework to maintain continuous localization across long sequences. In a first experiment, the full sequence of 3,175 frames is processed, corresponding to an average displacement of $13\,cm$ between two consecutive acquisitions. In a second experiment, only one every ten frames is considered, yielding 317 frames with an average displacement of $1.3\,m$ between two consecutive acquisitions. In both cases, tracking is initialized from the first frame, and subsequent poses were estimated iteratively by reusing the converged synthetic pose of the previous frame. For each frame, the control law (Section III-D) is executed over 30 iterations with a two-step coarse-to-fine rule (i.e., 15 for each step). Following the parameter schedule recommended by [8], the first step promotes rapid convergence from large initial errors, while the second step refines alignment around structural details. This strategy ensures stable optimization and accurate convergence.

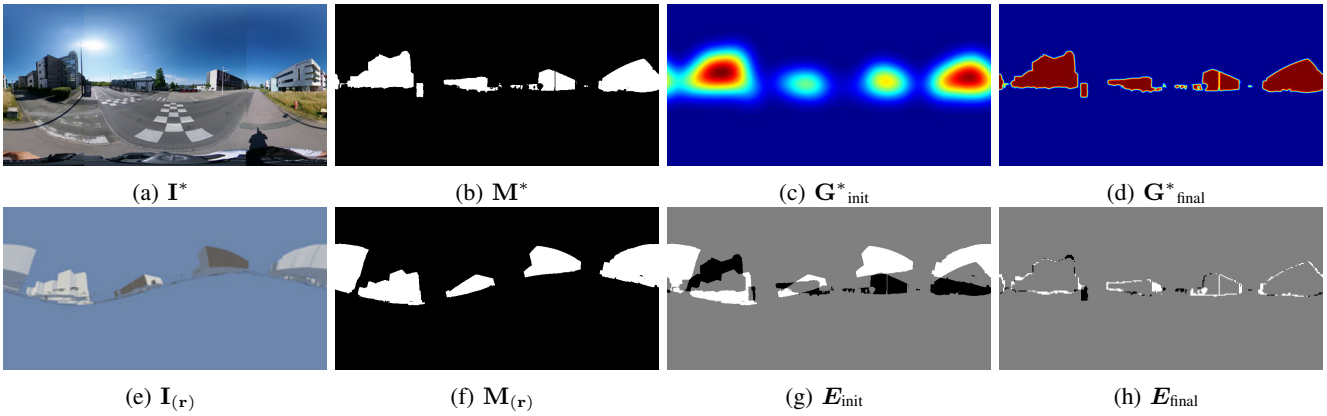
We denote our proposed approach as SEGMVVS when the 360° aspects of equirectangular acquisitions are not considered, and as SEGMVVS (360) when the seamless equirectangular GM computation described in Section III-C.2 is considered.

Quantitative and qualitative trajectory evaluations are conducted using evo[2], a widely adopted tool for benchmarking SLAM and visual odometry. This ensured standardized error computation and reproducibility of our experiments.

Considering the whole sequence, both methods exhibit similar behavior and complete the loop, demonstrating stable tracking. However, closer inspection of the trajectories near the bus stops (Fig. 6a) reveals local deviations on both SEGMVVS and SEGMVVS (360). Semantic mask errors, such as bus stop structures misclassified as buildings, introduce spurious GM gradients and cause trajectory jitter, which is further amplified by LoD model inaccuracies and sensitivity to initialization when overlap is low. Quantitatively, SEGMVVS (360) achieves slightly lower errors than SEGMVVS, as reported in Table II, indicating its improved robustness despite the localized deviations.

When only one frame every ten is considered, the larger inter-frame distance makes the initial visual alignment higher and more challenging. As shown in Fig. 6b, SEGMVVS

[2]https://github.com/MichaelGrupp/evo



(a) $\mathbf{I}^*$ (b) $\mathbf{M}^*$ (c) $\mathbf{G}^*_{\text{init}}$ (d) $\mathbf{G}^*_{\text{final}}$

(e) $\mathbf{I}_{(\mathbf{r})}$ (f) $\mathbf{M}_{(\mathbf{r})}$ (g) $\boldsymbol{E}_{\text{init}}$ (h) $\boldsymbol{E}_{\text{final}}$

Fig. 5: Initial and final alignment states. From the real input image $\mathbf{I}^*$ (a), its binary mask $\mathbf{M}^*$ (b), and GMs $\mathbf{G}^*_{\text{init}}$ (c) and $\mathbf{G}^*_{\text{final}}$ (d), and the synthetic input image $\mathbf{I}_{(\mathbf{r})}$ (e), its binary mask $\mathbf{M}_{(\mathbf{r})}$ (f), and alignment image difference errors $\mathbf{E}_{\text{init}}$ (g) and $\mathbf{E}_{\text{final}}$ (h) are compared before and after $360°$ virtual camera convergence.

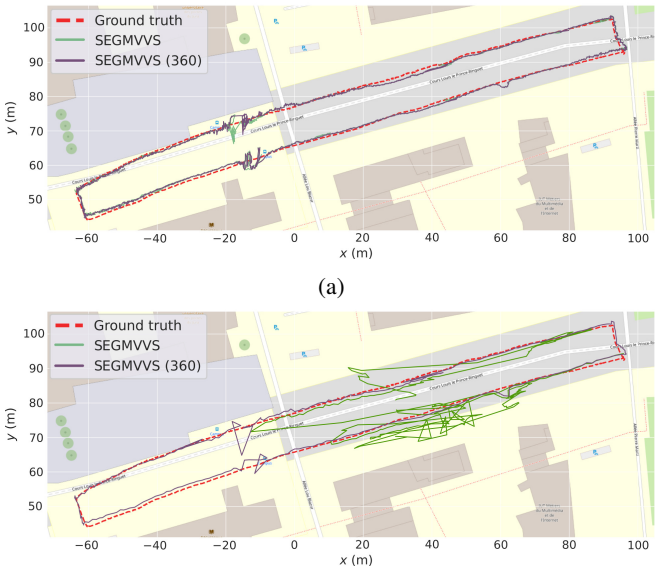
(360) continues to follow the ground-truth trajectory, whereas SEGMVVS diverges severely due to not explicitly considering the $360°$ aspect of equirectangular imagery (Section III-C.2), which introduces boundary artifacts. Quantitatively, this is reflected in Table II, where SEGMVVS shows a mean position error exceeding $40\,m$ with large orientation deviations, while SEGMVVS (360) maintains position and orientation errors close to those obtained in the whole sequence experiment. This confirms that the seamless $360°$ formulation helps maintain consistent alignment even under sparse updates.

Method	Position error (m)		Orientation error (°)	
	Mean	Std Dev	Mean	Std Dev
Whole sequence (3,175 frames)				
SEGMVVS	1.03	0.56	1.86	1.64
SEGMVVS (360)	0.99	0.53	1.85	1.63
One every ten (317 frames)				
SEGMVVS	40.78	43.27	10.10	14.11
SEGMVVS (360)	1.18	1.07	1.95	1.78

TABLE II: Quantitative trajectory evaluation: mean and standard deviation of position and orientation errors for SEGMVVS and SEGMVVS (360) on the whole and one every ten sequences.

Additional challenges arise in the presence of dynamic occlusions. A representative case is shown in Fig. 7, where a bus temporarily blocks the facades (Fig. 7a), leaving only the upper building regions visible (Fig. 7b). To handle this, we propose to integrate more semantic information in the alignment process. Pixels segmented as bus class are used to mask out the corresponding area in the synthetic binary mask $\mathbf{M}(\mathbf{r})$ (Fig. 7c), ensuring that both real and synthetic masks consistently ignore the occluded regions. This preserved the comparability of the resulting GMs, allowing alignment to proceed without corruption from transient dynamics.

Fig. 7d compares the estimated trajectory with and without bus handling against the ground truth. Without masking, the occlusion leads to noticeable deviations in the estimated path. Once the bus leaves the scene, the method immediately locks back onto the exposed building features and realigns with the ground truth. This example demonstrates that semantic cues can be exploited to filter out non-structural elements, providing a simple yet effective mechanism for robust alignment. More broadly, this strategy opens the way to extending the approach with semantic filtering of other non-permanent classes, such as poles, pedestrians, or vegetation, thereby increasing resilience in complex urban environments.



(a)

(b)

Fig. 6: Qualitative trajectory evaluation: (a) whole sequence; (b) one every ten frames, both showing SEGMVVS (green) and SEGMVVS (360) (purple) compared with ground truth (dashed red).

V. Conclusion and Future Work

This paper presented a semantic equirectangular GM–based visual servoing approach for aligning real panoramic images with synthetic views from lightweight

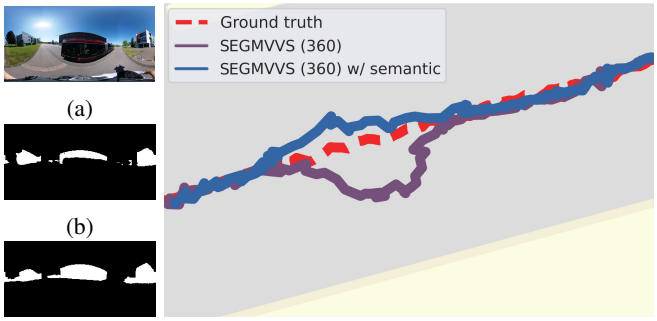


Fig. 7: Trajectory under bus occlusion: (a) real image, (b) real binary mask, (c) Synthetic binary mask, (d) trajectories with SEGMVVS (360) (purple) w/o and with semantic masking (blue) compared to ground truth (dashed red).

3D building models. By combining semantic building masks, Gaussian Mixtures, a seamless 360° formulation, and frequency-domain computation, the method enables efficient alignment. Experiments on long outdoor trajectories demonstrated accurate and stable tracking, resilience to frame skipping, and recovery from dynamic occlusions through semantic masking. These findings confirm that reliable localization is achievable with coarse city models, providing a scalable alternative to high-fidelity reconstructions.

Current limitations arise from segmentation errors, incomplete geometry, and large viewpoint changes. Future work will focus on integrating more sophisticated semantic rules to handle complex scenarios. For example, such rules could be used to resolve ambiguities in segmentation, enforce structural consistency, or predict the expected appearance of occluded or partially visible elements, thereby further strengthening alignment robustness.

REFERENCES

[1] V. Panek, Z. Kukelova, and T. Sattler, "Meshloc: Mesh-based visual localization," in *European Conference on Computer Vision*. Springer, 2022, pp. 589–609.

[2] J. Zhu, S. Peng, L. Wang, H. Tan, Y. Liu, M. Zhang, and S. Yan, "Lod-loc v2: Aerial visual localization over low level-of-detail city models using explicit silhouette alignment," *arXiv preprint arXiv:2507.00659*, 2025.

[3] M. Hatami, Q. Qu, Y. Chen, H. Kholidy, E. Blasch, and E. Ardiles-Cruz, "A survey of the real-time metaverse: Challenges and opportunities," *Future Internet*, vol. 16, no. 10, p. 379, 2024.

[4] F. Rottensteiner, G. Sohn, M. Gerke, J. D. Wegner, U. Breitkopf, and J. Jung, "Results of the isprs benchmark on urban object detection and 3d building reconstruction," *ISPRS journal of photogrammetry and remote sensing*, vol. 93, pp. 256–271, 2014.

[5] A. L. Ballardini, S. Fontana, D. Cattaneo, M. Matteucci, and D. G. Sorrenti, "Vehicle localization using 3d building models and point cloud matching," *Sensors*, vol. 21, no. 16, p. 5356, 2021.

[6] Y. Loeper, M. Gerke, A. Alamouri, A. Kern, M. S. Bajauri, and P. Fanta-Jende, "Visual localization in urban environments employing 3d city models," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 48, pp. 311–318, 2024.

[7] T. Schops, T. Sattler, and M. Pollefeys, "Bad slam: Bundle adjusted direct rgb-d slam," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[8] S.-E. Guerbas, N. Crombez, G. Caron, and E. M. Mouaddib, "Photometric gaussian mixtures for direct virtual visual servoing of omnidirectional camera," in *IEEE CVPR Workshop on 3D Vision and Robotics*, 2021.

[9] S. Schulte, A. N. André, N. Crombez, and G. Caron, "On the impact of the camera field-of-view to direct visual servoing robot trajectories when using the photometric gaussian mixtures as dense feature," in *IEEE/SICE International Symposium on System Integration (SII)*, 2025, pp. 1022–1027.

[10] S. Ramalingam, S. Bouaziz, P. Sturm, and M. Brand, "Geolocalization using skylines from omni-images," in *IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*, 2009, pp. 23–30.

[11] ——, "Skyline2gps: Localization in urban canyons using omni-skylines," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pp. 3816–3823.

[12] L. Martinez-Sanchez, D. Borio, R. d'Andrimont, and M. Van der Velde, "Skyline variations allow estimating distance to trees on landscape photos using semantic segmentation," *Ecological Informatics*, vol. 70, p. 101757, 2022.

[13] G. Caron, A. Dame, and E. Marchand, "Direct model based visual tracking and pose estimation using mutual information," *Image and Vision Computing*, vol. 32, no. 1, pp. 54–63, 2014.

[14] V. Panek, Z. Kukelova, and T. Sattler, "Visual localization using imperfect 3d models from the internet," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 13 175–13 186.

[15] G. Trivigno, C. Masone, B. Caputo, and T. Sattler, "The unreasonable effectiveness of pre-trained features for camera pose refinement," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 12 786–12 798.

[16] J. Zhu, S. Yan, L. Wang, S. Zhang, Y. Liu, and M. Zhang, "Lod-loc: Aerial visual localization using lod 3d map with neural wireframe alignment," *Advances in Neural Information Processing Systems*, vol. 37, pp. 119 063–119 098, 2024.

[17] Z. Zhang, M. Xu, W. Zhou, T. Peng, L. Li, and S. Poslad, "Bev-locator: An end-to-end visual semantic localization network using multi-view images," *Science China Information Sciences*, vol. 68, no. 2, p. 122106, 2025.

[18] Q. Ma, R. Yang, B. Ren, E. Konukoglu, L. Van Gool, and D. Pani Paudel, "Cityloc: 6 dof localization of text descriptions in large-scale scenes with gaussian representation," *arXiv e-prints*, pp. arXiv–2501, 2025.

[19] N. Crombez, E. M. Mouaddib, G. Caron, and F. Chaumette, "Visual servoing with photometric gaussian mixtures as dense features," *IEEE Transactions on Robotics*, vol. 35, no. 1, pp. 49–63, 2018.

[20] RICOH Imaging Company, Ltd., "Ricoh theta z1 360° camera," 2019, available at: https://theta360.com/en/about/theta/z1.html.

[21] S. Sumikura, M. Shibuya, and K. Sakurada, "OpenVSLAM: A Versatile Visual SLAM Framework," in *Proceedings of the 27th ACM International Conference on Multimedia*, ser. MM '19. ACM, 2019, pp. 2292–2295.

[22] J. Huang, J. Stoter, R. Peters, and L. Nan, "City3d: Large-scale building reconstruction from airborne lidar point clouds," *Remote Sensing*, vol. 14, no. 9, p. 2254, 2022.

[23] Institut National de l'Information Géographique et Forestière (IGN). (2025) GeoServices Catalogue. Accessed: Jul. 2025. [Online]. Available: https://geoservices.ign.fr/catalogue

[24] Unity Technologies, "Unity," 2023, game development platform. [Online]. Available: https://unity.com/

[25] É. Marchand, F. Spindler, and F. Chaumette, "Visp for visual servoing: a generic software platform with a wide class of robot control skills," *IEEE Robotics & Automation Magazine*, vol. 12, no. 4, pp. 40–52, 2006.

[26] J. Jain, J. Li, M. T. Chiu, A. Hassani, N. Orlov, and H. Shi, "Oneformer: One transformer to rule universal image segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 2989–2998.