사물인터넷융복합논문지 Vol. 12, No. 5, pp. 53-70, 2026 https://doi.org/10.20465/KIOTS.2026.12.5.053

Adaptive Cropped-SGBM Stereo Ranging with Edge-Deployed Real-Time Perception for Floating-Debris Retrieval by Inland-Waterway USVs

Wan Qi[1], Byung-Won Min[2*]
[1]Ph.D. Student, Department of IT Engineering, Mokwon University
[2]Professor, Department of Game Software Engineering, Mokwon University

적응형 크롭 SGBM 기반 스테레오 거리 측정과 내륙 수로 부유물 회수를 위한 엣지 디바이스 실시간 인지 시스템

만기[1], 민병원[2*]
[1]목원대학교 IT공학과 박사과정 [2]목원대학교 게임소프트웨어공학과 교수

Abstract Reliable three-dimensional localization of floating debris is essential for autonomous retrieval by unmanned surface vehicles (USVs) on inland waterways. However, stereo matching is degraded by specular reflection, ripple noise, and the texture-less surfaces of floating debris, and dense matching is too costly for edge platforms. This paper proposes a semantically guided binocular localization method that confines Semi-Global Block Matching (SGBM) and disparity refinement to a local region defined by a 2D detection box under epipolar constraints, and applies adaptive local weighted-least-squares (AL-WLS) filtering, a γ-mask, and median disparity extraction. On a 400-frame real-waterway sequence, the method reduces the mean absolute percentage error (MAPE) from 13.85% to 2.73% and the SGBM latency by 98.07%, with a mean deviation of 14.84 mm from full-frame matching. Deployed with TensorRT on an NVIDIA Jetson Orin Nano 8GB, and after correcting a training–inference geometric inconsistency (detection rate from 80.1% to 96.7%) and optimizing local-region rectification (rectification latency from 27.32 to 0.84 ms), the system runs at 12.67 FPS with a root mean square error (RMSE) of 128.91 mm at a mean power of 6.84 W, and its perception-to-control-state chain was implemented as a distributed ROS2 pipeline. These results show that guided local stereo matching enables real-time, low-power 3D ranging on embedded hardware within the tested conditions (a single sequence, ground-truth range about 3.3–3.7 m).

Key Words : Unmanned Surface Vehicle, Binocular Stereo Vision, SGBM, Edge Computing, Jetson Orin Nano, Real-Time Ranging, Floating Debris

요 약 내수로에서 무인수상정(USV)을 이용한 부유 쓰레기의 자율 회수를 위해서는 정확하고 신뢰성 있는 3차원 위치 추정이 필수적이다. 그러나 수면의 정반사와 잔물결 잡음, 부유 쓰레기의 낮은 텍스처 특성은 스테레오 매칭의 성능을 저하시키며, 조밀한 매칭 기법은 임베디드 플랫폼에서 높은 연산 비용을 요구한다. 본 논문에서는 에피폴라 제약 조건하에서 2차원 객체 검출 바운딩 박스로 정의된 국소 영역에 반전역 블록 매칭(SGBM)과 시차 정제를 적용하고, 적응형 국소 가중 최소제곱(AL-WLS) 필터링, γ-마스크 및 시차 중앙값 추출을 결합한 의미론적 가이드 기반 양안 위치 추정 방법을 제안한다. 실제 내수로에서 수집한 400프레임 시퀀스 실험 결과, 평균 절대 백분율 오차(MAPE)는 13.85%에서 2.73%로 감소하였으며, SGBM 처리 지연 시간은 98.07% 단축되었다. 전체 프레임 매칭 결과와의 평균 편차는 14.84 mm로 나타났다. NVIDIA Jetson Orin Nano 8GB에서 TensorRT 기반으로 구현한 결과, 학습과 추론 간 기하학적 불일치를 수정하여 객체 검출률을 80.1%에서 96.7%로 향상시켰으며, 국소 영역 정류 시간을 27.32 ms에서 0.84 ms로 단축하였다. 최종 시스템은 평균 소비전력 6.84 W에서 12.67 FPS의 처리 속도와 128.91 mm의 제곱평균제곱근오차(RMSE)를 달성하였으며, 인식부터 제어 상태 생성까지의 과정을 분산형 ROS2 파이프라인으로 구현하였다. 실험 결과는 제안 방법이 제한된 실험 조건(단일 시퀀스, 기준 거리 약 3.3~3.7 m)에서 임베디드 하드웨어를 활용한 실시간·저전력 3차원 거리 추정에 효과적임을 보여준다.

주제어 : 무인수상정(USV), 양안 스테레오 비전, SGBM, 엣지 컴퓨팅, Jetson Orin Nano, 실시간 거리 측정, 부유물

*교신저자 : 민병원(minfam@mokwon.ac.kr)
접수일 2026년 08월 14일 수정일 2026년 09월 10일 심사완료일 2026년 10월 17일

www.kci.go.kr

1. Introduction

River and inland-waterway plastic pollution is a major and persistent contributor to global marine debris, with a small number of major rivers accounting for a disproportionate share of the plastic entering the ocean [1], [2]. Unmanned surface vehicles (USVs) have accordingly been proposed as a scalable platform for autonomous water-surface debris retrieval and water-quality monitoring [3]. Reliable retrieval, however, first requires the USV to determine the three-dimensional position of a floating target relative to the vehicle, a task for which low-cost passive binocular stereo vision is an attractive alternative to LiDAR or active depth sensors on a small, power-constrained platform.

Conventional stereo-matching algorithms such as StereoBM and vanilla SGBM achieve good dense disparity estimation in structured, texture-rich terrestrial scenes [4], but three engineering challenges arise when they are applied directly to floating-target ranging on inland water surfaces. First, specular reflection and dynamic ripples introduce substantial high-frequency noise and matching-failure regions into the disparity map, a phenomenon corroborated by systematic benchmarks of stereo matching in inland-waterway scenes [5], [6]. Second, floating targets such as plastic bottles have smooth surfaces and sparse texture, so their disparity estimates remain unreliable even under otherwise ideal, reflection-free conditions [7], a challenge that similarly arises in strongly scattering underwater imaging [8]. Third, low-disparity background water pixels are frequently mixed into the detection box, and computing statistics over the box without distinguishing them introduces a systematic bias into the ranging result [9]. Recent stereo-monocular fusion work confirms that purely geometric stereo matching commonly fails under texture-less, repetitive, and reflective conditions, and that even fusion-based correction with a monocular prior still requires an additional confidence-routing mechanism [10].

A further, largely orthogonal challenge is computational cost. Dense stereo matching over a full frame scales with image area, and even relatively efficient semi-global matching struggles to meet real-time constraints on resource-constrained embedded GPUs and ARM processors [11]. Object-centric and region-of-interest-guided stereo matching have been proposed to reduce background contamination and computational scope in general 3D object detection and embedded stereo vision [12], [13], but combining this idea with an explicit accuracy/efficiency justification specific to the reflective, texture-less water-surface setting, and carrying it through to a fully deployed edge system, has received comparatively little attention. On the deployment side, prior work has shown that YOLO-family detectors can be run on Jetson-class hardware for maritime surveillance and driving-assistance tasks [14]–[16], and that ROS2 provides a workable middleware for distributed USV perception [17]–[19], but edge deployment is not a simple matter of model migration: it requires systematic validation and optimization tailored to the specific hardware platform and task [16].

This paper builds on our previous work on a reflection-robust, physics-prior-guided YOLOv8 detection front end for floating debris on inland waterways [20], and focuses on the two remaining links in the perception chain: converting a 2D detection into an accurate, stable 3D position under adverse water-surface conditions, and making the complete pipeline run in real time on embedded edge hardware. The specific contributions are as follows.

(1) A semantically guided computational-domain reconstruction strategy that confines both SGBM matching and disparity refinement to a local region of interest defined by the 2D detection box, together with an epipolar-constrained asymmetric cropping scheme that preserves the

www.kci.go.kr

accuracy of the cost-aggregation path within the cropped window. On a 400-frame field sequence, this reduces SGBM latency by 98.07%, with a mean deviation of 14.84 mm from full-frame matching (37.9% of frames identical; 4.5% deviating by more than 50 mm).

(2) An ablation confirms that while median disparity extraction effectively eliminates extreme outliers, the synergistic combination of AL-WLS filtering and γ-mask purification provides the decisive accuracy and stability gains (reducing RMSE from 432.42 mm to 120.54 mm and eliminating jumps $>$ 300 mm). In a parametric sweep, γ=0.35 gives the lowest anchor-frame error (RMSE 120.54 mm), balancing peripheral ripple suppression against disparity sample sufficiency.

(3) A complete edge-deployment study on an NVIDIA Jetson Orin Nano 8GB, including the diagnosis and correction of a training-inference geometric-consistency issue that had been silently suppressing detection confidence (raising the detection rate from 80.1% to 96.7%), and a local-region epipolar-rectification optimization that raises the end-to-end frame rate from 9.49 to 12.67 FPS without loss of ranging accuracy.

(4) Deployment-level validation of the system, including resource and power measurement (mean 6.84 W) and a ROS2-based distributed implementation whose measured control-output rate closely matches the monolithic-script throughput, indicating the engineering feasibility of the perception-to-control-state chain on embedded hardware. Closed-loop autonomous retrieval is not evaluated in this work.

The remainder of this paper is organized as follows. Section 2 reviews related work on water-surface stereo vision and edge-deployed USV perception. Section 3 presents the adaptive binocular localization method. Section 4 describes the edge-deployment process, including the diagnosis of the deployment-consistency issue and the efficiency optimization. Section 5 reports experimental results, and Section 6 concludes the paper.

2. Related Work

2.1 Stereo Vision for Water-Surface and Underwater Targets

Classical dense stereo-matching algorithms have been extensively benchmarked on structured, texture-rich terrestrial scenes [4], but their extension to water surfaces is complicated by specular reflection and ripple-induced noise, as confirmed by systematic evaluations on inland-waterway USV datasets [5], [6]. Related work on lightweight stereo matching for inland waterways has likewise sought efficient, task-specific matching strategies for this setting [7], while underwater stereo-matching studies report comparable degradation under strongly scattering, strongly reflective conditions [8]. To avoid contaminating ranging statistics with background pixels, detection-box-guided processing has been used both in autonomous-driving 3D object detection, where Stereo R-CNN restricts stereo reasoning to the region implied by a 2D box [9], and in general object-centric stereo matching, which performs matching only within a detection-guided region of interest [12]. Fusion-based approaches that combine monocular priors with stereo geometry have also been proposed to handle texture-less and reflective regions, though typically at the cost of an additional confidence-routing mechanism [10]. Real-time embedded stereo matching has separately been studied through UAV-borne SGM optimized for ARM/CUDA devices [11] and through system-level co-design of region-of-interest and temporal sparsity for resource-constrained platforms [13]. The disparity-refinement stage in this work builds on the fast global image smoothing (weighted least squares) framework [21], closely related to guided image filtering [22], and left-right consistency

www.kci.go.kr

checking, a classical disparity-confidence cue that has also been embedded into end-to-end stereo-matching networks [23]. Unlike the ROI-guided approaches above, which rely on learned networks [9], [12] or system-level co-design [13], this work confines classical SGBM matching itself to the detection-guided region, with a disparity-range extension that keeps the epipolar search complete (Section 3.4). Likewise, WLS refinement [21] is applied only within this region and followed by γ-shrunk core-region median extraction (Section 3.6), rather than as whole-image post-processing.

2.2 Edge-Deployed Perception for Unmanned Surface Vehicles

Deploying deep-learning perception models onto embedded platforms such as NVIDIA Jetson has become the mainstream route to autonomous perception for USVs and drones. YOLOv8 has been deployed on a Jetson TX2 for maritime target recognition and tracking [14], and combined with SGBM stereo vision for embedded collision warning in autonomous driving [15], an architecture closely related to the detection-plus-stereo route adopted in this paper. Systematic benchmarks of the YOLO family on Jetson Orin Nano/NX platforms further show that the match among model size, quantization precision, and hardware platform directly determines achievable speed and accuracy [16]. At the software-architecture level, ROS2 has been used to build complete USV perception-and-control frameworks on Jetson Orin Nano [17] and bio-inspired SLAM systems for USVs [18], and its real-time constraint boundaries have been systematically assessed through end-to-end latency tracing in autonomous-driving contexts [19]. This body of work confirms that edge deployment requires systematic, platform-specific validation rather than simple model migration, which motivates the deployment study in Section 4.

3. Adaptive Binocular Three-Dimensional Localization

3.1 Problem Formulation

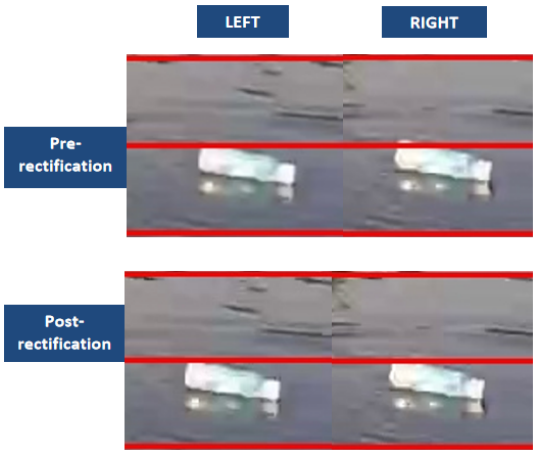
Given the challenges outlined in Section 2.1, this paper proposes a semantically guided, adaptive binocular three-dimensional localization method that seeks a substantial improvement in ranging accuracy and stability through targeted modification of the conventional SGBM pipeline alone, without relying on additional sensors or a deep-learning depth-estimation network. Figure 1–Figure 3 illustrate the successive stages of the method: epipolar rectification, computational-domain reconstruction with epipolar-constrained SGBM cropping, and AL-WLS filtering with γ-mask purification.

3.2 Epipolar Rectification

Under epipolar geometric constraints [24], the projections of a single spatial point onto the left and right images of an uncalibrated binocular system lie on a corresponding pair of epipolar lines that are generally neither horizontal nor collinear, requiring an expensive two-dimensional correspondence search. This work adopts the compact stereo-rectification algorithm of Fusiello et al. [25], applying the camera intrinsic and extrinsic parameters obtained from calibration (Section 5.1) to remap the left and right images so that their epipolar lines become strictly horizontal and row-aligned, reducing the correspondence search to one dimension. Figure 1 compares the left and right images before and after rectification: before rectification, camera-mounting error introduces a noticeable vertical offset for the same target between the two views, which is eliminated after rectification.

After rectification, the effective focal length fx and principal point cx are dynamically re-extracted from the rectified projection matrix rather than taken from the original, uncorrected calibration,

www.kci.go.kr



[Fig. 1] Left and right images before and after epipolar rectification

which removes a static bias from the yaw-angle computation in Section 3.7. Rectification is performed at the native 1920×1080 resolution, after which the images are downscaled to 960×540 for stereo matching and filtering; all pixel quantities in Sections 3.3–3.7 (box size, Ω, $D_{\max}$, and the re-extracted f_x and c_x) refer to this 960×540 scale.

3.3 Semantically Guided Computational-Domain Reconstruction

Given the limited computational capacity of edge platforms, dense stereo matching over the entire image is prohibitively expensive: profiling on a development workstation (Ryzen 9700X + RTX 5060 Ti) shows that full-frame (960×540) SGBM cost aggregation alone averages 551.9 ms per frame, accounting for more than 98% of the three-dimensional localization module's total time (Section 5.4). This paper therefore applies a semantically guided computational-domain reconstruction strategy, in which a local computational domain Ω, guided by the 2D detection box from the front-end detector [20], constrains both the SGBM matching stage itself and the subsequent disparity-refinement and statistics-extraction stages. Letting w_{box} and h_{box}



[Fig. 2] Computational-domain reconstruction from full-frame search to a localized region of interest.

denote the detection box width and height, Ω is defined by a symmetric, adaptive expansion:

$$req_w = \max(w_{box} + 20,\ 200),\ req_h = h_{box} + 20 \quad (1)$$

A 200-pixel lower bound on width supplies the WLS filter with sufficient edge-gradient information and avoids internal limits of OpenCV's disparity-filter interface; the height uses only a 20-pixel buffer. Figure 2 illustrates the resulting compression from full-frame search to a localized ROI, whose pixel area is typically less than 10% of the full frame, giving a theoretical spatial-complexity reduction from O(W·H) to O(w·h) of more than 90% (the measured wall-clock reduction is reported separately in Section 5.4).

3.4 Adaptive SGBM Cropping under Epipolar Constraints

This paper applies the local computational domain Ω to the SGBM matching stage itself, rather than only to WLS filtering and statistics extraction, so that dense matching is carried out directly within Ω. Confining SGBM matching to a cropped region must, however, satisfy the epipolar constraint that disparity is the horizontal

www.kci.go.kr

coordinate difference of the same point between the left and right images: if Ω is cropped and matched independently, the right-image search window will, owing to the missing physical boundary, be unable to cover the maximum disparity $D_{\max}$. To address this, Ω is extended by an additional $D_{\max}$ pixels along the disparity-search direction, forming an asymmetric matching window:

$$x_{search-start} = \max(0,\ x_{\Omega} - D_{\max}) \tag{2a}$$

$$x_{search-end} = \min(W,\ x_{\Omega} + w_{\Omega}) \tag{2b}$$

where x_{Ω} and w_{Ω} are the horizontal coordinate of the left boundary and the width of Ω in the rectified image. W is the image width, and $D_{\max}$ is the maximum SGBM disparity search range ($D_{\max} = 192$ in this work). The right boundary needs no extension because, with disparity defined as x_L-x_R, the matching point in the right image always lies to the left of its left-image counterpart. This asymmetric extension lets the cost-aggregation path accumulate sufficient path-penalty constraints before reaching the target's core region, approximately reproducing, within a bounded window, the cost-aggregation effect of full-frame matching. The cost-aggregation path is nonetheless still truncated at the window boundary, a principled difference from full-frame matching that may introduce a disparity-estimation bias near the window edge; the practical extent of this effect is quantified through the deviation analysis in Section 5.4.

3.5 Adaptive Local Weighted-Least-Squares Filtering

Within the local computational domain, adaptive local weighted-least-squares (AL-WLS) filtering is applied to the left- and right-eye disparity maps d_l and d_r. This filter is a fast approximate implementation of the weighted-least-squares framework of Min et al. [21], whose smoothing strength is adaptively modulated by the color/texture boundaries of the left-eye guidance image, an idea closely related to the guided image filter [22]: it smooths aggressively within homogeneous regions while automatically lowering the smoothing weight near target boundaries to preserve edges. The filter solves the regularized energy functional

$$E(u) = \sum_{p} C(p) \cdot (u_p - d_L(p))^2 + \lambda \sum_{p} \sum_{q \in N(p)} w_{pq}(I_L) \cdot (u_p - u_q)^2 \tag{3}$$

where the confidence weight $C(p) \in [0,255]$ is determined jointly by a left-right consistency check (a classical disparity-confidence cue, also embedded into end-to-end stereo networks [23]) and detection of depth-discontinuity regions; C(p) approaches zero when the disparity difference between $d_l(p)$ and the corresponding position in d_r exceeds roughly 1.5 pixels. $w_{pq}(I_L)$ is an edge-preserving weight based on color difference in the left-eye guidance image:

$$w_{pq}(I_L) = \exp\left(-\frac{|I_L(p) - I_L(q)|^2}{2\sigma_c^2}\right) \tag{4}$$

In this work, $\lambda = 2000$ and $\sigma_c = 1.5$. Solving Eq. (3) yields the refined dense disparity field u(p), converted back to physical units by the fixed-point disparity convention (a stored value of 16 corresponds to one pixel):

$$d_{filtered}(p) = \frac{u(p)}{16} \tag{5}$$

3.6 γ-Mask Purification and Median Extraction

Because Ω has itself been symmetrically expanded, its expanded edge region may introduce water-ripple texture and background noise, making it unreliable to compute statistics

www.kci.go.kr

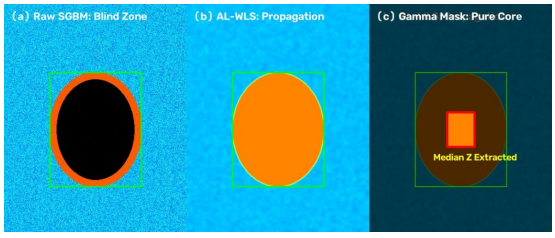
over the whole of Ω. The original detection box is therefore shrunk, relative to its center (u_c, v_c), by a shrinkage coefficient γ applied to the box's original dimensions:

$$w_s = \max(1, \lfloor w_{box} \cdot \gamma \rfloor), \quad h_s = \max(1, \lfloor h_{box} \cdot \gamma \rfloor) \tag{6}$$

Within this shrunken core region, the median (rather than the mean) of the filtered disparity field $d_{filtered}$ is taken, to suppress residual outlier noise:

$$d_{core} = median\{dfiltered(p) : p \in \text{core region}\} \tag{7}$$

Figure 3 illustrates the complete process, from the blind zone in the raw SGBM disparity map, through AL-WLS propagation and repair, to the extraction of a purified core region after γ-mask shrinkage. The parametric sweep that selects γ=0.35, the value with the lowest error on the field ground-truth anchors, is presented in Section 5.3.



[Fig. 3] (a) Raw SGBM blind zone; (b) AL-WLS propagation; (c) γ-mask pure core with median depth extraction.

3.7 Navigation Parameter Computation

Based on the core-region disparity median dcore obtained after filtering and purification, together with the focal length and baseline from binocular calibration, the target's three-dimensional spatial position is computed as:

$$Z = \frac{f_x \cdot B}{d_{core}} \tag{8}$$

$$X = \frac{(u - c_x) \cdot Z}{f_x} \tag{9}$$

$$\theta = \arctan\left(\frac{X}{Z}\right) \tag{10}$$

where f_x and c_x are the horizontal focal length and principal-point coordinate re-extracted after rectification (Section 3.2), B is the binocular baseline, and u is the horizontal pixel coordinate of the target's center. Both X and θ are derived directly from the depth Z, so the core design concern is obtaining a stable and reliable Z, which is precisely the role of the AL-WLS filtering and γ-mask purification mechanism of Section 3.6.

4. Edge Deployment

Validating an algorithm's effectiveness on a desktop-class GPU is not the same as validating its engineering deployability. This section takes the NVIDIA Jetson Orin Nano 8GB as the target edge-computing platform and proceeds through model conversion, diagnosis and resolution of a deployment-consistency issue, and a computational-efficiency optimization, ahead of the comprehensive evaluation reported in Section 5.5–5.6.

4.1 Deployment Environment and TensorRT Conversion

The target platform runs JetPack 6.2 (L4T R36.4.3), CUDA 12.6, and TensorRT 10.3.0.30 in 15 W power mode. The detection model (YOLOv8n_sids_p2) trained in our previous work [20] was exported to ONNX (opset 12, input 640×640) and compiled into a half-precision (FP16) TensorRT engine using trtexec, which uses layer fusion, precision calibration, and hardware-aware operator scheduling to produce an execution engine optimized for the target GPU architecture [26]. Table 1 summarizes the

www.kci.go.kr

conversion. The resulting 22.75 MiB engine is tied to the target hardware's Compute Capability (8.7) and cannot, in principle, be reused directly across hardware platforms; the one-time build cost (≈18.2 min) is not counted in subsequent runtime timing statistics.

〈Table 1〉 TensorRT engine conversion summary

Parameter	Value
Input format	ONNX (opset 12)
Input shape	1×3×640×640
Output shape	1×5×34000
Precision	FP16
Engine build time	1094.7 s (≈18.2 min)
Engine file size	22.75 MiB
Target device	Jetson Orin Nano (Compute Capability 8.7)

4.2 Diagnosis of aDeployment-Consistency Issue

Batch validation over the full 400-frame test sequence, at the same confidence threshold (0.45) used on the PC side, initially yielded a successful detection on only 211 of 392 readable frames (53.8%), far below the PC-side detection performance of the same model. Lowering the threshold to 0.15 raised the detected-frame count to 314 (80.1%), but 2 of the 10 laser-anchor frames used for ranging evaluation still failed to yield any detection, with maximum candidate-box confidence of only 0.063 and 0.113 respectively, indicating that threshold lowering alone could not resolve the problem.

The first candidate cause considered was FP16 quantization suppressing confidence output. Compiling the same ONNX model into matched FP16 and FP32 engines and comparing outputs on identical frames showed near-identical maximum confidence under both precisions (e.g., 0.0032 for a representative low-confidence frame under both), consistent with the normal FP16/FP32 numerical range and with prior evidence that FP16 quantization typically preserves accuracy close to FP32, with pronounced loss more commonly arising at lower precisions such as INT8 [27], [28]. Quantization was therefore ruled out.

Examining the preprocessing pipeline instead revealed that, in the PC-side workflow, detection had always operated on raw, unrectified images, whereas the initial edge-deployment implementation performed detection on images that had already undergone epipolar rectification (in order to let the downstream SGBM+WLS stage reuse them directly). Table 2 compares detection confidence on rectified versus unrectified input for representative low-confidence frames.

〈Table 2〉 Detection confidence: rectified vs. unrectified input (representative frames).

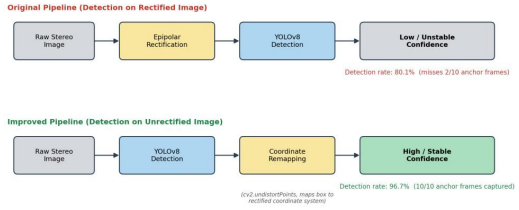
Frame	Confidence (rectified)	Confidence (unrectified)	Change
00164	0.4891	0.6026	+23.2%
00165	0.1613	0.5603	+247.4%
00166	0.0562	0.3763	+569.6%
00167	0.3494	0.5260	+50.5%
00191	0.0188	0.0922	+390.4%
00192	0.0032	0.1126	+3418.8%
00193	0.0355	0.4660	+1213.0%

Switching to the unrectified image produced a substantial confidence improvement across all tested frames. This is consistent with a covariate shift between training and inference distributions: the detection model's training data consisted of unrectified images, and rectification, as a nonlinear geometric transformation, alters the pixel arrangement and local texture, degrading detector performance in a manner consistent with prior analyses of geometric/domain shift in object detection generally [29] and in underwater/water-surface detection specifically [30]. Although rectification remains a necessary preprocessing step for the 3D localization module, it should not be applied as input preprocessing for the 2D detection module.

The deployment pipeline was adjusted accordingly: YOLO detection now operates on the unrectified

www.kci.go.kr

raw image, and the resulting bounding-box coordinates are mapped into the rectified coordinate system via the camera's intrinsic parameters, distortion coefficients, and rectification parameters (cv2.undistortPoints) before being passed to the 3D localization module, which continues to operate on rectified images. Figure 4 compares the pipeline structure before and after the fix, and Table 3 compares overall performance.



[Fig. 4] Detection pipeline structure before and after the fix

〈Table 3〉 Detection performance before and after the fix

Metric	Before (rectified input)	After (unrectified input + coordinate remapping)
Confidence threshold	0.15	0.15
Detected frames (of 392)	314 (80.1%)	379 (96.7%)
False positives (of 400)	1	1
Laser anchor frames captured (of 10)	8	10
RMSE on matched 8 anchors (mm)	129.21	129.58
Mean relative error on matched 8 anchors	3.16%	3.16%
RMSE on full 10-anchor set (mm)	–	128.91

After the fix, the detection rate rose from 80.1% to 96.7%. Both conditions use the same confidence threshold (0.15), confirmed by inspection of the deployment script. Of the 400-frame sequence, 392 frames were readable; detection rate is computed over these 392 frames, whereas the false-positive count is computed over all 400 frames. On a matched 8-anchor comparison, RMSE is essentially unchanged (129.21 → 129.58 mm), confirming that the earlier low-confidence problem originated in classification confidence being affected by geometric consistency, rather than in any degradation of bounding-box localization accuracy. The Before configuration detected only 8 of the 10 anchors, so a full 10-anchor RMSE is not defined for it. The additional coverage of the 2 previously-undetected anchor frames in the full 10-anchor set (RMSE 128.91 mm) further confirms that these recovered detections do not degrade measurement quality. These two anchors correspond to frames with strong specular reflection, which the Before configuration missed entirely; the After configuration captures both, and its 10-anchor RMSE remains close to the 8-anchor value, indicating that the additional detections are reliable. False positives, defined as detections whose measured Z falls outside the physically plausible range [2950, 4150] mm (derived from the observed ground-truth range plus a margin of three times the measurement RMSE), remain at 1 frame under both conditions.

4.3 Computational-Efficiency Optimization: Local-Region Epipolar Rectification

Stage-wise timing after the fix in Section 4.2 (Table 4) identified full-frame epipolar rectification as a previously unrecognized second-largest cost, at 25.9% of the end-to-end total, because the 3D localization module in fact needs only the matching window around the detection box (392 px wide at 960×540 in the representative frame of Fig. 5), while the existing implementation rectified the complete 1920×1080 frame before cropping.

Given single-target detection in this task (at most one floating target per frame), an NMS-bypass simplification was first attempted (selecting the globally highest-confidence candidate directly), but this changed YOLO inference latency only marginally (51.21 → 50.66 ms) and was not adopted. The rectification operation was instead

www.kci.go.kr

〈Table 4〉 Stage-wise timing breakdown before optimization.

Stage	Mean time (ms)	Proportion
YOLO + TensorRT inference	51.21	48.6%
Full-frame rectification (1920×1080 remap ×2)	27.32	25.9%
Letterbox preprocessing	6.00	5.7%
SGBM (cropped)	17.61	16.7%
WLS filtering (local)	1.26	1.2%
Others	~2.00	1.9%
Total	~105.40	100%

changed from rectify-full-frame-then-crop to rectify-only-the-local-region: after bounding-box coordinates are mapped into the rectified coordinate system via cv2.undistortPoints, a cv2.remap operation is applied only to the co rresponding slice of the distortion map. Since cv2.remap is an independent per-pixel lookup operation, this is mathematically equivalent to rectify-then-crop and introduces no additional error, consistent with prior findings that embedded stereo-vision efficiency gains often come from pipeline-specific optimization of the effective computational range rather than generic hardware acceleration [31], [32]. Figure 5 visualizes the resulting computation window for a representative frame.



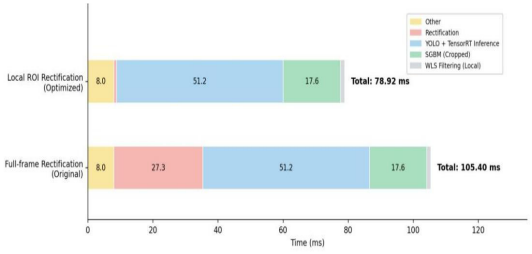
[Fig. 5] Detection box and WLS computation window (frame 00081).

Local-region rectification cuts rectification latency by about 96.9% (27.32 → 0.84 ms) and raises the end-to-end frame rate by about 33.5%, from ~9.49 to 12.67 FPS, while ranging RMSE remains unchanged (128.91 mm) and anchor-frame detection coverage remains 10/10, confirming that this structural optimization preserves both

〈Table 5〉 Timing comparison before and after local-ROI rectification

Stage	Full-frame rectification	Local ROI rectification
Rectification	27.32 ms	0.84 ms
YOLO + TensorRT inference	51.21 ms	51.21 ms
SGBM (cropped)	17.61 ms	17.61 ms
WLS filtering (local)	1.26 ms	1.26 ms
End-to-end total	~105.40 ms	78.92 ms
Real-time frame rate (FPS)	~9.49	12.67

ranging accuracy and detection reliability. Figure 6 compares the stage-wise timings visually.



[Fig. 6] End-to-end timing comparison: full-frame vs. local-ROI rectification

5. Experimental Results

5.1 Experimental Setup

Test data were collected on a real inland-waterway scene exhibiting high-frequency light-ripple disturbance, alternating front/back lighting, and target-motion blur, using a calibrated binocular camera (1920×1080 monocular resolution, 3840×1080 raw stitched output, 60 FPS). A representative 400-frame continuous sequence (00000.bmp–00400.bmp) was selected for continuous ranging-accuracy and stability evaluation, together with 10 key frames whose ground-truth distances were measured with a tape measure and a laser rangefinder (nominal accuracy ±1 mm) while the target was held stationary, each reading being aligned with the video through its spoken announcement on the recording.

www.kci.go.kr

The binocular system was calibrated using Zhang's planar method [33] on a 12×9 checkerboard (20 mm squares, 11×8 detectable inner corners), using more than thirty valid stereo image pairs spanning near/middle/far depths and incidence angles of roughly 0–40°. Table 6 summarizes the resulting calibration and hardware parameters. The reprojection error (RMS) was 0.5243 px (left) and 0.6730 px (right) for single-camera calibration and 0.6139 px for the joint binocular calibration, indicating sub-pixel calibration accuracy. Desktop-side timing measurements (Section 5.4) were obtained on a Ryzen 9700X + RTX 5060 Ti workstation (PyTorch 2.9.0.dev, CUDA 12.8); edge-side measurements (Section 5.5) were obtained on the Jetson Orin Nano 8GB platform described in Section 4.1. Unless stated otherwise, PC-side ranging statistics are computed over the 396 valid frames of the 400-frame sequence, timing over 391 frames after excluding 5 warm-up and 4 no-detection frames, and Jetson-side results over the 392 readable frames (379 detections).

〈Table 6〉 Stereo camera calibration and hardware parameters

Parameter	Value
f_x (left / right)	998.64 px / 999.71 px
f_y (left / right)	999.43 px / 1000.48 px
c_x (left / right)	966.46 px / 988.06 px
c_y (left / right)	532.63 px / 532.12 px
k1, k2 (left)	0.0749, −0.1320
k1, k2 (right)	0.0686, −0.1181
Rotation matrix R	[0.9999 −0.0003 0.0109; 0.0003 1.0000 0.0002; −0.0109 −0.0002 0.9999]
Translation vector T (mm)	$[-118.59, -0.03, 0.33]^T$
Baseline B	118.59 mm
Camera resolution	1920×1080 (mono) / 3840×1080 (stitched) @ 60 FPS

5.2 Ranging Accuracy and Stability

Table 7 compares ranging errors of the baseline (raw SGBM with full-box averaging) against the proposed method (AL-WLS filtering with γ-mask median extraction) at 10 ground-truth key frames, using mean absolute error (MAE), root mean square error (RMSE), and mean absolute percentage error (MAPE):

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|Z_i - Z_{gt,i}| \tag{11}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(Z_i - Z_{gt,i})^2} \tag{12}$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{|Z_i - Z_{gt,i}|}{Z_{gt,i}} \times 100\% \tag{13}$$

〈Table 7〉 Ranging accuracy comparison (10 ground-truth key frames)

Frame	Ground truth (mm)	Baseline (mm)	Ours (mm)	Baseline error	Ours error
00000	3734.9	2416.8	3736.8	35.29%	0.05%
00020	3705.0	2662.1	3531.5	28.15%	4.68%
00041	3674.9	3831.4	3493.1	4.26%	4.95%
00061	3645.0	3799.3	3531.5	4.23%	3.11%
00081	3614.6	3636.6	3505.8	0.60%	3.02%
00100	3585.2	2928.7	3597.4	18.31%	0.34%
00140	3525.0	3144.1	3544.5	10.80%	0.55%
00200	3435.0	3735.6	3584.0	8.75%	4.34%
00300	3360.1	3260.6	3544.5	2.96%	5.49%
00400	3344.8	2505.3	3370.9	25.10%	0.78%

Overall, the baseline achieves MAE = 497.1 mm, RMSE = 652.3 mm, and MAPE = 13.85%; the proposed method reduces these to MAE = 97.1 mm, RMSE = 120.5 mm, and MAPE = 2.73%, an error reduction of approximately 80%. Stereo matching is performed at 960×540 ($f \approx 500$ px after rectification and downscaling, B = 118.59 mm), so one pixel of disparity corresponds to about 190–235 mm over the tested range of 3.34–3.73 m; the RMSE of 120.54 mm thus corresponds to about 0.6 px of disparity. Table 8 further reports sequence-wide temporal stability over 396 valid frames (400-frame sequence minus 4 matching-failure frames).

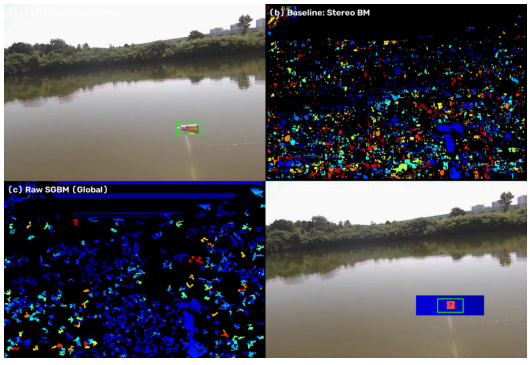
www.kci.go.kr

〈Table 8〉 Sequence-wide ranging stability (396 valid frames)

Metric	Baseline	Ours
Mean (mm)	3249.2	3524.1
Std. dev. (mm)	456.4	105.7
Detrended residual std (linear, mm)	456.3	93.1
Range (mm)	2010.0–4391.7	3313.0–3848.7
Mean frame-to-frame jump (mm)	385.2	45.0
Max frame-to-frame jump (mm)	1389.0	283.7
Frames with jump 〉 300 mm	53.2%	0%

The target approaches the camera during the sequence (ground truth decreases by 390.1 mm over 400 frames, ≈1 mm/frame), so the raw standard deviation includes true motion. After removing a linear trend in frame number (sample standard deviation, n−1), the residual standard deviation drops from 456.3 mm to 93.1 mm (−79.6%). The mean frame-to-frame jump is 385.2 mm for the baseline and 45.0 mm for the proposed method, i.e., noise-dominated rather than motion-dominated; the detrended residual is therefore an upper bound on measurement noise.

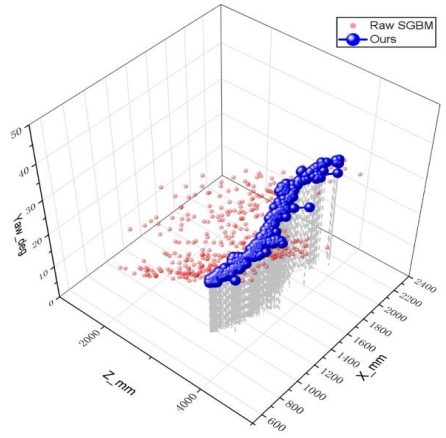
Under the baseline, 53.2% of frame-to-frame transitions exceed a 300 mm jump, with a standard deviation of 456.4 mm; under the proposed method, no jump exceeds 300 mm across all 396 frames and the standard deviation



[Fig. 7] Disparity map comparison: (a) left image with 2D ROI; (b) baseline StereoBM; (c) raw SGBM (global); (d) proposed local WLS in ROI

falls to 105.7 mm (a 77% reduction). Figure 7 compares pseudo-color disparity maps produced by StereoBM, raw SGBM, and the proposed local-WLS method under a rippled water-surface scene, following the stereo-vision evaluation paradigm of Scharstein and Szeliski [4].

Panel (b) shows large-scale matching failure with severe holes and fragmentation under the dynamic water surface; panel (c) removes some ripple-induced mismatches via speckle filtering but retains pronounced jagged edges at the target boundary; panel (d), anchored on the detection box (green) and operating within the local computational domain Ω (blue), achieves smooth, dense, high-fidelity depth continuity, with the orange box marking the γ-mask core region used for median extraction. Figure 8 further compares the resulting three-dimensional spatial trajectories.



[Fig. 8] 3D spatial trajectory comparison: raw SGBM (red) vs. proposed method (blue).

The raw-SGBM trajectory (red) exhibits substantial outlier noise and violent jumps that would readily cause loss of control if fed directly to a downstream controller, whereas the proposed method (blue) produces a continuous, oscillation-free trajectory consistent with the target's true physical motion.

www.kci.go.kr

5.3 Parametric Evaluation and Optimization of the Shrinkage Coefficient γ

The shrinkage coefficient γ directly governs the trade-off between the number of effective pixels retained in the core region and ranging accuracy. Table 9 reports MAE, RMSE, and MAPE at the 10 ground-truth anchors, together with the effective-pixel retention rate Rep, for a sweep of γ values. Rep is the mean ratio (%) of valid disparity pixels (disparity $>$ 0) inside the γ-shrunk core region to the detection-box area, averaged over the 396 frames with a detection; it is slightly below γ^2 because of integer truncation of the shrunk window and invalid disparity pixels.

〈Table 9〉 Effect of γ on ranging accuracy under the deployed configuration (960×540, AL-WLS λ=2000, σ=1.5).

γ	MAE	RMSE	MAPE (%)	Rep (%)
1.0	153.97	200.70	4.56	100.00
0.9	143.60	186.68	4.22	78.58
0.8	135.04	173.80	3.93	62.03
0.7	124.10	159.10	3.59	46.92
0.6	115.87	147.04	3.32	34.33
0.5	105.29	132.98	3.00	24.03
0.45	103.17	129.27	2.92	18.90
0.40	99.72	124.65	2.81	14.93
0.35	97.07	120.54	2.73	11.08
0.30	103.04	123.79	2.89	8.01
0.25	99.90	123.66	2.80	5.50
0.20	112.95	132.10	3.15	3.41
0.15	104.89	127.84	2.94	1.78
0.10	135.66	159.01	3.81	0.66

Table 9 sweeps γ from 1.00 to 0.10 under the deployed configuration. For $\gamma \geq 0.50$, RMSE increases steadily with γ, from 132.98 mm at γ=0.50 to 200.70 mm at γ=1.00, consistent with boundary ripples and background water entering the statistics region. For γ = 0.25–0.45, RMSE stays within 120.54–129.27 mm and MAPE within 2.73–2.92%, forming a plateau. Below γ=0.20 the core region contains too few valid pixels (Rep = 3.41% at γ=0.20 and 0.66% at γ=0.10) and the error rises; at γ=0.10, 189 of 400 frames have fewer than five valid pixels and only 6 of 10 anchors are usable, so this setting is infeasible. We adopt γ=0.35 (Rep = 11.08%), which gives the lowest MAE(97.07 mm), RMSE (120.54 mm) and MAPE (2.73%) while retaining enough valid pixels for a stable median; other values in the plateau perform comparably, and we do not claim that γ=0.35 is a unique optimum. Because γ was selected and evaluated on the same ten anchor frames, the anchor-based accuracy is likely somewhat optimistic; however, any γ in the plateau changes RMSE by less than 9 mm, and even γ=1.00 yields MAPE 4.56%, against 13.85% for the baseline.

〈Table 10〉 Ablation of median extraction, AL-WLS filtering and γ-mask purification (10anchor frames; stability over 396 valid frames)

Configuration	MAE (mm)	RMSE (mm)	MAPE (%)	Std (mm)	Jumps $>$300 mm (%)
Baseline (full-box mean, raw SGBM)	497.10	652.31	13.85	456.40	53.20
Raw SGBM, full-box median	352.67	432.42	9.95	188.34	5.00
Raw SGBM, core-region, median (γ=0.35)	232.40	444.12	6.41	203.60	4.30
AL-WLS, full-box median (γ=1.00)	153.97	200.70	4.56	174.60	6.90
AL-WLS + core-region median (γ=0.35, proposed)	97.07	120.54	2.73	105.7	0.0

As summarized in Table 10, replacing the full-box mean with a median alone reduces RMSE from 652.31 mm to 432.42 mm and suppresses large jumps from 53.20% to 5.00%, proving that median extraction successfully curtails extreme water-surface outliers. However,

www.kci.go.kr

raw SGBM median remains insufficient for precise localization due to pervasive texture-less blind zones (MAPE 9.95%). Introducing AL-WLS filtering over the full box (γ=1.00) substantially improves disparity completeness, lowering RMSE to 200.70 mm and MAPE to 4.56%, yet boundary ripple noise still causes temporal instability (6.90% jumps 〉 300 mm). Conversely, applying the core mask without WLS is unreliable (RMSE 444.12 mm) because the target center in raw SGBM is dominated by disparity holes. Only the synergistic combination of AL-WLS (which propagates valid boundary disparities inward) and γ-mask purification (which discards surrounding water ripples) achieves both high ranging accuracy (RMSE 120.54 mm, MAPE 2.73%) and flawless temporal stability (0.0% jumps 〉 300 mm, std 105.70 mm), outperforming the raw median baseline by 72.1% in RMSE.

5.4 SGBM Cropping: Computational Efficiency and Deviation from Full-Frame Matching

Table 11 reports measured latency for full-frame (960×540) SGBM + full WLS (global scheme) versus cropped SGBM + local WLS (proposed scheme) over 391 valid timing samples (400 frames minus 5 warm-up frames and 4 no-detection frames), on the Ryzen 9700X + RTX 5060 Ti workstation.

〈Table 11〉 SGBM and WLS stage-wise latency: global vs. cropped strategy (391 samples).

Pipeline stage	Global SGBM + full WLS	Cropped SGBM + local WLS (proposed)	Reduction
SGBM matching	551.9 ms	10.66 ms	98.07%
WLS filtering	9.97 ms	4.71 ms	52.79%
3D localization total	561.8 ms	15.4 ms	97.27%

Full-frame SGBM (551.9 ms) accounts for over 98% of the global scheme's total time; confining SGBM to Ω reduces this to 10.66 ms (98.07% reduction), while local WLS filtering falls more modestly (9.97 → 4.71 ms, 52.79%), likely because the cost of computing the guidance image and confidence weights correlates more strongly with ROI size than with full-image size. Combined, the 3D localization module's latency falls from 561.8 ms to 15.4 ms (97.27% reduction); these figures cover only the 3D localization module (SGBM + WLS) and exclude 2D detection inference, whose system-level, end-to-end figures incorporating the Jetson Orin Nano platform are reported in Section 5.5. These latencies were measured on the workstation CPU with 2D detection also running on the CPU in the same process; they support only the relative comparison between the global and cropped schemes, which were measured under identical conditions, and are not directly comparable with the Jetson-side stage timings in Table 13, where detection runs on the GPU.

Because cropping truncates the cost-aggregation path at the window boundary, Table 12 quantifies the resulting deviation between the cropped and global schemes' distance outputs over 396 valid frames.

〈Table 12〉 Ranging deviation: cropped SGBM vs. global SGBM (396 valid frames)

Metric	Value
Valid comparison frames	396 / 400
Mean absolute deviation, MAE (mm)	14.84
Median deviation (mm)	12.79
Deviation std. dev. (mm)	19.71
Max deviation (mm)	166.43
Fully identical frames (diff = 0)	37.9% (150/396)
Frames with deviation 〉 50 mm	4.5% (18/396)
Mean relative error (%)	0.42
Max relative error (%)	4.85

The two schemes show a mean absolute deviation of only 14.84 mm, with 37.9% of frames producing identical output (a consequence of 1/16-pixel fixed-point disparity quantization combined with median filtering). Only 4.5% of

www.kci.go.kr

frames show a deviation exceeding 50 mm, concentrated at moments of rapid target displacement or sudden reflection change. Within the observed deviation level (mean 14.84 mm; 95.5% of frames within 50 mm), the cropped scheme reproduces the ranging output of full-frame matching on the great majority of frames. No formal equivalence margin or hypothesis test is claimed. Together with the 98% efficiency gain, this is the basis for adopting cropped SGBM as the final implementation.

5.5 Edge Deployment: Latency, Accuracy, and Resource Usage

Final system-level validation on the Jetson Orin Nano 8GB (15 W power mode), over the 379 frames with successful detection after the fixes and optimizations of Section 4, gives the stage-wise timing statistics in Table 13.

〈Table 13〉 Final end-to-end timing statistics on Jetson Orin Nano 8GB (n = 379)

Stage	Mean (ms)	Median (ms)	Std (ms)
YOLO + TensorRT inference	51.21	50.96	2.38
SGBM (cropped)	17.61	17.42	0.91
WLS filtering (local)	1.26	1.19	0.20
End-to-end total	78.92	78.80	2.59

The mean end-to-end time is 78.92 ms (12.67 FPS); YOLO inference remains the primary bottleneck (64.9%), followed by SGBM (22.3%) and WLS (1.6%). Detection coverage reached 379/392 readable frames (96.7%), with all 10 laser anchor frames detected. Table 14 compares the system's measured depth against ground truth at these anchors.

〈Table 14〉 Ranging accuracy evaluation against laser ground truth (edge deployment)

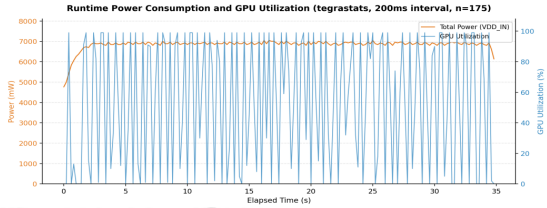
Metric	Value
Number of anchor frames	10
RMSE	128.91 mm
Mean relative error	3.22%

This accuracy is of the same order as the PC-side result (Section 5.2), indicating no marked degradation after TensorRT-quantized edge deployment. Using tegrastats at a 200 ms sampling interval over the complete batch test (175 samples), Table 15 reports resource utilization and power consumption.

〈Table 15〉 Runtime resource utilization and power consumption

Metric	Mean	Max	Min
Memory usage	1924.1 MB	1930.0 MB	1736.0 MB
GPU utilization	52.0%	99.0%	0.0%
Total power (VDD_IN)	6842.7 mW	7046.0 mW	4767.0 mW
CPU+GPU power	1781.7 mW	1907.0 mW	641.0 mW
GPU temperature	46.9°C	47.7°C	45.7°C

Mean memory usage (~1.9 GB) is about 25% of the 8 GB total, and mean total power (~6.84 W) is only about 46% of the 15 W power-mode ceiling, indicating headroom for future functional expansion, consistent with prior benchmarks noting considerable room to optimize the relationship between power draw and compute utilization on Jetson-class hardware [34], [35]. GPU utilization fluctuates between full load during YOLO inference and near-idle during SGBM/WLS (mean 52%, peak 99%), and both GPU and CPU temperatures remain stable within 45–48°C, well below the platform's safe ceiling, indicating no thermal throttling over the batch test (175 samples at 200 ms, about 35 s); longer-duration operation was not tested. Figure 9 shows the real-time power and GPU-utilization curves recorded throughout the test.



[Fig. 9] Runtime power consumption and GPU utilization (tegrastats, n = 175)

www.kci.go.kr

5.6 ROS2 System Integration Validation

The validated perception algorithms were encapsulated as distributed ROS2 (Humble) nodes — a camera-driver node, a 2D-detection node, a 3D-localization node, and a navigation-computation node — reproducing the complete perception-to-control-state chain. The final control-state output topic, /control/state, stabilizes at approximately 12.5 Hz (ranging between 12.48 and 12.53 Hz, as shown in Figure 10), closely matching the 12.67 FPS measured with the monolithic script in Section 5.5, indicating that inter-node message-passing overhead is negligible and throughput remains governed by the core algorithms rather than the communication framework. The isolated 2D-detection node publishes at a noticeably higher rate (~16.3 Hz), reflecting the pipeline-parallelism potential of a distributed publish-subscribe architecture relative to a monolithic serial script. Across the 379 processed frames, the navigation module's lateral-offset and yaw-angle outputs (mean X = 1386.31 mm, std 296.14 mm; mean θ = 21.33°, std 3.87°) are consistent with the target's persistent off-center imaged position throughout the sequence, completing the chain from image acquisition to control-state output on embedded hardware. This verifies the output of the perception-to-control-state chain only; closed-loop autonomous retrieval (actuation and vehicle-motion feedback) was not evaluated.

5.7 Limitations

All ranging results come from a single sequence of one floating target, with ground truth available at ten anchor frames spanning about 3.34–3.73 m; these anchors were used both to select γ and to evaluate accuracy, so no independent test set was available. Strong overexposure, other target types and sizes, longer ranges, and other water surfaces were not covered, and the Jetson-side evaluation reflects a short batch run (about 35 s). The reported

```
average rate: 12.630
        min: 0.063s max: 0.097s std dev: 0.00919s window: 14
average rate: 12.528
        min: 0.063s max: 0.112s std dev: 0.01061s window: 27
average rate: 12.508
        min: 0.038s max: 0.134s std dev: 0.01405s window: 41
average rate: 12.497
        min: 0.038s max: 0.134s std dev: 0.01263s window: 54
average rate: 12.526
        min: 0.038s max: 0.134s std dev: 0.01174s window: 67
average rate: 12.523
        min: 0.038s max: 0.134s std dev: 0.01130s window: 80
average rate: 12.496
        min: 0.038s max: 0.134s std dev: 0.01080s window: 93
average rate: 12.511
        min: 0.038s max: 0.134s std dev: 0.01023s window: 106
average rate: 12.504
        min: 0.038s max: 0.134s std dev: 0.00986s window: 119
average rate: 12.507
        min: 0.038s max: 0.134s std dev: 0.00954s window: 132
average rate: 12.500
        min: 0.038s max: 0.134s std dev: 0.00993s window: 145
average rate: 12.506
        min: 0.038s max: 0.134s std dev: 0.00962s window: 158
average rate: 12.510
        min: 0.038s max: 0.134s std dev: 0.00941s window: 171
average rate: 12.505
        min: 0.038s max: 0.134s std dev: 0.00928s window: 184
average rate: 12.487
        min: 0.038s max: 0.134s std dev: 0.01004s window: 197
average rate: 12.481
        min: 0.038s max: 0.134s std dev: 0.01008s window: 210
```

[Fig. 10] Convergence of the /control/state topic publishing rate

accuracy should therefore be read as valid only within these tested conditions.

6. Conclusion

This paper has proposed a semantically guided, adaptive binocular three-dimensional localization method for floating-debris ranging under low-texture, strongly reflective inland-waterway conditions, and validated its complete deployment on an NVIDIA Jetson Orin Nano 8GB edge platform. Epipolar-constrained SGBM cropping, guided by a 2D detection box, reduces stereo-matching latency by 98.07% with a mean deviation of 14.84 mm from full-frame matching; median-based disparity extraction, together with AL-WLS filtering and γ-mask purification, reduces ranging MAPE from 13.85% to 2.73%; the ablation reveals that while median extraction ensures basic outlier robustness, the combination of AL-WLS filtering and γ-mask purification delivers the primary accuracy and stability enhancements, achieving smooth, oscillation-free trajectory output. On the deployment side, diagnosing a training-inference geometric-consistency issue

www.kci.go.kr

raised the detection rate from 80.1% to 96.7%, and a local-region rectification optimization raised the end-to-end frame rate from 9.49 to 12.67 FPS, without loss of ranging accuracy. The final system achieves 12.67 FPS at 128.91 mm RMSE (3.22% relative error) and 6.84 W mean power, and was validated end-to-end as a ROS2 distributed perception chain. Together with the detection front end reported in our previous work [20], these results advance the perception pipeline from algorithmic feasibility to engineering deployability for real-world inland-waterway USV operation. Future work includes extending validation to a wider range of water conditions and target types, and exploring further quantization (e.g., INT8) and power-mode trade-offs on the Jetson platform.

REFERENCES

[1] L.C.M.Lebreton et al., "River plastic emissions to the world's oceans," Nature Communications, Vol.8, No.1, Article 15611, 2017.

[2] L.J.J.Meijer et al., "More than 1000 rivers account for 80% of global riverine plastic emissions into the ocean," Science Advances, Vol.7, No.18, Article eaaz5803, 2021.

[3] H.C.Chang et al., "Autonomous water quality monitoring and water surface cleaning for unmanned surface vehicle," Sensors, Vol.21, No.4, Article 1102, 2021.

[4] D.Scharstein and R.Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," International Journal of Computer Vision, Vol.47, No.1-3, pp.7-42, 2002.

[5] Y.Zhang et al., "Research on the visual SLAM algorithm for unmanned surface vehicles in nearshore dynamic scenarios," Journal of Marine Science and Engineering, Vol.13, No.4, Article 679, 2025.

[6] Y.Cheng et al., "Are we ready for unmanned surface vehicles in inland waterways? The USVInland multisensor dataset and benchmark," IEEE Robotics and Automation Letters, Vol.6, No.2, pp.3964-3970, 2021.

[7] J.Su et al., "A lightweight target-driven network of stereo matching for inland waterways," arXiv preprint arXiv:2410.07915, 2024.

[8] L.Zhu et al., "Reliable and effective stereo matching for underwater scenes," Remote Sensing, Vol.16, No.23, Article 4570, 2024.

[9] P.Li, X.Chen and S.Shen, "Stereo R-CNN based 3D object detection for autonomous driving," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, 2019, pp.7644-7652.

[10] H.Ryu et al., "CR-FusionNet: hybrid stereo-monocular fusion for depth perception in autonomous and robotic systems," IEEE Access, 2026.

[11] B.Ruf et al., "ReS²tAC: UAV-borne real-time SGM stereo optimized for embedded ARM and CUDA devices," Sensors, Vol.21, No.11, Article 3938, 2021.

[12] A.D.Pon, J.Ku, C.Li and S.L.Waslander, "Object-centric stereo matching for 3D object detection," in Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Paris, 2020, pp.8383-8389.

[13] J.Erhardt, Z.Li, R.Pinkham, A.Berkovich and Z.Zhang, "SteROI-D: system design and mapping for stereo depth inference on regions of interest," in Proceedings of the tinyML Research Symposium, Austin, 2025.

[14] A.Haijoub, A.Hatim, A.Guerrero-Gonzalez et al., "Enhanced YOLOv8 ship detection empower unmanned surface vehicles for advanced maritime surveillance," Journal of Imaging, Vol.10, No.12, Article 303, 2024.

[15] S.E.Tsai and C.H.Hsieh, "A real-time collision warning system for autonomous vehicles based on YOLOv8n and SGBM stereo vision," Electronics, Vol.14, No.21, Article 4275, 2025.

[16] L.Rey, A.M.Bernardos, A.D.Dobrzycki et al., "A performance analysis of you only look once models for deployment on constrained computational edge devices in drone applications," Electronics, Vol.14, No.3, Article 638, 2025.

[17] M.Mamani, M.Fernandez, G.Luna et al., "High-resolution water sampling via a solar-powered autonomous surface vehicle," arXiv preprint arXiv:2512.09798, 2025.

[18] F.Coelho, J.V.T.Borges, P.Padrao et al., "Bioinspired SLAM approach for unmanned surface vehicle," arXiv preprint arXiv:2509.19522, 2025.

[19] A.Bonci, F.Brunella, M.Colletta et al., "ROS 2-based architecture for autonomous driving systems: design and implementation," Sensors, Vol.26, No.2, Article 463, 2026.

[20] W.Qi and B.W.Min, "A reflection-robust and lightweight YOLOv8-based model for water-surface plastic bottle detection," Journal of Internet of Things and Convergence, Vol.11, No.6, pp.33-47, 2025.

[21] D.Min, S.Choi, J.Lu et al., "Fast global image smoothing based on weighted least squares," IEEE Transactions on Image Processing, Vol.23, No.12, pp.5638-5653, 2014.

[22] K.He, J.Sun and X.Tang, "Guided image filtering," IEEE

www.kci.go.kr

Transactions on Pattern Analysis and Machine Intelligence, Vol.35, No.6, pp.1397-1409, 2013.

[23] Z.Jie, P.Wang, Y.Ling et al., "Left-right comparative recurrent model for stereo matching," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, 2018, pp.3838-3846.

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

[25] A.Fusiello, E.Trucco and A.Verri, "A compact algorithm for rectification of stereo pairs," Machine Vision and Applications, Vol.12, No.1, pp.16-22, 2000.

[26] Y.Zhou and K.Yang, "Exploring TensorRT to improve real-time inference for deep learning," in Proceedings of IEEE HPCC/DSS/SmartCity/DependSys, 2022, pp.2011-2018.

[27] P.Serio, H.Wang, Z.Wei et al., "EdgeLPR: on the deep neural network trade-off between precision and performance in LiDAR place recognition," arXiv preprint arXiv:2605.02275, 2026.

[28] M.Nagel, M.van Baalen, T.Blankevoort et al., "Data-free quantization through weight equalization and bias correction," in Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, 2019, pp.1325-1334.

[29] Y.Chen, W.Li, C.Sakaridis et al., "Domain adaptive Faster R-CNN for object detection in the wild," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, 2018, pp.3339-3348.

[30] M.Wille, D.Miller, T.Fischer et al., "Why domain matters: a preliminary study of domain effects in underwater object detection," arXiv preprint arXiv:2604.26174, 2026.

[31] D.Hernandez-Juarez, A.Chacón, A.Espinosa et al., "Embedded real-time stereo estimation via semi-global matching on the GPU," in Proceedings of the International Conference on Computational Science (ICCS), San Diego, 2016, pp.143-153.

[32] J.Zhao, T.Liang, L.Feng et al., "FP-Stereo: hardware-efficient stereo vision for embedded applications," arXiv preprint arXiv:2006.03250, 2020.

[33] Z.Zhang, "A flexible new technique for camera calibration," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.22, No.11, pp.1330-1334, 2000.

[34] D.Minott, S.Siddiqui and R.J.Haddad, "Benchmarking edge AI platforms: performance analysis of NVIDIA Jetson and Raspberry Pi 5 with Coral TPU," in Proceedings of IEEE SoutheastCon, Concord, 2025, pp.1384-1389.

[35] M.Kang and M.Park, "Power estimation and energy efficiency of AI accelerators on embedded systems," Energies, Vol.18, No.14, Article 3840, 2025.

만 기(Wan Qi) [정회원]



- Sep 2000 ~ Jul 2004, Sichuan University, Bachelor of Engineering in Automation.
- Mar 2010 ~ Dec 2012, Guizhou University, Master of Engineering in Control Engineering.
- Dec 2023 ~ Present, Mokwon University (Daejeon, South Korea), Ph.D. Candidate in IT Convergence (Intelligent Systems)

〈관심분야〉

Software Development, Artificial Intelligence & Machine Learning, Robotics & Intelligent Control.

민 병 원(Byung-Won Min) [정회원]



- He received M.S. degree in computer software from Chungang University, Seoul, Korea in 2005.
- He received Ph.D. degree in the dept. of Information and Communication Engineering, Mokwon University, Daejeon, Korea, in 2010.
- He is currently a professor of Mokwon University since 2010.

〈관심분야〉

digital communication systems, Big Data

www.kci.go.kr