Transactions on Comparative Education (2026)
Clausius Scientific Press, Canada

DOI: 10.23977/trance.2026.080108
ISSN 2523-5818 Vol. 8 Num. 1

SLAM Technology Teaching Practice Course Design

Mingzhi Chen[1,a], Chenrui Wu[1,b,*]

[1]*School of Mechanical Engineering, University of Shanghai for Science and Technology, No.516 Jungong Road, Shanghai, China*
[a]*327404319@qq.com*, [b]*wuchenrui@usst.edu.cn*
**Corresponding author*

Keywords: Visual SLAM (Simultaneous Localization and Mapping); Practice-based Teaching; ORB-SLAM3; ROS2 Humble, Gazebo; Process-outcome Evaluation

Abstract: To address common issues in practice-based teaching of "Visual SLAM Technology" for undergraduate robotics engineering students—namely, "students may understand lectures but cannot implement," and "students may run code but cannot modify it"—this paper designs and implements a 16-hour progressive practical course centred on the ORB-SLAM3 and ROS2 Humble/Gazebo toolchain. Following the main line of "knowledge–implementation–evaluation–integration," the course organizes practical tasks into four modules: front-end visual odometry, back-end graph optimization (g2o/BA), loop-closure detection and constraint incorporation, and system integration. Students are required to complete a closed-loop training process—ranging from source installation, node development, and source-code localization to mapping and localization demonstrations—using public datasets and a self-built simulation environment. The assessment adopts an individual-completion policy and a dual-track "process–outcome" evaluation scheme. Through on-site inspections, programming deliverable checks, and individual presentations/viva, the assessment focuses on verifying continuous trajectory output from the front end, error reduction or trajectory improvement brought by g2o optimization, loop-closure candidates with geometric verification evidence, and end-to-end reproducible execution and visualization of ORB-SLAM3 in ROS2 Humble with a self-built Gazebo scene. Teaching practice indicates that this design strengthens students' engineering-level understanding of key Visual SLAM modules, debugging and troubleshooting capabilities, and standardized delivery skills, thereby effectively supporting the associated theoretical course.

1. Introduction

SLAM (Simultaneous Localization and Mapping) is one of the key foundational technologies for autonomous robot navigation [1] and an essential required topic for undergraduate students majoring in Robotics Engineering. Teaching SLAM not only involves theoretical knowledge such as camera models, geometric vision, probabilistic estimation, and optimization, but also emphasizes practical capabilities to implement and validate algorithms in real or near-real environments.

In the curriculum of the Department of Robotics Engineering at the University of Shanghai for Science and Technology, SLAM comprises 32 total hours, evenly split between theory and practice.

The quality of practical-course design directly affects students' mastery of SLAM "from conceptual understanding to engineering implementation". However, in traditional engineering education, SLAM teaching often remains at the level of formula derivation, offline data demonstrations, or instructor-led demos. Students lack reproducible experimental pipelines and systematic software engineering training, which leads to the gap that "they may follow the lecture but cannot implement, and they may run it but cannot modify it".

Since the 1990s, information technology has gradually entered education and developed rapidly [2]. Introducing virtual simulation, open-source frameworks, and engineering toolchains into SLAM practice teaching can form a "learning by doing" model: students implement algorithms, tune parameters, and evaluate performance in a controllable simulation environment, and then transfer these skills to more complex robot system integration tasks.

Wen [3] proposed and implemented a modular innovative practice course on LiDAR SLAM for mechanical students, adopting progressive modules and OBE-based quantitative assessment, which alleviated the disconnect between "technical application" and "innovation capability cultivation". Correspondingly, this course focuses on visual SLAM. It designs tightly coupled programming and operational practice content around key knowledge points of the theoretical course, forming a complete practical pipeline from front-end visual odometry, back-end graph optimization, and loop closure detection to ORB-SLAM3 system setup and ROS2/Gazebo integration.

2. Course Objectives

To facilitate teaching implementation and assessment, the course objectives are designed progressively along "knowledge–implementation–evaluation–integration," and are expressed as verifiable deliverables where possible [4]. The course textbook is *Introduction to Visual SLAM: From Theory to Practice* [5], which includes theoretical content and corresponding sample code. Based on the sample code, the practical course requires students to achieve the following objectives.

Objective 1: Develop implementation capability for the front end of visual SLAM (visual odometry). Given a dataset or simulated camera sequence, students shall be able to: independently implement ORB feature extraction and descriptor computation; complete feature matching and outlier rejection (e.g., geometric-consistency checks based on RANSAC); estimate camera pose using 2D–2D (essential matrix) or 3D–2D (PnP) methods; output trajectories and keyframe results; and perform basic quantitative evaluation (e.g., trajectory error or reprojection error trends).

Objective 2: Develop implementation capability for back-end optimization (graph optimization) in visual SLAM. Students shall be able to: formulate a pose–landmark graph optimization problem (basic BA form), clarify error terms and optimization variables; implement a minimal runnable pose-graph/BA optimization program using g2o (graph construction, vertices and edges, robust kernels, solver configuration, and iteration stopping criteria); compare error changes before and after optimization; and explain common reasons for convergence or failure (initialization, outliers, scale drift, degenerate motion, etc.).

Objective 3: Understand and implement loop-closure detection and the introduction of loop constraints. Students shall be able to: understand the role of loop closure in suppressing drift; implement at least one loop-closure strategy (e.g., ORB bag-of-words candidate retrieval plus geometric verification); generate loop-closure constraints when loops occur; observe changes in trajectory/map consistency before and after loop closure; and provide screenshots or error comparisons.

Objective 4: Acquire capability in reproducing open-source systems and integrating robot systems. Students shall be able to: successfully compile ORB-SLAM3 (including dependency configuration and troubleshooting common build errors); build a Gazebo simulated robot scene in

ROS2 (including sensor models and topic publication), enabling the camera data stream to feed into SLAM; complete an end-to-end demonstration of "simulation scene mapping, localization, and visualization"; and submit reproducible experimental documentation.

3. Practical Course Design

Based on the 16-hour practical component and the course objectives, a progressive design with four modules is adopted: front end, back end, loop closure, and system integration.

3.1. Teaching Mode and Resource Configuration

Teaching mode: Practical sessions are conducted after completing the corresponding theoretical instruction for each module. There are 4 practical classes, each lasting 4 hours, totalling 16 hours.

Experimental environment: Ubuntu 22.04 / ROS2 Humble, OpenCV, Eigen, g2o, Pangolin, Gazebo, ORB-SLAM3.

Data and scenes: In the introductory phase, public datasets (EuRoC and TUM RGB-D) are used. In the integration phase, a self-built indoor Gazebo scene (e.g., corridor or room) is used.

3.2. Practical Modules and Task Arrangement

The practical course includes four modules: front-end fundamentals and pose estimation, back-end optimization (g2o graph optimization/BA), loop-closure detection and loop-closure constraint construction, and ORB-SLAM3 integration with ROS2 Humble/Gazebo (system integration). It proceeds in a progressive manner from basic application to advanced enhancement and then comprehensive application. Details are shown in Table 1.

3.3. Practical Course Assessment

Each student must independently complete all programming tasks and provide a process explanation and viva. The overall score is 100 points, adopting a dual-track "process–outcome" evaluation, covering three competency dimensions: modern tool usage, solution design/implementation, and engineering communication [6]. AI tools are allowed to assist with the released tasks. During assessment, the front-end pose estimation must output continuous trajectories; g2o optimization must demonstrate error reduction or trajectory improvement; loop closure detection must provide loop candidates and geometric verification evidence; ORB-SLAM3 must run and be visualized under ROS2 Humble with a self-built Gazebo scene, completing robot localization and mapping, and a course final report must be written. From the evidence on instructional improvement and teacher professional development, effective pedagogical interventions typically require clear objectives, actionable task designs, and a verifiable chain of learning evidence to support continuous iteration and quality enhancement [7].

4. Final Requirements and Evaluation System for the Visual SLAM Technology Course

4.1. Final Requirements

The final task of this course follows an individual-completion policy. Each student must independently complete all programming and integration/debugging tasks under the specified environment and submit complete deliverables. Division of labour by splicing work together or mutual copying is not allowed. The final task must complete a closed loop of "reproduction installation – node development – source-code understanding – mapping experiments," with the

following requirements:

Table 1: Practical Teaching Content and Plan for Visual SLAM.

Module	Key Content & Activities	Key Learning Objectives	Hours
Basic Application	(1) Pre-class: toolchain and preparation (CMake/package management/data I/O; OpenCV/Eigen basics); (2) Task release: pose estimation on a given sequence for two frames/multi-frames; (3) Front-end practice: ORB feature extraction, matching, RANSAC outlier rejection; (4) Pose estimation & visualization: essential matrix / PnP; output trajectory or pose sequence	(1) Understand and run the basic visual front-end pipeline; (2) independently implement ORB feature extraction and matching and complete pose estimation; (3) perform basic analysis of matching quality and estimation errors	4
Advanced Enhancement	(1) Back-end optimization: build error terms and optimization variables; use g2o to implement BA/pose-graph optimization; (2) Robust kernels and outlier handling: compare convergence and results with/without robust kernels; (3) Loop closure: candidate retrieval (BoW/similarity) and geometric verification to form loop constraints; (4) Thresholding and failure analysis: false positives/false negatives, degenerate motion, sensitivity to initialization	(1) Understand and implement the "vertex–edge–error–solver" paradigm of back-end graph optimization; (2) master the loop-closure pipeline and generate loop constraints; (3) develop tuning and troubleshooting ability (convergence, robustness, threshold selection)	8
Comprehensive Application	(1) ORB-SLAM3 build and reproduction: dependency configuration, common error troubleshooting, camera parameter configuration; (2) ROS2 Humble integration: image topic input, timestamps and TF, RViz2/Pangolin visualization; (3) Self-built Gazebo scene and system integration: scene building and sensor model configuration; complete mapping and localization in a self-built scene; (4) demo and viva: scene design rationale, system architecture, metrics and issue review	(1) Ability to reproduce open-source SLAM systems and integrate them engineering-wise; (2) complete end-to-end SLAM execution and demonstration in a self-built Gazebo scene; (3) produce reproducible documentation and complete demo materials	4

Students must install and run ORB-SLAM3, successfully compile it and run at least one official demo (monocular/stereo/RGB-D); write a final report that truthfully records and analyzes 2–3 installation/compilation issues and solutions (including key error messages and troubleshooting process). Based on ROS2 Humble, students must write an image subscriber node, and correctly feed camera/simulation images into the ORB-SLAM3 interfaces (TrackMonocular / TrackRGBD / TrackStereo).

Final deliverables include: a final report (recommended 15+ pages) organized per course requirements (installation & troubleshooting, node design & integration, principle & source-code location analysis, experimental design & results analysis, summary/reflection, references); code and running instructions (README) including environment/dependency versions, ORB-SLAM3 source acquisition (repo/commit), build steps, run commands or launch/scripts; and (optional bonus) a 3–5 minute demo video (system runtime footage, trajectory/map visualization, brief explanation).

Table 2: Evaluation System for the Visual SLAM Practical Course.

Graduate Attribute	Related Course Objectives	Assessment Component	Achievement Method	Weight(%)
Modern tools & engineering platforms	Objective 4 primarily, plus Objective 1	On-site check; process explanation	On-site spot check: ROS2 Humble project structure and topic connectivity; ORB-SLAM3 compilation and execution; explanation of dependency configuration, parameter meanings, and troubleshooting of common errors (oral and live demo)	40%
Design/develop solutions	Objectives 1, 2, 3	Programming deliverables	Submission and acceptance: (1) front end (ORB extraction/matching, RANSAC and pose estimation); (2) back end (g2o graph construction and optimization, including robust-kernel comparison); (3) loop closure (candidate retrieval and geometric verification; able to generate loop constraints or provide evidence of loop triggering). Code must be runnable, well-structured, and results reproducible.	40%
		Presentation / viva	5–8 minute individual presentation: implementation pathway, key design choices (thresholds/robust kernels/data processing), failure cases and fixes; instructor questions (e.g., outlier handling, convergence conditions, coordinate frames/timestamps)	
Professional ethics & engineering communication	Objective 4	Deliverables	Individual README (environment, dependencies, build/run commands, data/scene description, reproduction steps); code style (modularization, comments, logs); result materials (trajectory/error comparison plots, loop-closure hit records, etc.)	20%

4.2. Final Evaluation Criteria

The total course score is 100 points. The continuous assessment accounts for 50% (40% for practical content and 10% for class performance), and the final report accounts for 50%. The evaluation criteria for practical content are shown in Table 2. Class performance mainly assesses attendance and in-class engagement. The final report is graded by: environment setup and troubleshooting (20 points); ROS2 image node and system integration (30 points); understanding of ORB-SLAM3 principles and source code (30 points); report quality (20 points); and extension and innovation (0–10 points, bonus items such as designing more systematic experiments and deeper analyses across parameters and scenes).

During the SLAM teaching practice, the content of the course practical design helps students more systematically master the basic principles of visual SLAM, enabling them to complete “learning by doing” tasks in a project-based manner, and effectively promoting and supporting the theoretical course.

5. Summary

This paper proposes a practical course design for visual SLAM for undergraduates, highlighting engineering training goals of "reproducible, verifiable, and explainable." The course adopts a four-module progressive structure, incorporating front-end pose estimation, back-end g2o optimization, and loop-closure mechanisms into a unified practical pipeline. Through the system integration task combining ORB-SLAM3 with ROS2/Gazebo, students' algorithm implementation capabilities are extended to robot software system integration, debugging, and deployment skills. In terms of evaluation, the course uses an individual-completion policy and a dual-track process–outcome assessment with on-site checks, deliverable verification, and viva presentations. This approach emphasizes not only final runtime performance but also process evidence such as troubleshooting, parameter choices, and source-code understanding, effectively reducing the learning bias of "only being able to run but lacking understanding." In future work, bonus tasks may further introduce multi-sensor fusion, navigation integration, and more systematic comparative experiment designs to enhance openness, innovation, and comprehensive application depth.

Acknowledgements

This project is supported by the National Natural Science Foundation of China (Grant No. 52371331).

References

[1] Tsubouchi, T. (2019). Introduction to simultaneous localization and mapping. Journal of Robotics and Mechatronics, 31(3), 367-374.
[2] Leidner, D. E., & Jarvenpaa, S. L. (1995). The use of information technology to enhance management school education: A theoretical view. MIS quarterly, 265-291.
[3] Wen, C. (2025, June). Innovative Practice Course Design for Laser SLAM Robotics. In World Education Forum 2(11).
[4] Shaheen, S. (2019). Theoretical perspectives and current challenges of OBE framework. International Journal of Engineering Education, 1(2), 122-129.
[5] Gao, X., & Zhang, T. (2021). Introduction to Visual SLAM from Theory to Practice, Singapore: Springer Singapore.
[6] Duffee, L., & Aikenhead, G. (1992). Curriculum change, student evaluation, and teacher practical knowledge. Science education, 76(5), 493-506.
[7] Sims, S., Fletcher-Wood, H., O'Mara-Eves, A., Cottingham, S., Stansfield, C., Goodrich, J., ... & Anders, J. (2025). Effective teacher professional development: New theory and a meta-analytic test. Review of educational research, 95(2), 213-254.