

Top LinkedIn Posts — Technical Writing

High-performing technical LinkedIn posts on camera calibration, C++, and robotics — inspiration and templates for technical content.

Dr. Farshid Pirahansiah · pirahansiah.com

Over the past year, my standout posts have featured camera calibration, Python, C++, OpenCV, NVIDIA optimizations, advanced C++ techniques, AI integration in robotics with LLMs, computer vision, and machine learning. Explore more: [Link](#)

My Top LinkedIn Posts from the Past Year

- [Camera Calibration](#)
- [Camera Calibration](#)
- [Optimization Methods Computer Vision](#)
- [multi-GPU](#)
- [NVIDIA Jetson](#)
- [On-Device LLM](#)
- [On-Device LLM](#)
- [C++](#)
- [C++](#)
- [C++](#)
- [My experience Robotics 3D SLAM LLM Vision Multimodal](#)

1. Camera Calibration and Subpixel Precision

Camera Calibration Post

In this post, I delved into advanced methods of camera calibration, focusing on geometric analysis and calibration patterns using tools like **MATLAB**, **Python**, and **OpenCV**. I highlighted a **C++** algorithm implemented for high-speed, high-accuracy corner detection within calibration patterns, emphasizing rotation and orientation. The process was further refined using subpixel accuracy and noise reduction techniques, enhancing precision in computer vision applications.

2. Optimization Methods in Computer Vision

Optimization Methods Post

Optimizing **NVIDIA GPUs** for deep learning has been a crucial topic, especially in multi-GPU setups. In this post, I explored how leveraging **CUDA** and **cuDNN** can lead to high-performance AI applications such as video analytics, face recognition, and smart systems. By optimizing GPU resources, we can achieve significant performance gains in complex computer vision tasks.

3. Mastery of C++ for High-Performance Computing

C++ Programming Post

C++ remains a cornerstone in high-performance computing, supporting object-oriented, procedural, and generic programming paradigms. In this post, I explored key features of C++ including:

- **Classes and Objects** for encapsulation.
- **Inheritance** for code reuse.
- **Polymorphism** through function overloading and overriding.
- **Memory Management** with `new` / `delete` and smart pointers.
- Utilization of the **Standard Template Library (STL)** for containers, algorithms, and iterators.
- Modern features like **lambda expressions** and **auto type deduction**.
- Focus on **concurrency**, **templates**, and efficient debugging and performance optimization tools.

These features make C++ an essential language for developing efficient and robust applications in computer vision and AI.

4. Advancements in Robotics: 3D SLAM, LLMs, and Multimodal Vision

Robotics and AI Post

This comprehensive post explored advanced technologies in robotics, focusing on AI inference, computer vision, and generative AI on robotics platforms. Key highlights included:

Applications in Robotics

- Development of **autonomous robotic systems**.
- Enhancements in **industrial automation**.

AI Inference in Robotics

- **Active Tracking** and **Intrusion Detection**.
- **Safety Monitoring** of workers.
- **Visual Inspection** and **Assembly Detection**.
- **Object and Anomaly Detection** in manufacturing.
- **Quality Control** through image analysis.
- **Real-time Equipment Monitoring**.
- **Facial Recognition** for security and attendance.
- **Automated Vehicle Navigation** in warehouses.
- **Gesture Recognition** for hands-free operation.

Computer Vision on Robotics Platforms

- Real-time adaptation of models like **ChatGPT** on robotics devices, including **Raspberry Pi**, **Intel Neural Compute Stick**, and **NVIDIA Jetson**.

- Focus on computer vision, large language models, and generative AI applications in practical robotics settings.

Robotics Hardware and Tools

- **Hardware:**
 - **Raspberry Pi** series (3, 4, and 5 with accelerators required).
 - **Google Coral TPU.**
 - **Intel® Neural Compute Stick 2.**
 - **NVIDIA Jetson Nano** (2GB, 4GB, 8GB RAM).
 - **NVIDIA JETSON AGX XAVIER.**
 - **NVIDIA AGX Orin.**
 - **OpenCV AI Kit.**
- **Tools and Libraries:**
 - **LLAMA C++**
 - **OpenVINO GenAI**
 - **Intel® Distribution of OpenVINO™ Toolkit**

Introduction to Robotics AI

Robotics computing processes data locally on devices like smart robots to reduce latency and enhance reliability. This supports real-time decision-making across various robotics applications, offering increased security and reduced network strain.

Benefits of Robotics AI

- **Cost-effective:** Reduces data transmission costs.
- **Speed:** Essential for quick decision-making in autonomous robotics.
- **Privacy:** Keeps sensitive information secure on-device.

My Top LinkedIn Posts of 2024

As we approach the end of 2024, I wanted to reflect on some of my most impactful LinkedIn posts from the year. From deep dives into camera calibration techniques to insights on optimizing VRAM consumption during LLM training, it's been an exciting journey exploring the forefront of computer vision, robotics, and AI.

Camera Calibration and Subpixel Precision

Camera Calibration Post

In this post, I delved into advanced methods of camera calibration, focusing on geometric analysis and calibration patterns using tools like **MATLAB**, **Python**, and **OpenCV**. I highlighted a **C++** algorithm implemented for high-speed, high-accuracy corner detection within calibration patterns,

emphasizing rotation and orientation. The process was further refined using subpixel accuracy and noise reduction techniques, enhancing precision in computer vision applications.

My Research About Camera Calibration

Geometric Analysis, Calibration Patterns, MATLAB, Python, C++, OpenCV, Subpixel Precision. A **C++** implemented algorithm was used for high-speed, high-accuracy corner detection within calibration patterns, focusing on rotation and orientation. The process was refined by subpixel accuracy and noise reduction techniques.

Camera Calibration

In computer vision methods, image information from cameras can yield geometric information pertaining to three-dimensional objects. The correlation between the topographical point and camera image pixel is necessary for camera calibration. Hence, the camera's parameters, which constitute the geometric model of camera imaging, are utilized to establish the association among the 3D geometric location of one point and its consistent point in an image. Typically, experiments are conducted to obtain the aforementioned parameters and relevant evaluation, which is a process called camera calibration.

Camera Calibration Methods

- **Active Vision Calibration**
- **Calibration with Known Object**

Key Concepts in Camera Calibration

- **Internal and External Parameters:** Calibration involves estimating these parameters to correct for lens distortion, measure object sizes, and position the camera within a scene.
- **Self-Calibration:** This method doesn't rely on specific calibration objects but utilizes the camera's movement through a scene to determine parameters.
- **Accuracy and Evaluation:** The accuracy of calibration depends on the construction tolerances of the calibration pattern and the quality of images used for calibration. Reprojection error is used as a qualitative measure of calibration accuracy.

Challenges and Innovations

- **Image Quality:** The quality of calibration images plays a crucial role. Techniques like the Structural Similarity Index Measure (SSIM) and Peak Signal-to-Noise Ratio (PSNR) are used to assess image quality.
- **Optimization Techniques:** Advanced algorithms and optimization methods enhance calibration accuracy and efficiency.

Applications

- **Stereo Vision:** Fundamental for stereo vision systems, determining the accuracy of 3D scene reconstructions.
- **Robotics and Autonomous Vehicles:** Critical for navigation and object recognition tasks.

References

- Book chapter titled "Camera Calibration and Video Stabilization Framework for Robot Localization" in the book **Control Engineering in Robotics and Industrial Automation**, published by Springer (24/07/2021).
 - **Pattern Image Significance for Camera Calibration**, IEEE Student Conference on Research and Development (SCOREd 2017). [Link](#)
 - **Auto-Calibration for Multi-Modal Robot Vision based on Image Quality Assessment.**
-

Optimization Methods in Computer Vision

Optimization Methods Post

Optimizing **NVIDIA GPUs** for deep learning has been a crucial topic, especially in multi-GPU setups. In this post, I explored how leveraging **CUDA** and **cuDNN** can lead to high-performance AI applications such as video analytics, face recognition, and smart systems.

Deep Learning Optimization Post

Additionally, I discussed monitoring system resources on the **NVIDIA Jetson Nano**:

- **Tegrastats Utility:** For real-time system monitoring.
 - Command: `sudo /usr/bin/tegrastats`
- **Installing jtop:** An advanced tool for monitoring the Jetson Nano.
 - Installation: `sudo -H pip3 install jetson-stats`
 - Usage: `sudo jtop`

Monitoring System Resources Post

On-Device Training in **ONNX Runtime (ORT)** allows models to be trained directly on edge devices, enhancing user privacy by keeping data local. ORT, a cross-platform engine, supports various machine learning models and now extends to on-device training, making it simpler for developers to train models with data on the device for a personalized experience. This capability is designed to be efficient in memory and performance, fitting the constraints of edge devices and supporting federated learning for privacy-preserving global model updates.

- **On-Device Training with ORT**
-