A Framework For Evaluating Visual SLAM

Jan Funke
Jan.Funke@inf.tu-dresden.de

Tobias Pietzsch
Tobias.Pietzsch@inf.tu-dresden.de

Fakultät Informatik
Technische Universität Dresden
01062 Dresden, Germany

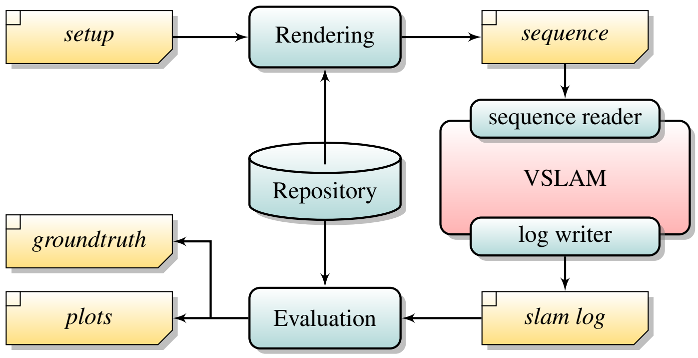

Figure 1: Framework overview. An image sequence is rendered and passed to the evaluated VSLAM system. A log of the systems operation is created and processed by the *evaluation system*. From the known scene structure and trajectory, the log is used to compute a sparse ground truth feature map, as well as ground truth (or rather “ideal”) measurements. Finally, plot data is generated comparing estimation results from the log against ground truth.

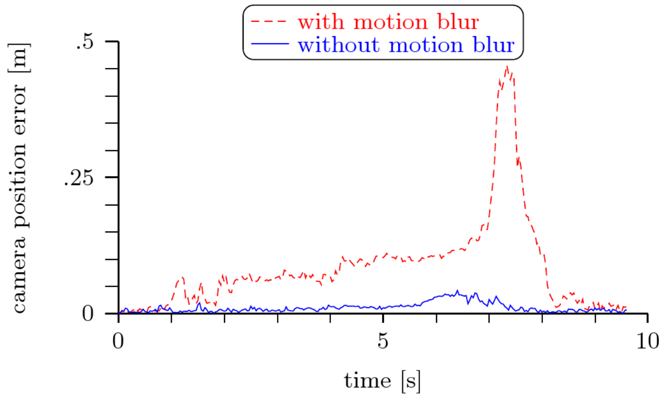

Figure 2: Visualisation of the errors that heavy motion blur causes to the estimation of the camera position.

Performance analysis in the field of camera-based simultaneous localisation and mapping (Visual SLAM, VSLAM) is still an open problem. There is a lack of generally accepted performance measures, test frameworks, and benchmark problems. Most researchers test by visually inspecting their systems on recorded image sequences (e.g., [3, 4, 6]) or measuring accuracy on simulated data of simplified point-cloud-like environments (e.g., [1, 2]). Both approaches have their disadvantages. Recorded sequences lack ground truth, and performance is usually described in broad categories such as whether loops are closed successfully or not. Simulations tend to oversimplify some aspects of the problem. For instance they abstract from the low-level image processing layer.

In this paper, we propose to evaluate VSLAM performance on rendered image sequences. The intention is to move simulations towards more realistic conditions while still having ground truth. High-quality rendered images are close enough to real-world images to violate many of the assumptions usually made in the image processing part of VSLAM. This allows for a deeper study and evaluation of the whole VSLAM system from the lowest to the highest level.

For this purpose, we provide a complete and extensible framework which addresses all aspects, from rendering to ground truth generation and automated evaluation. As illustrated in Figure 1, the framework comprises a *rendering system* and an *evaluation system*. Both systems use a common *repository* of scene, camera and trajectory descriptions. The *rendering system* is used to create image sequences from a specific combination of such *repository* items. Images are rendered using the open source ray-tracer POV-Ray. Artifacts such as non-Lambertian surfaces are naturally available in a ray-tracer. Additionally, the *rendering system* provides image degradation through motion blur and additive Gaussian noise.

The rendered sequences are then read by the to-be-evaluated VSLAM system for processing and the results are stored in a set of log files. Our framework provides a lightweight C++ interface which aides in both these tasks. A *sequence reader* replaces the regular camera driver of the VSLAM system. It provides access to the rendered images, as well as the intrinsic camera parameters. A *log writer* creates structured log data during the experiment. Log data is stored in a set of XML files which are processed by the *evaluation system* afterwards.

The *evaluation system* consists of an extensible set of Python scripts that perform various evaluation tasks, e.g., the computation of ground truth and the creation of data plots for the experiments. Among other data, the log includes feature initialisation events, i.e., when and where in the image the VSLAM system chose to initialise new features. Together with the known trajectory and scene this initialisation log is used to generate a ground truth sparse feature map. This ground truth map can in turn be used to generate ground truth “ideal” measurements. Finally, log data is evaluated with respect to the ground truth. Currently, we provide evaluation scripts to calculate the mean map error per frame or feature, to analyse the camera trajectory error with respect to position and rotation, and to analyse the measurement errors of features in the image plane.

The main motivation for our work was the possibility to quickly and easily set up and carry out simulated (but realistic) VSLAM experiments. As we are convinced that this is of interest to other researchers as well, our goal in developing this framework was to make it easily accessible to the community. The framework is built exclusively on open source software. All information that is exchanged between the components, as well as all intermediate results (such as ground truth) are stored in XML formats. For easy extensibility we provide basic routines for parsing the XML files and resolving dependencies between evaluation scripts. Source code and documentation are publicly available at [5].

Besides giving a more in-depth description of the framework, the paper also illustrates the kind of experiments that can be carried out by running VSLAM systems on simple generated sequences: We can compare the trajectory, map, and measurement error for different settings of motion blur (on an otherwise identical sequence, see figure 2 for an example plot). We can analyse the effect of sub-pixel accurate matching on measurement error. The framework can even be used to quickly evaluate hypotheses, e.g., “What would be the benefit of accurate estimation of feature normals?”.

[1] A. Chiuso, P. Favaro, H. Jin, and S. Soatto. Structure from Motion Causally Integrated Over Time. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):523–535, April 2002.

[2] Javier Civera, Andrew J. Davison, and J. M. M. Montiel. Inverse Depth to Depth Conversion for Monocular SLAM. In *ICRA 2007*, 2007.

[3] Laura A. Clemente, Andrew J. Davison, Ian Reid, José Neira, and Juan D. Tardós. Mapping Large Loops with a Single Hand-Held Camera. In *RSS*, 2007.

[4] E. Eade and T. Drummond. Unified Loop Closing and Recovery for Real Time Monocular SLAM. In *BMVC 2008*, 2008.

[5] Jan Funke and Tobias Pietzsch. VSLAM evaluation framework. URL `http://vslam.inf.tu-dresden.de`.

[6] Georg Klein and David Murray. Improving the agility of keyframe-based slam. In *ECCV 2008*, volume 5303, pages 802–815, 2008. Springer.