



**HAL**  
open science

## Image quantization towards data reduction: robustness analysis for SLAM methods on embedded platforms

Quentin Picard, Stephane Chevobbe, Mehdi Darouich, Jean-Yves Didier

### ► To cite this version:

Quentin Picard, Stephane Chevobbe, Mehdi Darouich, Jean-Yves Didier. Image quantization towards data reduction: robustness analysis for SLAM methods on embedded platforms. ICIP 2022 - The 29th IEEE International Conference on Image Processing, Oct 2022, Bordeaux, France. pp.4158-4162, 10.1109/ICIP46576.2022.9897315 . cea-03858795

**HAL Id: cea-03858795**

**<https://cea.hal.science/cea-03858795v1>**

Submitted on 17 Nov 2022

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# IMAGE QUANTIZATION TOWARDS DATA REDUCTION: ROBUSTNESS ANALYSIS FOR SLAM METHODS ON EMBEDDED PLATFORMS

Quentin Picard<sup>\*†</sup>

Stephane Chevobbe<sup>\*</sup>

Mehdi Darouich<sup>\*</sup>

Jean-Yves Didier<sup>†</sup>

<sup>\*</sup> Université Paris-Saclay, CEA, List, F-91120, Palaiseau, France

<sup>†</sup>IBISC, Univ Evry, Université Paris-Saclay, 91025, Evry, France

quentin.picard@cea.fr

## ABSTRACT

Embedded simultaneous localization and mapping (SLAM) aims at providing real-time performances with restrictive hardware resources of advanced perception functions. Localization methods based on visible cameras include image processing functions that require frame memory management. This work reduces the dynamic range of input frame and evaluates the accuracy and robustness of real-time SLAM algorithms with quantified frames. We show that the input data can be reduced up to 62% and 75% while maintaining a similar trajectory error lower than 0.15m compared to full precision input images.

**Index Terms**— SLAM, localization, image processing, embedded systems, real-time, frame reduction

## 1. INTRODUCTION

Real-time localization methods such as simultaneous localization and mapping/visual-inertial odometry (SLAM/VIO) are used in the context of autonomous and mobile systems, such as service robots, MAVs (micro air vehicles) and AR/VR (augmented/virtual reality) devices. These algorithms implement advanced functions that must provide real-time performances with restrictive hardware resources.

The goal in embedded SLAM is to minimize the power consumption, memory consumption, latency while maintaining real-time processing. Many works have implemented SLAM/VIO algorithms in different embedded platforms to demonstrate that parameters have to be fine-tuned for real-time performances. In [1], the number of extracted features in image processing functions has been reduced to 40% in average for the VIO methods implemented in two hardware platforms for MAVs, ODROID and UpBoard. The works in [2] proposes the implementation of advanced localization functions in an energy-efficient accelerator for AR/VR applications. It proposes to reduce the use of frame buffers in the VIO front-end by a factor of 4 by applying the quantization process on the input frame that minimizes the localization error.

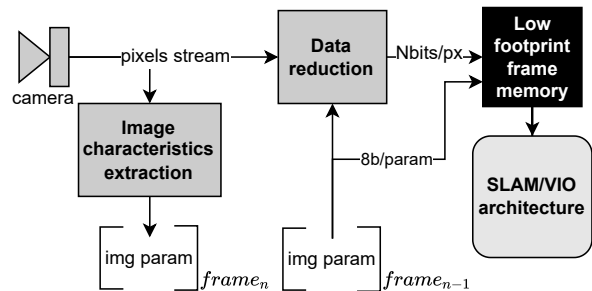


Fig. 1: Input frame reduction approach in SLAM pipeline.

This work aims at reducing the memory footprint of SLAM/VIO algorithms by evaluating the impact of the pixel dynamic reduction on the accuracy and the robustness of the localization. Our main contribution shows that we can reduce the memory footprint of frame buffers with the quantization of input frames while maintaining the error of localization equivalent to the one obtained with full precision images in various SLAM/VIO front-end techniques. The goal is to define a dynamic reduction method that includes pixels stream as illustrated in Figure 1.

This paper is organized as follows: the used quantization methods are described in Section 2. In Section 3 the SLAM/VIO algorithms selection are detailed. Then, Section 4 provides the methodology employed and the experimental results.

## 2. QUANTIZATION METHODS FOR MEMORY FOOTPRINT REDUCTION

The quantization methods used to reduce the memory footprint are based on the dynamic reduction of the entire input image or per block of pixels. Three approaches are used, the median cut (MC) [3], the block-wise quantization (BW) [2] and the naive quantization method (NV). The latter is used to reduce the pixel dynamics for reference purposes. It involves assigning the value of the corresponding bin that contains the intensity value to the pixel.

## 2.1. Median cut quantization

The median cut method [3] is a pixel value quantization approach used to generate a range of  $2^n$  intensity values, with  $n$  the number of bits (cf. Algo. 1). The output of the algorithm is a 2D array, representing the pixels map, associated with a small table containing the candidate pixel values (average of each subarray). The stored values in the 2D array are the position of the candidate pixel value in the table. This method represents the input pixels in  $\log_2 n$  bit(s). The data reduction is up to 87% for a 1 bit per pixel representation with an input image resolution of  $752 \times 480$ .

---

### Algorithm 1 Median Cut algorithm

---

1. Move all pixels into a single array
  2. Sort each intensity values
  3. Find the median of the array
  4. Cut the array by the median index
  5. Compute the average of each subarray
  6. Repeat from 3. to obtain a palette of  $2^n$  intensity values
- 

## 2.2. Block-wise quantization

The block-wise quantization method [2] quantizes the input image per block of  $N \times N$  pixels, with  $N$  corresponding to the size of pixels. Algorithm 2 describes that each intensity value in the  $N \times N$  block is either equal to the defined threshold or the minimum value. This technique represents every pixel in the image divided into blocks of size  $N$  in 1 bit, the threshold and the minimum value in 8 bit each. Therefore, a  $4 \times 4$ ,  $8 \times 8$  and  $16 \times 16$  block is quantized for a 2 bits, 1.25 bits and 1.06 bits per pixel representation respectively. The percentage of data reduction is calculated based on  $(nb_{px} * n)/8$ , with  $nb_{px}$  the number of pixels in the input image. With a  $752 \times 480$  image, the data reduction for each block size configuration corresponds to 75%, 84% and 87% respectively.

---

### Algorithm 2 Block-wise quantization function

---

```

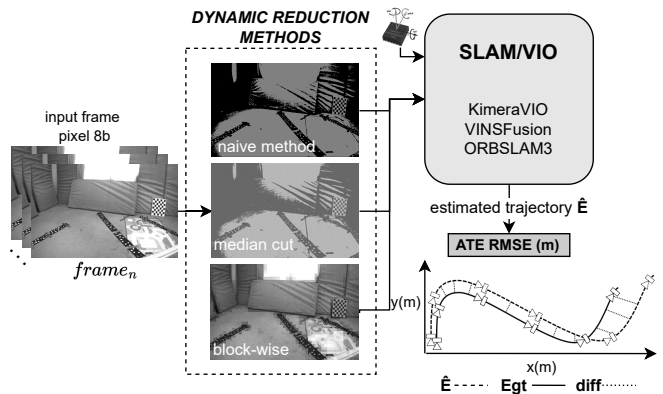
block = N × N                                ▷ N = 4; 8 or 16
for each pixel(i, j) in N × N block do
    Find max() and min() values
    thresh = (max + min)/2                    ▷ Define a threshold
    if value(i, j) >= thresh then
        value(i, j) = thresh
    else
        value(i, j) = min

```

---

## 3. SLAM/VIO ALGORITHMS SELECTION

There is a wide variety of SLAM and VIO methods available in the literature. We identified and selected three localization methods based on the criteria of localization accuracy and



**Fig. 2:** Methodology employed to evaluate the estimated trajectory  $\hat{E}$  based on three dynamic reduction methods.

computational complexity. As we want to measure the robustness of localization algorithms as a function of the reduction of the pixel dynamics, we have selected methods whose front ends are either based on feature tracking (FT) [4] or feature detection and description (FD) [5] techniques.

For this study we have chosen 3 methods that have been ported to embedded hardware targets. Each one of them represent different trade-offs between computational complexity and localization accuracy.

ORB\_SLAM3 [6] is a multi-map visual-inertial SLAM system based on FD that relies on the maximum a posteriori estimation. The method has been implemented on Raspberry Pi 3B+ and Jetson embedded platforms [7, 8].

KimeraVIO [9] is the first module of the Kimera’s architecture and provides the estimated localization based on FT. KimeraVIO has been used on Jetson TX2 [10].

VINSFusion [11] is based on VINSMono [12] and also uses FT as the early stage of the pipeline. In [1], VINSMono has been implemented on the ODROID and UpBoard embedded platforms.

## 4. EXPERIMENTAL RESULTS

### 4.1. Methods

In order to evaluate the robustness of localization to the reduction of image dynamics we set up an execution pipeline that transforms the input images according to the three selected quantization methods (cf. Fig. 2). The input images feed the dynamic reduction methods to output the naive method, the median cut or the block-wise. The input of the SLAM/VIO methods include inertial measurements and the frames with dynamic reduction. The absolute trajectory error root mean square error (ATE RMSE) is computed based on the difference of the aligned estimated trajectory ( $\hat{E}$ ) to the ground truth ( $E_{gt}$ ) with the  $SE(3)$  Umeyama alignment.

The default configurations have been used for ORBSLAM3 [6] and VINSFusion [11]. In order to compare

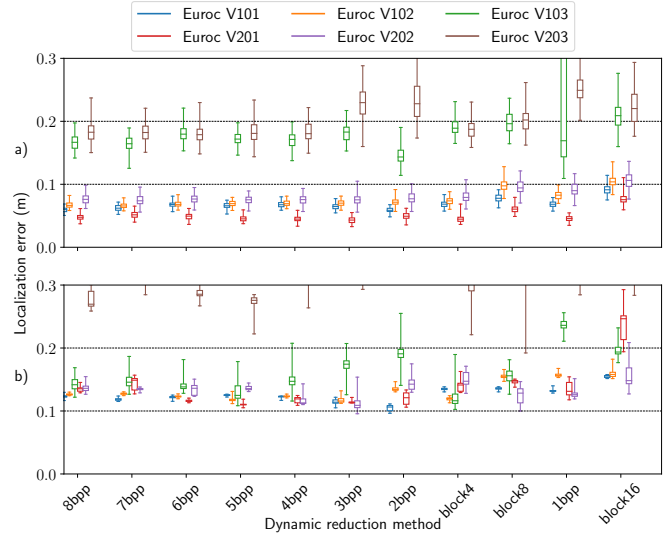
KimeraVIO [9] with these algorithms, three parameters have been modified. The *autoInitialize* is set as true to initialize the process with inertial measurements instead of the provided ground truth from the dataset. The parameters *deterministic\_random\_number\_generator* is set to false and *ransac\_randomize* to true. The evaluation of the trajectory from the ground truth is based on the ATE RMSE metric [13] with the *evo* package [14] (version 1.13.5 and relying on Python version 3.6.9).

The Euroc dataset [15] has been used to assess the accuracy of the localization with quantified input images. It provides 11 sequences recorded by an UAV equipped with a visual-inertial sensor unit providing stereo frames and inertial measurements. The *Vicon* sequences have been used in this study. They present various characteristics from slow motion with bright scene to fast motion with motion blur for the difficult sequences. Each sequence has been pre-processed with the dynamic reduction methods and then injected into each localization methods to retrieve the corresponding estimated trajectory. The accuracy error results of non-deterministic localization methods were computed over 100 runs for the three quantization methods per sequence. Therefore, we had a total of 54000 runs.

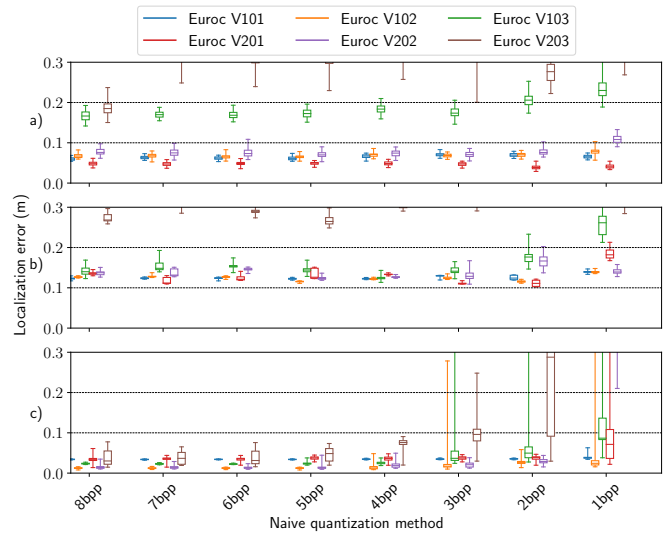
#### 4.2. Performance analysis

In order to define the right quantization method, the localization error for each SLAM/VIO algorithms is plotted as a function of the data reduction method. Figures 3 and 4 show the variability of the localization error. The baseline of comparison is referred as 8bpp (8 bits/px). The median cut and naive quantization is denoted as 7bpp to 1bpp. The block-wise quantization is denoted as blockN with N, the size of the block. The quantization configurations are defined as the dynamic reduction range and are sorted in ascending order based on the percentage of data reduction. For each configuration, the boxplots correspond from left to right to the results with the Euroc sequences *V101*, *V102*, *V103*, *V201*, *V202*, *V203*. Easy sequences are denoted as *V101*, *V201*, moderate ones as *V102*, *V202* and difficult ones as *V103*, *V203*.

In Figure 3, KimeraVIO presents a performance of localization similar to the baseline with significant dynamic reduction, especially for the easy and moderate sequences. In V201, the accuracy of the 1bpp configuration is even higher than the 8bpp but the localization is highly deteriorated for the hardest sequences with 1bpp. The block4 configuration results show that the accuracy is similar to the baseline with a memory footprint reduction of 75%. The results of VINS-Fusion show that the variability of error is increasing for a majority of sequences with quantified input images especially for hard sequences. Easy sequences are not impacted by the significant data reduction except for the block16 configuration.



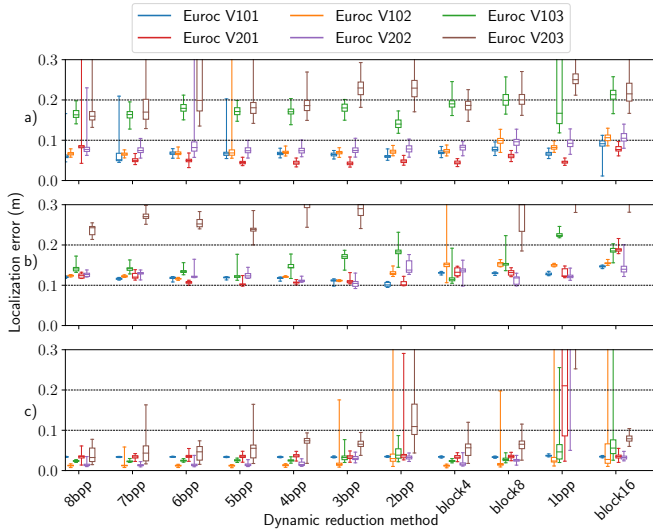
**Fig. 3:** Localization error (100 runs): a) KimeraVIO, b) VINSFusion with the MC and BW.



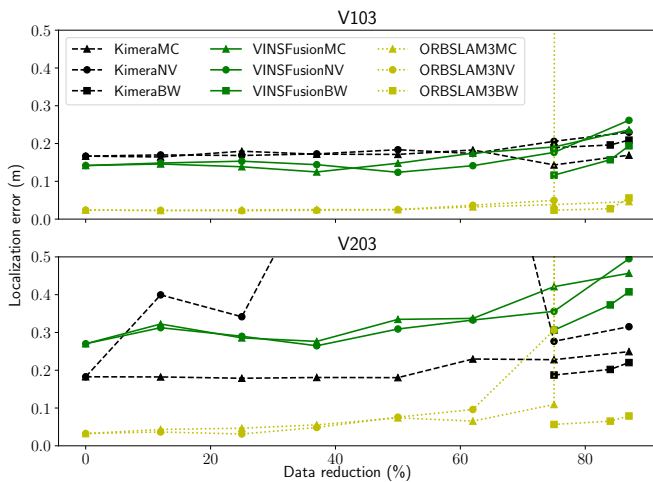
**Fig. 4:** Localization error (100 runs) : a) KimeraVIO, b) VINSFusion, c) ORBSLAM3 with the NV.

The NV results are shown in Figure 4 from 8bpp to 1bpp. This method provides globally similar results to the median cut algorithm but with less stable behavior. The latter provides a higher accuracy on the low dynamic reduction, such as 2bpp and 1bpp for KimeraVIO and VINSFusion. In Figure 4c), the localization error is below 0.1m for all sequences until the 4bpp configuration. No data has been retrieved from the 1bpp configuration on V203 as the input image does not provide enough textures for FD.

Figure 5 shows the variability of the localization error for SLAM algorithms with loop closure (lc) parameter activated with KimeraVIO-lc in 5a), VINSFusion-lc in 5b) and



**Fig. 5:** Localization error (100 runs) a) KimeraVIO-lc, b) VINSFusion-lc, c) ORBSLAM3 with MC and BW.



**Fig. 6:** Localization error (RMSE in meters) based on the median results for KimeraVIO, VINSFusion and ORBSLAM3 for each dynamic reduction method.

ORBSLAM3 in 5c). Surprisingly, the localization error is higher with *V201* and *V202* for the baseline than the quantized configurations in 5a). Overall the robustness of the localization is more impacted compared to the configuration without lc. VINSFusion-lc provides robust results until 2bpp for MC and for the block8 configuration except for the *V203* sequence. ORBSLAM3 is robust to the MC until 3bpp and does not estimate an accurate trajectory for the first moderate and hard sequences on the block16 configuration. The localization method based on feature detection and description provides more robustness compared to the feature tracking technique employed in 5a) and 5b) thanks to the binary feature descriptor [5].

Figure 6 shows the median of the localization error for KimeraVIO, VINSFusion and ORBSLAM3 with MC, NV and BW methods on the difficult sequences. The VIO accuracy has small variation of the localization error while significantly reducing the input dynamic range data up to 87%. ORBSLAM3 is robust until 3bpp for MC and NV. The BW quantization allows a median accuracy similar to full precision images with a data reduction up to 87%. The *V203* sequence results for KimeraNV have an important loss of accuracy. The accuracy similar to 8bpp is then retrieved at 2bpp. Results for KimeraMC show low localization errors with a high data reduction.

The choice of the dynamic reduction method depends on the SLAM method and the difficulty of the dataset. Overall, the block4 configuration proposes a good tradeoff between the input data reduction of 75% and the equivalent accuracy with the baseline.

## 5. CONCLUSION

This work proposes to evaluate the robustness of SLAM/VIO algorithms based on different front-end techniques with quantified input images with data reduction up to 87%. Three data reduction approaches have been selected, the median cut, the block-wise and the naive quantization methods.

Overall, we show that real-time localization methods are robust to the dynamic reduction of input pixels. The proposed algorithms maintain a similar accuracy to the full precision baseline. It is also highly dependent to the difficulty of the dataset. The results shows that SLAM/VIO methods are robust to quantized images up to 3bpp for the median cut and with the block4 configuration for all datasets. These configurations provide a data reduction of 62% and 75% respectively. The latter offers a good tradeoff between accuracy, robustness and percentage of data reduction.

This study leads to the following perspectives. The memory footprint reduction in SLAM functions where frame buffers are used. This would quantify the impact of the dynamic range reduction on specific parts of the pipelines. It requires a review of the localization implementation based on restrictive resources available. Then, the reduction of the input pixels dynamic range from cameras characteristics used for real-time localization systems.

## 6. REFERENCES

- [1] Jeffrey Delmerico and Davide Scaramuzza, "A benchmark comparison of monocular visual-inertial odometry algorithms for flying robots," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 2502–2509.
- [2] Amr Suleiman, Zhengdong Zhang, Luca Carlone, Sertac Karaman, and Vivienne Sze, "Navion: A 2-mW fully integrated real-time visual-inertial odometry accelerator for autonomous navigation of nano drones," *IEEE Journal of Solid-State Circuits*, vol. 54, no. 4, pp. 1106–1119, apr 2019.
- [3] Paul Heckbert, "Color image quantization for frame buffer display," *SIGGRAPH Comput. Graph.*, vol. 16, no. 3, pp. 297–307, jul 1982.
- [4] Jean yves Bouguet, "Pyramidal implementation of the lucas kanade feature tracker," *Intel Corporation, Microprocessor Research Labs*, 2000.
- [5] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *2011 International Conference on Computer Vision*, 2011, pp. 2564–2571.
- [6] Carlos Campos, Richard Elvira, Juan J. Gómez Rodríguez, José M. M. Montiel, and Juan D. Tardós, "Orb-slam3: An accurate open-source library for visual, visual-inertial, and multimap slam," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, 2021.
- [7] O. C. B. Silveira, J. G. O. C. de Melo, L. A. S. Moreira, J. B. N. G. Pinto, L. R. L. Rodrigues, and P. F. F. Rosa, "Evaluating a visual simultaneous localization and mapping solution on embedded platforms," in *2020 IEEE 29th International Symposium on Industrial Electronics (ISIE)*, 2020, pp. 530–535.
- [8] Stefano Aldegheri, Nicola Bombieri, Domenico D. Bloisi, and Alessandro Farinelli, "Data flow orb-slam for real-time performance on embedded gpu boards," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 5370–5375.
- [9] Antoni Rosinol, Marcus Abate, Yun Chang, and Luca Carlone, "Kimera: an open-source library for real-time metric-semantic localization and mapping," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2020.
- [10] Jinwoo Jeon, Sungwook Jung, Eungchang Lee, Duckyu Choi, and Hyun Myung, "Run your visual-inertial odometry on nvidia jetson: Benchmark tests on a micro aerial vehicle," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5332–5339, 2021.
- [11] Tong Qin, Shaozu Cao, Jie Pan, and Shaojie Shen, "A general optimization-based framework for global pose estimation with multiple sensors," *arXiv preprint arXiv:1901.03642*, 2019.
- [12] T. Qin, P. Li, and S. Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.
- [13] Zichao Zhang and Davide Scaramuzza, "A tutorial on quantitative trajectory evaluation for visual(-inertial) odometry," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 7244–7251.
- [14] Michael Grupp, "evo: Python package for the evaluation of odometry and slam.," <https://github.com/MichaelGrupp/evo>, 2017.
- [15] Michael Burri, Janosch Nikolic, Pascal Gohl, Thomas Schneider, Joern Rehder, Sammy Omari, Markus W Achtelik, and Roland Siegwart, "The EuRoC micro aerial vehicle datasets," *The International Journal of Robotics Research*, vol. 35, no. 10, pp. 1157–1163, jan 2016.