## 7.3 An 879GOPS 243mW 80fps VGA Fully Visual CNN-SLAM Processor for Wide-Range Autonomous Exploration

Ziyun Li, Yu Chen, Luyao Gong, Lu Liu, Dennis Sylvester, David Blaauw, Hun-Seok Kim

University of Michigan, Ann Arbor, MI

Simultaneous localization and mapping (SLAM) estimates an agent's trajectory for all six degrees of freedom (6 DoF) and constructs a 3D map of an unknown surrounding. It is a fundamental kernel that enables head-mounted augmented/virtual reality devices and autonomous navigation of micro aerial vehicles. A noticeable recent trend in visual SLAM is to apply computation- and memory-intensive convolutional neural networks (CNNs) that outperform traditional hand-designed feature-based methods [1]. For each video frame, CNN-extracted features are matched with stored *keypoints* to estimate the agent's 6-DoF pose by solving a *perspective-n-points* (PnP) non-linear optimization problem (Fig. 7.3.1, left). The agent's long-term trajectory over multiple frames is refined by a *bundle adjustment* process (BA, Fig. 7.3.1 right), which involves a large-scale (~120 variables) non-linear optimization. Visual SLAM requires massive computation (>250GOP/s) in the CNN-based feature extraction and matching, as well as data-dependent dynamic memory access and control flow with high-precision operations, creating significant low-power design challenges. Software implementations are impractical, resulting in 0.2s runtime with a ~3GHz CPU+ GPU system with >100MB memory footprint and >100W power consumption. Prior ASICs have implemented either an incomplete SLAM system [2,3] that lacks estimation of ego-motion or employed a simplified (non-CNN) feature extraction and tracking [2,4,5] that limits SLAM quality and range. A recent ASIC [5] augments visual SLAM with an off-chip high-precision inertial measurement unit (IMU), simplifying the computational complexity, but incurring additional power and cost overhead.

This paper presents an accurate, low-power, real-time CNN-SLAM processor that implements full-visual SLAM on a single chip. The proposed major design features are 1) feature extraction with a highly parallelized and programmable CNN engine with 32% better accuracy in feature matching than SIFT; 2) aggressively pruned feature matching with temporal pose prediction and address hashing to eliminate 97% of unnecessary matchings; 3) numerically stable fixed-point implementation with function re-organization and pivoting in the linear solver; and 4) hierarchical memory organization, eliminating external DRAM accesses for BA optimization. We report the ASIC's SLAM performance on the industrial standard KITTI benchmark that renders large-scale automobile trajectory over 1km. The design supports localization and mapping of >1000 keypoints/frame on VGA (640×480) resolution in real-time at 80fps, consuming 243.6mW from a 0.9V supply in 28nm CMOS.

Figure 7.3.1 shows the overall procedure of the SLAM system, consisting of frame-by-frame PnP and multi-frame BA. PnP processing computes the 6-DoF pose (combination of 3D translation $\boldsymbol{T}$ and 3D rotation $\boldsymbol{R}$) for each frame by analyzing the projected coordinate differences of matched keypoints in two subsequent frames. CNN-extracted keypoints are matched with others in the previous frame to establish correspondence of the same points in the environment. The 6-DoF pose is obtained by solving a non-linear optimization to minimize the 2D reprojection error of these matched points. After acquiring the pose of the current frame, all newly added keypoints are projected onto 3D real-world coordinates for PnP processing of the next frame. A frame that contains ≥50% new keypoints is registered as a *keyframe*. To refine the long-term trajectory of the camera, BA is performed over the last 20 keyframes, whenever a new keyframe is registered.

Figure 7.3.2 shows the overall architecture of the processor, including a programmable CNN engine, PnP engine, and BA engine. The CNN engine has a mesh of 512 MAC units in 8 clusters (8×8 MACs each), a 192KB memory for CNN weights/activations, and two interleaved 32KB image buffers for streaming (Fig. 7.3.3, left). The CNN uses 8b weights and does not require any off-chip weight loading. If the current frame is identified as a keyframe, the list of 3D world and 2D projected coordinates of keypoints is sent to the graph memory in the BA engine, where iterative Levenberg–Marquardt (LM) optimization is performed over the last 20 keyframes.

Figure 7.3.3 details the operation of the programmable CNN engine. First, 1D convolutions are performed with shifting input activations (IAs) using a row of 8 MAC units with k (kernel size) cycles. This operation is repeated on a second and third row of IAs to complete the 2D convolution. Pooling and extrema detection are performed in the same fashion as 2D convolution in the MAC array. Partial 2D convolution outputs are accumulated locally in each MAC unit. Using 8 clusters of 8×8 MAC arrays, 8 consecutive input channels are convolved in parallel to accumulate partial output activations with 8 accumulators, and 8 consecutive output channels are processed in parallel to maximize reuse of the same IAs. A 4-layer CNN network used for SLAM evaluation is shown in Fig. 7.3.3 (bottom left). The proposed architecture computes convolution, pooling, and ReLU together rather than treating them as separate layers with buffering of inputs-outputs in between. For keypoint detection, the engine is programmed to compute the difference of Gaussian filters.

As each frame contains ~1000 keypoints, enumerating all possible matchings (~1000×1000) is extremely costly. Addressing this challenge, we propose a *prediction-based* pruned keypoint matching scheme depicted in Fig. 7.3.4. We use a locally linearized camera movement model to predict the new pose of the current frame from the poses of the previous two frames. Based on this pose prediction, the search range for each keypoint is reduced to ~±24 pixels/dimension on the 2D projected image. This eliminates 97% of unnecessary matchings with negligible accuracy degradation (<0.1%). Prediction-based matching efficiency is further improved by employing a hierarchical memory system (492KB) using keypoint position-based hashing to store/load feature descriptors and 3D coordinates. The hash input is the keypoint location, and its output points to the keypoint entry in the memory that stores a list of feature descriptor and 3D coordinates at that location. Each feature has 64 elements, and the feature matching cost is calculated with 8 parallel processing units, taking up to 8 cycles. Early termination reduces this to ~4.8 cycles when feature elements mismatch in early stages (Fig. 7.3.4).

BA processing involves 20 keyframes and 4096 keypoints stored in the hierarchical graph memory (Fig. 7.3.5). Because each keypoint appears on multiple keyframes, each entry is structured to contain a single 3D coordinate and multiple (≤8) 2D projected coordinates associated with different keyframe IDs. A separate FIFO serves to eliminate and insert keypoints into the memory, while matched keypoints are merged into a single entry. In each BA iteration, the keypoints' 3D coordinates are updated according to the frame pose, and very small increments are applied to numerically compute the Jacobian matrix. Computing Jacobians requires an extremely large dynamic range and high precision; thus, prior works [4,5] used double-precision floating-point operations. We reformulate the reprojection so that common offset is subtracted before normalizing the projected 2D points into homogeneous coordinates. This allows a 32b fixed point implementation with ~40% energy reduction, while maintaining numerical stability in computing the Jacobian. After linearization, we construct a sparse Hessian matrix that only has non-zero 6×6 submatrices diagonally that relate to the 6-DoF pose of each keyframe. Thanks to the sparse and deterministic matrix structure, we block partition the Hessian and solve each 6×6 submatrix sequentially instead of solving the full 120×120 linear system. We apply pivoting to Gaussian elimination (GE) to maintain numerical stability in the linear system solver. With pivoting, non-zero elements in other rows are eliminated with the maximum element of the selected row. Row shuffling is performed before back substitution (BS), and GE/BS share 6 parallel computing units.

The SLAM processor is fabricated in 28nm HPC CMOS. Real-time image input and 6-DoF SLAM output are streamed using USB3.0. Fig. 7.3.6 (top left) shows the trajectory produced by the chip for KITTI automobile scenes with >1000 images and >500m range, tracking >1000 keypoints per frame. Fig. 7.3.6 (top right) shows frequency/energy scaling of the chip across voltage. At 0.9V nominal voltage, the VGA processing latency is 12.5ms. It achieves 97.9% accuracy in translation and 99.34% in rotation on KITTI rendering automotive scenes over 1km. The chip consumes 243.6mW to process 80fps VGA at 3.6TOPS/W, marking a 15× improvement in performance and 1.44× in energy efficiency over listed prior works. Prior works use hand-crafted features or off-chip IMU and do not support large-scale KITTI evaluation. Power reduces to 61.8mW for VGA at 30fps @ 0.63V, yielding 48% additional energy efficiency. Fig. 7.3.7 shows the die photo and a performance summary.

*Acknowledgements:*
We thank TSMC the University Shuttle Program for chip fabrication and Samsung for funding.

*References:*
[1] K. Tateno, et al., "CNN-SLAM: Real-time dense monocular SLAM with learned depth prediction," *Computer Vision and Pattern Recognition*, vol. 2, 2017.
[2] Z. Li, et al., "A 1920× 1080 30fps 2.3 TOPS/W Stereo-Depth Processor for Robust Autonomous Navigation," *ISSCC*, pp. 62-63, 2017.
[3] J. Yoon, et al., "A Unified Graphics and Vision Processor With a 0.89 μW/fps Pose Estimation Engine for Augmented Reality," *IEEE TVLSI,* vol. 21, no. 2, pp. 206-216, 2013.
[4] I. Hong, et al., "A 27 mW Reconfigurable Marker-Less Logarithmic Camera Pose Estimation Engine for Mobile Augmented Reality Processor," *IEEE JSSC*, vol. 50, no. 11, pp. 2513-2523, 2015.
[5] A. Suleiman, et al., "Navion: A Fully Integrated Energy-Efficient Visual-Inertial Odometry Accelerator for Autonomous Navigation of Nano Drones," *IEEE Symp. VLSI Circuits*, 2018.

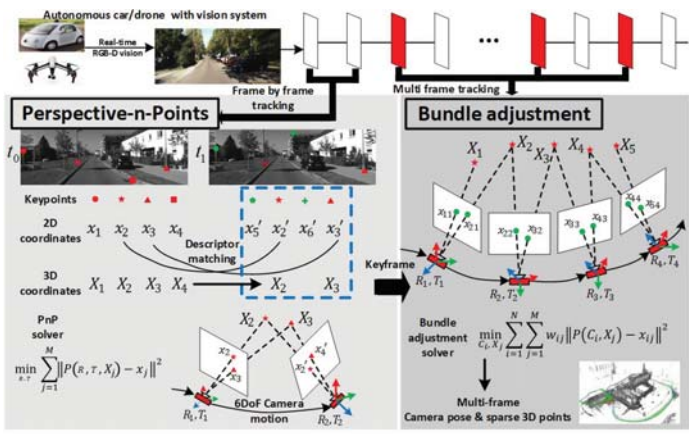



Figure 7.3.1: Proposed visual SLAM system for autonomous vehicles and associated processing procedures.

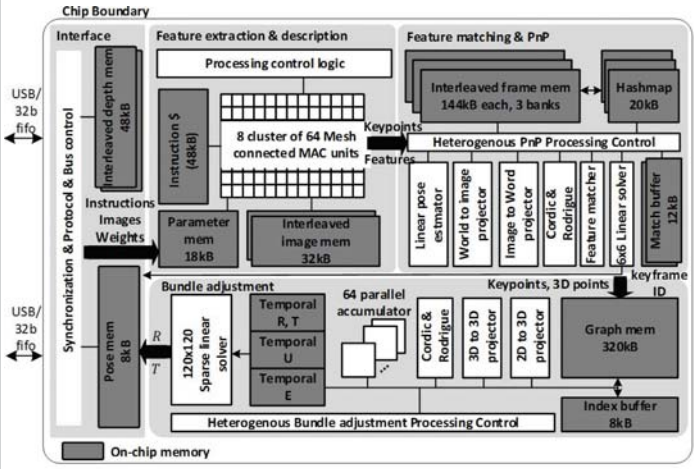
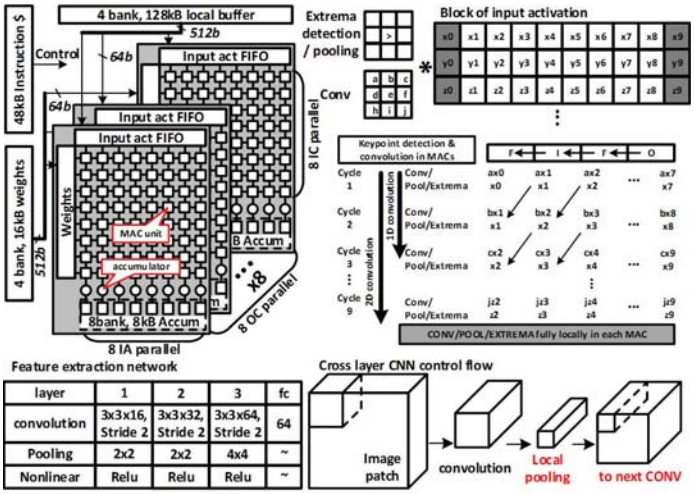



Figure 7.3.2: Overall CNN-SLAM chip architecture.




| layer | 1 | 2 | 3 | fc |
|---|---|---|---|---|
| convolution | 3x3x16, Stride 2 | 3x3x32, Stride 2 | 3x3x64, Stride 2 | 64 |
| Pooling | 2x2 | 2x2 | 4x4 | ~ |
| Nonlinear | Relu | Relu | Relu | ~ |

Figure 7.3.3: Proposed CNN processor architecture with reconfigurable keypoint detection and feature description.

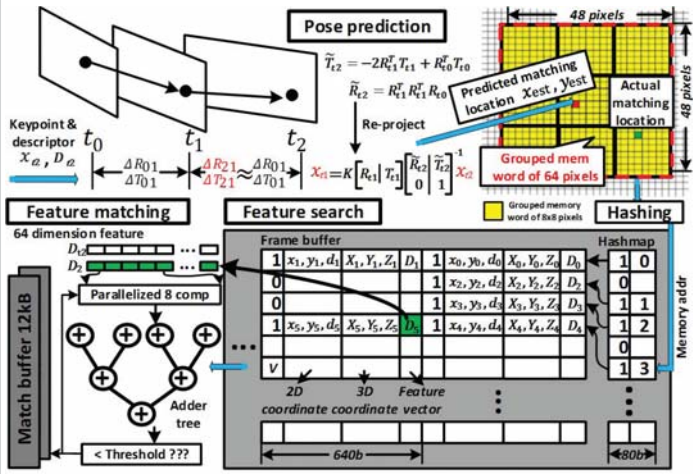



Figure 7.3.4: Proposed frame-based tracking with pruned feature matching with linear pose estimation.

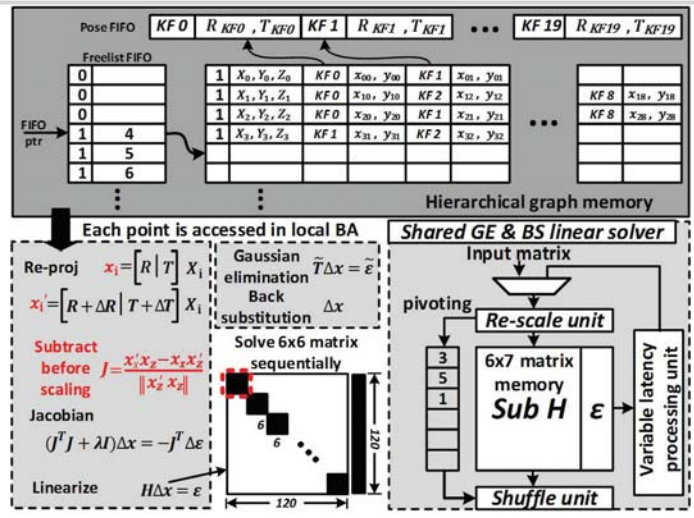
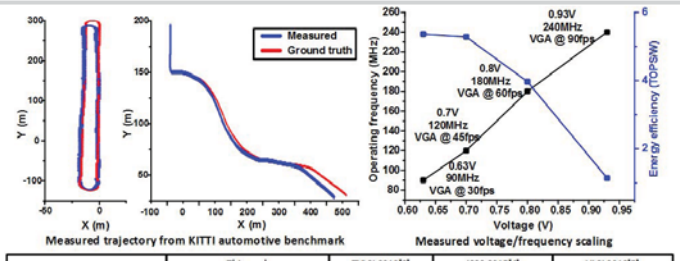
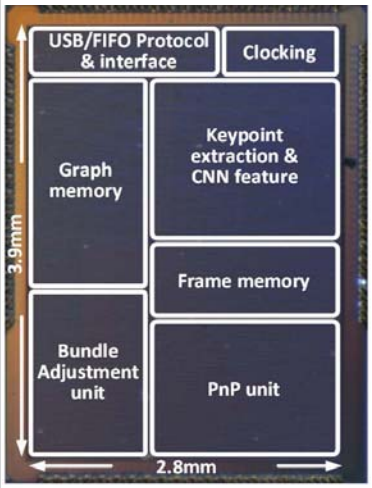



Figure 7.3.5: Hierarchical graph memory organization and proposed numerically stable fixed-point implementation with function re-formulation and pivoting in linear solver.




| | | This work | TVLSI 2013[3] | JSSC 2015[4] | VLSI 2018[5] |
|---|---|---|---|---|---|
| Complete Visual SLAM system | Vision frontend | √ CNN feature | √ Marker recognition | × | √ Harris feature |
| | Pose tracking | √ Visual | × | √ Visual | × Inertial sensor |
| | Graph optimization | √ Local BA | × | × | √ Local BA |
| Technology | | 28nm | 180nm | 65nm | 65nm |
| Chip area | | 10.92mm² | 28.75mm² | 3.25mm² | 20mm² |
| On-chip memory | | 1126kB | N.A | 96KB | 854kB |
| Frequency | | 240MHz | 200MHz | 133MHz | 83.3MHz |
| Throughput & image size | | 640 X 480, 80fps | 640 X 480, 100fps | 640 X 480, 95fps | 752 X 480, 90fps |
| Operation range | | ± 500m | Not reported due to limited range | | |
| Track points / frame | | 1000 | 9 | 33 | 200 |
| Operating voltage | | 0.63-0.9V | 1.8V | 1.2V | 1.2V |
| Performance | | 879.6 GOPS @ 0.9V, 215MHz 329.8 GOPS @ 0.63V, 90MHz | 153.6 GOPS | 20.2 GOPS | 10.5-59.1 GOPS |
| Power | | 243.6 mW @ 0.9V, 215MHz 61.75 mW @ 0.63V, 90MHz | 413 mW | 27 mW | 24 mW* (excluding IMU power) |
| Energy efficiency | | 3.6 TOPS/W @ 0.9V, 215MHz 5.34 TOPS/W @ 0.63V, 90MHz | 0.37 TOPS/W | 0.75 TOPS/W | 0.43-2.5 TOPS/W |

Figure 7.3.6: Chip measurements and comparison with recent prior works.




| | This work |
|---|---|
| Vision frontend | CNN feature |
| Pose tracking | PnP |
| Graph optimization | Local BA |
| Technology | 28nm |
| Chip area | 10.92mm$^2$ |
| On-chip memory | 1126kB |
| Frequency | 240MHz |
| Throughput & image size | 640 X 480, 80fps |
| Operation range | ±500m |
| Track points / frame | 1000 |
| Operating voltage | 0.9V |
| Performance | 879.6 GOPS @ 0.9V, 215MHz<br>329.8 GOPS @ 0.63V, 90MHz |
| Power | 243.6 mW @ 0.9V<br>61.75 mW @ 0.63V |
| Energy efficiency | 3.6 TOPS/W @ 0.9V<br>5.34 TOPS/W @ 0.63V |

**Figure 7.3.7: Die photo and summary of performance.**


978-1-5386-8531-0/19/$31.00 