PixVOD: Pixel-Distributed Direct Visual Odometry and Depth Estimation

Shinjeong Kim Ignacio Alzugaray Callum Rhodes Paul H. J. Kelly
Andrew J. Davison

Department of Computing, Imperial College London
{s.kim, i.alzugaray, c.rhodes, p.kelly, a.davison}@imperial.ac.uk

Abstract

Images composed of 2D pixel arrays are the standard input to computer vision algorithms, yet many underlying computations can be distributed across pixels. Transmitting raw, redundant, and noisy pixel data off the sensor remains inefficient, motivating a shift toward focal-plane sensor-processors that perform a significant part of the computation directly within each pixel. We envision pixels synthesizing higher-level signals locally, reducing downstream load, and providing richer inputs for higher-level vision tasks.

We propose a fully parallelizable form of visual odometry and depth estimation across pixels, where sensor-processors exchange information through Gaussian Belief Propagation (GBP) to achieve consensus about camera motion and infer depth from per-pixel photometric observations and a surface normal prior. To maintain geometric stability during optimization, we introduce a keyframe-like anchoring mechanism that regulates the effective baseline between frames, enabling consistent motion and depth updates.

Our method is evaluated on realistic datasets, demonstrating the feasibility of GBP-based pixel-level distributed odometry and depth estimation with keyframe anchoring on-sensor. Upon acceptance, we will release the code publicly on GitHub by the camera-ready deadline.

1. Introduction

Mobile edge applications such as robotics and wearables demand high frame rates, low latency, and minimal energy consumption. A promising path to achieving all of these is to move away from the separation of camera and processor connected by video transmission, toward hardware which combines visual sensing and processing in a single unit. In the *On Sensor Vision* paradigm, processing is built into the image sensor itself. Prototypes already exist, such as the SCAMP series of vision chips [5, 8]. These implement a Pixel Processor Array (PPA) design, where each photo-

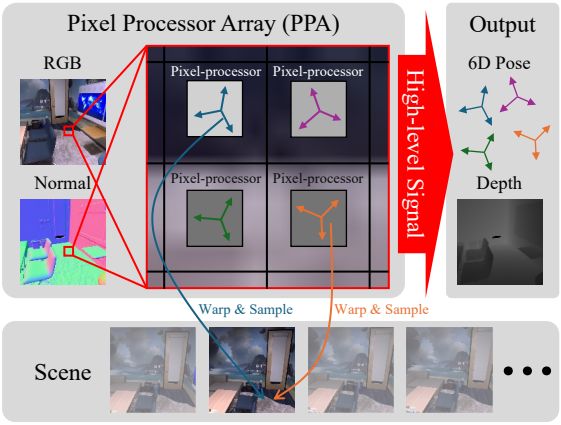


Figure 1. Most vision algorithms assume an off-sensor central processor with access to all pixels. We instead present an algorithm to track camera pose and estimate depth assuming that each pixel has its own local processing capability with only local memory and the ability to exchange messages with other pixels.

sensitive pixel has a programmable local processing capability, a set of memory registers, and the ability to exchange information with neighbours. A PPA achieves maximum performance when it uses *only* this local, pixelwise memory.

Despite the limited computational capacity of these chips, impressive on-sensor capabilities such as feature extraction and matching, object tracking, and small-CNN classification have been demonstrated [6, 17, 24]. Research is now underway to design the next generation of such chips. In this paper, we focus on the fundamental vision competence of local 6DoF motion estimation from a single camera — visual odometry (VO) — and investigate how this could be achieved with pixel-parallel computation suitable for near-future on-sensor vision devices. Previous work using SCAMP for VO/SLAM has performed feature extraction and matching on-sensor [6, 17], but required frame-rate communication with an external CPU to solve 6DoF motion estimation.

We propose an algorithm to achieve 6DoF VO estimation with pure in-pixel computation. The key challenge is to

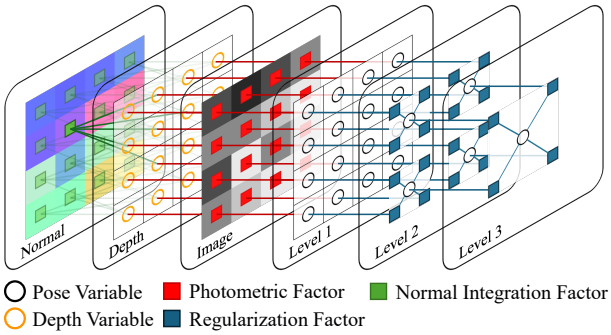


Figure 2. Topology of the proposed factor graph. The right side of the Image layer, related to camera pose estimation, follows a sharded topology [1] extended to $\mathbb{SE}(3)$. The left side reconstructs depth through normal-integration and photometric factors; for clarity, all normal-integration factors except the one at row 1, column 1 are rendered semi-transparent.

achieve consistent estimates of camera motion, a global property, across the distributed memory of a PPA. We argue that a *dense* VO approach, where camera motion is estimated alongside a full dense depth map, is actually easier to achieve on a PPA than a sparse approach, because the continuity of real scenes can be harnessed as local pixel-wise priors. We formulate our approach as a distributed factor graph stored across the whole pixel array, with variables at every pixel for camera motion and scene depth. Identity factors synchronize a global camera motion estimate across the array.

Related work. The closest works to ours explore whether global geometric estimation can be achieved fully in-pixel. BP-SF [23] investigates 6DoF visual odometry for RGB-D sensors with a pixel-parallel model implemented on a Graphcore IPU [15], while PixRO [1] studies in-pixel camera rotation tracking and compares gridwise and hierarchical communication structures. Both adopt factor graph and GBP formulations similar to ours; we tackle the full monocular 6DoF VO and depth estimation problem. Surface normal estimation is known to be achievable with local receptive fields [2], justifying the assumption that a per-frame normal prior is available from a PPA. Normal integration [7] recovers depth from a normal field, and we exploit its spatial sparsity for pixel-level parallelization. Probabilistic graphical models [3, 11] and GBP [18, 21] remain powerful inference tools in resource-constrained settings. Unlike rotation/translation averaging [9, 30] or hardware-accelerated SLAM [4, 12], our work pursues pixel-level rather than image-level parallelization.

2. Method

2.1. Direct Visual Tracking: Centralized Baseline

We consider a calibrated camera undergoing 6DoF motion in a static scene. Given a source image $\mathcal{I}_s$ and a target image $\mathcal{I}_t$, we seek the relative pose $\mu_{[0:6]} \in \mathbb{SE}(3)$ and a source-view log-depth $\mu_{[6]}$ jointly explaining their photometric differences. In high frame-rate tracking, we perform direct iterative photometric alignment over the overlap $\Omega(\mu)$:

$$\mu^* = \arg\min_{\mu} \sum_{p \in \Omega(\mu)} \rho(\|\mathcal{I}_s[p] - \mathcal{I}_t[\mathcal{W}(p;\mu)]\|)^2, \quad (1)$$

where ρ is a robust cost and $\mathcal{W}(p;\mu) = \pi\big(K \exp(\mu_{[0:6]}) K^{-1} \pi^{-1}(p, \exp(\mu_{[6]}))\big)$ is the standard warping [20, 23]. Optimizing Eq. (1) assumes globally accessible memory — the centralized setting we use as a later baseline.

2.2. Distributed Formulation with a Gaussian Factor Graph

We formulate dense VO and depth estimation as a Gaussian factor graph [11] describing the probabilistic relationship between variables $v \in \mathcal{V}$ and factors $f \in \mathcal{F}$, with $p(\mathcal{V}) = \prod_i f_i(\mathcal{V}_{f_i})$. Each pixel stores its local variables, and factor information is stored at the pixels at either end of the connection, so that marginal estimates can be obtained via Gaussian Belief Propagation (GBP) using only local computation and message passing. Figure 2 illustrates the factor graph topology for one timestep. At each pixel p_i we store a stacked variable $\mu_i \in \langle \mathbb{SE}(3), \mathbb{R} \rangle$, with $\mu_{i,[0:6]}$ a local estimate of global camera motion and $\mu_{i,[6]}$ a log-depth up to scale. Our model uses the following factors.

Photometric factors. Decomposing Eq. (1) into per-pixel terms:

$$E_D^i(\mu_i) = \tfrac{1}{2}\rho\big(\|\mathcal{I}_s[p_i] - \mathcal{I}_t[\mathcal{W}(p_i;\mu_i)]\|_{\Lambda_D}\big)^2. \quad (2)$$

Each pixel reads the photometric value at its warped location in the target image assuming its own pose estimate; on real PPAs, this uses local routing between neighboring pixels.

Prior factors. A per-pixel prior on camera pose and log-depth stabilizes early iterations and propagates motion continuity across timesteps: $E_P^i(\mu_i) = \frac{1}{2}\|\mu_i \ominus \hat{\mu}_i\|^2_{\Lambda_P}$.

Camera motion identity factors. These encourage the per-pixel motion estimates to agree on a single global value:

$$E_R^{i,j}(\mu_{i,[0:6]}, \mu_{j,[0:6]}) = \tfrac{1}{2}\|\mu_{i,[0:6]} \boxminus \mu_{j,[0:6]}\|^2_{\Lambda_R}. \quad (3)$$

These factors sync a global variable and can in principle take any connection pattern. We use a hierarchical sharded pattern matching PixRO [1], which achieves good global pose estimates with few GBP iterations.

Normal integration factors. To reconstruct smooth dense depth, we exploit a per-pixel surface-normal prior —

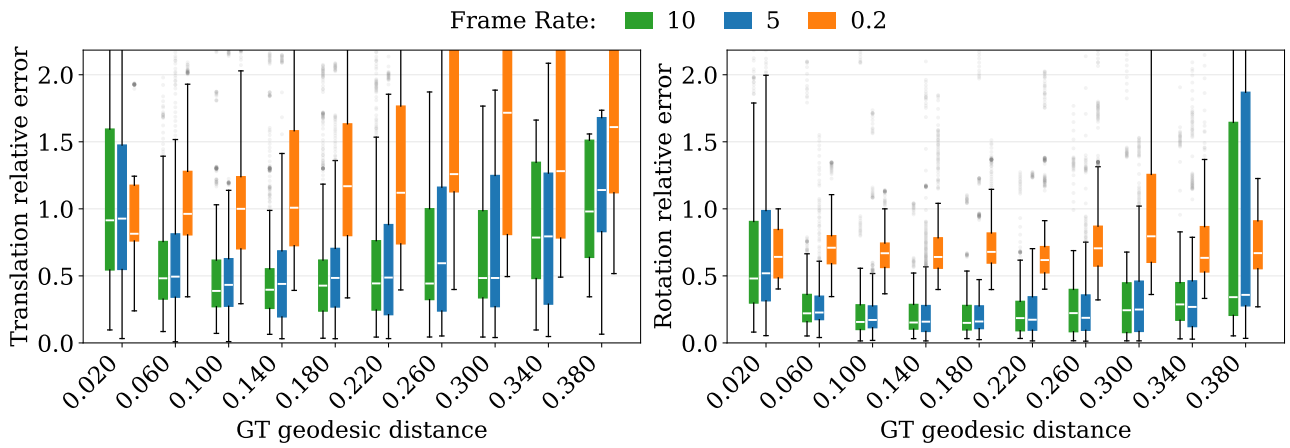


Figure 3. Estimated pose error across frame rates, as a function of the deviation of the target camera pose from the keyframe. As the frame rate decreases (larger inter-frame motion), the convergence basin shrinks and error grows. However, higher frame rates exhibit diminishing returns (cf. framerates **10** vs. **5**), consistent with observations in event-based vision [13].

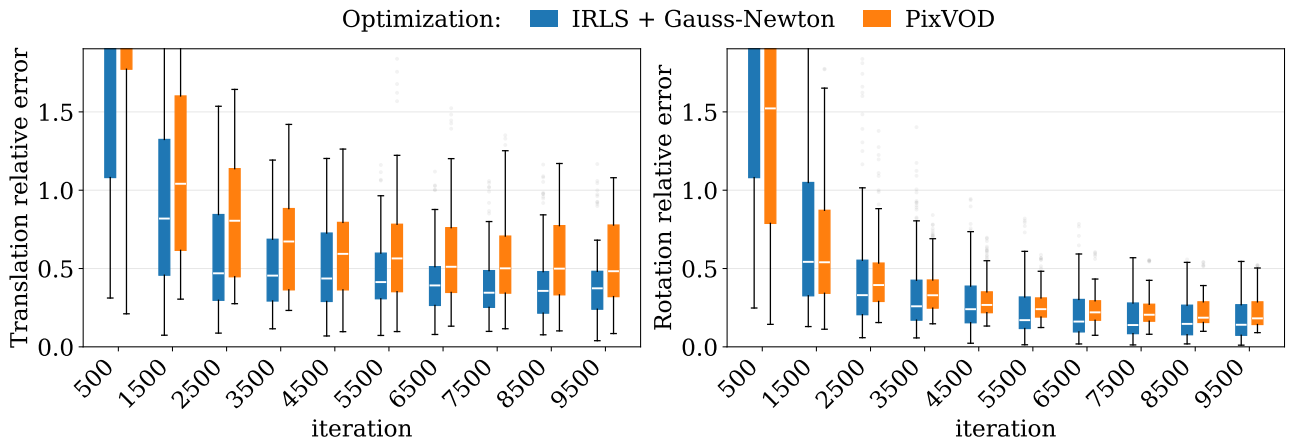


Figure 4. The proposed pixel-parallel method against the centralized baseline (IRLS with Gauss–Newton).

which is estimable from local information [2] — following BINI [7]:

$$E^{i>j}_{N_x} = \tfrac{1}{2}\left\|(\mu_{i,[6]}-\mu_{j,[6]})+\tfrac{n_x}{\tilde{n}_z}\right\|^2_{\Lambda_N},$$
$$E^{i>j}_{N_y} = \tfrac{1}{2}\left\|(\mu_{i,[6]}-\mu_{j,[6]})+\tfrac{n_y}{\tilde{n}_z}\right\|^2_{\Lambda_N}, \quad (4)$$

where $(n_x, n_y, \tilde{n}_z)$ is the local normal in image-plane coordinates [7].

2.3. Iterative Gaussian Belief Propagation

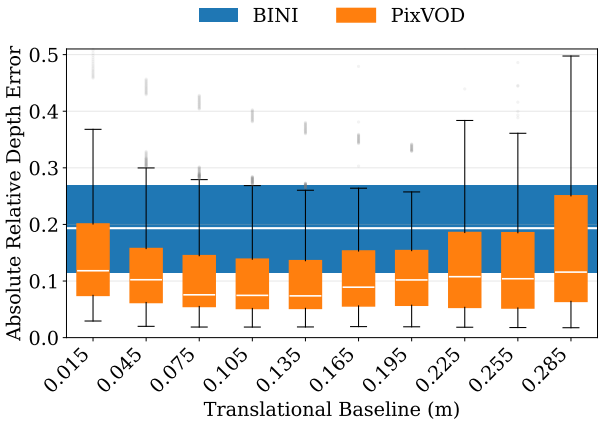
We use GBP to obtain marginals via fully distributed computation with analytic Gaussian message updates [1, 18, 19]. Since the variables are nonlinear, we linearize around the current mean $\bar{\mu}$ on the composite manifold, write $v = \bar{\mu} \boxplus {}^{\bar{\mu}}\xi$ with the Lie-algebra element ${}^{\bar{\mu}}\xi$ treated as Euclidean [25], and alternate message passing and belief updates following [10, 22], with a Huber robust loss and covariance weighting [10].

2.4. Keyframe-based Tracking Strategy

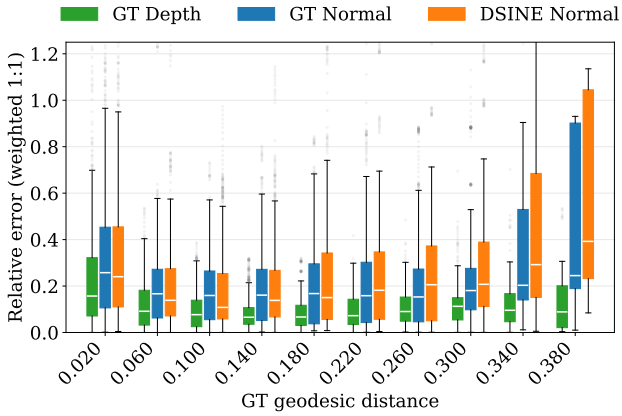
Direct photometric tracking has a limited basin of convergence, but the PPA setting permits very high frame rates and therefore small inter-frame motions. However, if that motion becomes excessively small, rotation and translation become hard to disambiguate. We therefore adopt a local keyframing strategy: the reference frame is kept as a fixed keyframe for a number of steps, while the target frame is updated incrementally, and 6DoF poses are always estimated relative to this keyframe (Sec. 3.1).

3. Experiments

All steps of the GBP algorithm are parallelized with JAX as a GPU simulation of a future on-pixel-array imple-



(a)



(b)

Figure 5. **(a)** Depth reconstruction of our method vs. converged BINI [7] at various translational baselines. **(b)** Camera-pose evaluation with various structural priors.

mentation, run on a desktop with an RTX 4090 GPU. We generate sequences of 128×128 images from the Replica dataset [26] (*office_{0,1,2,3}* from iMap [27]/NICE-SLAM [28]), upsampled by 10× using ScLERP [14] to simulate the small inter-frame baselines expected at PPA frame rates. Translation and depth are evaluated after aligning their scale with ground truth via L_2 minimization. Unless stated otherwise, we use GT surface normals as the local prior and set $(\sigma_D, \sigma_P, \sigma_R, \sigma_N, t_{\text{huber}}) = (5{\times}10^{-3}, 1, 4{\times}10^{-4}, 10^{-3}, 400)$ at the leaf level, halving σ_P and σ_R at each higher shard level of Fig. 2.

3.1. Analysis on Keyframe-based Tracking

Along each upsampled Replica trajectory we sample 10 keyframes per scene; for each, we extract GT normals as the prior for Eq. (4), and treat the subsequent 300 frames as sequentially-updated target frames for Eq. (2), running 100 synchronized GBP iterations per frame. Per-pixel pose estimates are averaged into a single estimate and propagated as the next frame's initialization.

We analyze why keyframe–frame optimization is essential by varying the target update frequency. Fig. 3 compares the original protocol (framerate 10) against updating every two frames with 2× iterations (framerate 5) and every fifty frames with 50× iterations (framerate 0.2). The x-axis is the geodesic distance of the GT relative pose from the keyframe; y-axis is relative translation/axis-angle rotation error. At framerate 0.2, the method settles at suboptimal solutions far more often — target updates cannot be too infrequent. Framerates 10 vs. 5 show diminishing returns, a limit that in real sensors is accentuated by reduced SNR from shorter exposures. Tracking is hardest at both extremes: small motion makes translation and rotation hard to disambiguate, and large motion shrinks the usable visual overlap.

3.2. Pixel-Parallel vs. Centralized Optimization

We compare our GBP-based method against the centralized baseline of Sec. 2.1, solved as IRLS with Gauss–Newton. As shown in Fig. 4, the centralized method converges faster and more accurately, as expected — but only under per-iteration access to all pixels. Our method accesses pixels locally, offering an alternative where global access is prohibitively expensive.

3.3. Photometrically Guided Normal Integration and Prior Ablation

We compare our method against pure normal integration without photometric guidance [16, 29]. When initialized with BINI's converged depth, our method achieves substantially higher depth quality (Fig. 5a): photometric correspondences extend normal-integration reconstruction beyond the object level by pulling apart the two sides of depth-discontinuous boundaries, with a translation sweet-spot analogous to Sec. 3.1. We also evaluate PixVOD with various structural priors (Fig. 5b): GT depth is best, as expected, while DSINE [2] normals give quality close to GT normals aside from a slightly reduced basin — showing that our method generalizes to practical normal estimators.

4. Conclusion

We present the first per-pixel distributable algorithm for jointly optimizing camera pose and depth, in which each pixel-processor infers global image properties using only its own measurement, a local geometric prior, and messages exchanged with neighbors. Our keyframe-based tracking strategy is effective in the typical PPA regime of high frame rates with small inter-frame changes, and the analysis of its operating range offers insights for the design of next-generation pixel-parallel vision chips.

References

[1] Ignacio Alzugaray, Riku Murai, and Andrew Davison. Pixro: Pixel-distributed rotational odometry with gaussian belief propagation. *arXiv preprint arXiv:2406.09726*, 2024. 2, 3

[2] Gwangbin Bae and Andrew J Davison. Rethinking inductive biases for surface normal estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9535–9545, 2024. 2, 3, 4

[3] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*. Springer, 2006. 2

[4] Konstantinos Boikos and Christos-Savvas Bouganis. Semi-dense slam on an fpga soc. In *2016 26th International Conference on Field Programmable Logic and Applications (FPL)*, pages 1–4. IEEE, 2016. 2

[5] Laurie Bose, Piotr Dudek, Stephen J Carey, and Jianing Chen. Live demonstration: Scamp-7. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3995–3996, 2023. 1

[6] Laurie Bose, Jianing Chen, and Piotr Dudek. Descriptor-in-pixel: Point-feature tracking for pixel processor arrays. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5392–5400, 2025. 1

[7] Xu Cao, Hiroaki Santo, Boxin Shi, Fumio Okura, and Yasuyuki Matsushita. Bilateral normal integration. In *European Conference on Computer Vision*, pages 552–567. Springer, 2022. 2, 3, 4

[8] Stephen J Carey, Alexey Lopich, David RW Barr, Bin Wang, and Piotr Dudek. A 100,000 fps vision sensor with embedded 535gops/w 256× 256 simd processor array. In *2013 symposium on VLSI circuits*, pages C182–C183. IEEE, 2013. 1

[9] Avishek Chatterjee and Venu Madhav Govindu. Efficient and robust large-scale rotation averaging. In *Proceedings of the IEEE international conference on computer vision*, pages 521–528, 2013. 2

[10] Andrew J Davison and Joseph Ortiz. Futuremapping 2: Gaussian belief propagation for spatial ai. *arXiv preprint arXiv:1910.14139*, 2019. 3

[11] Frank Dellaert, Michael Kaess, et al. Factor graphs for robot perception. *Foundations and Trends® in Robotics*, 6(1-2): 1–139, 2017. 2

[12] Weikang Fang, Yanjun Zhang, Bo Yu, and Shaoshan Liu. Fpga-based orb feature extraction for real-time visual slam. In *2017 International Conference on Field Programmable Technology (ICFPT)*, pages 275–278. IEEE, 2017. 2

[13] Ankur Handa, Richard A Newcombe, Adrien Angeli, and Andrew J Davison. Real-time camera tracking: When is high frame-rate best? In *European Conference on Computer Vision*, pages 222–235. Springer, 2012. 3

[14] Ladislav Kavan, Steven Collins, Carol O'Sullivan, and Jiri Zara. Dual quaternions for rigid transformation blending. *Trinity College Dublin*, 5:4, 2006. 4

[15] Graphcore Ltd. IPU Processors. https://www.graphcore.ai/products/ipu. 2

[16] Kirill Mazur, Gwangbin Bae, and Andrew J Davison. Superprimitive: Scene reconstruction at a primitive level. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4979–4989, 2024. 4

[17] Riku Murai, Sajad Saeedi, and Paul HJ Kelly. Bit-vo: Visual odometry at 300 fps using binary features from the focal plane. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8579–8586. IEEE, 2020. 1

[18] Riku Murai, Joseph Ortiz, Sajad Saeedi, Paul HJ Kelly, and Andrew J Davison. A robot web for distributed many-device localization. *IEEE Transactions on Robotics*, 40:121–138, 2023. 2, 3

[19] Riku Murai, Ignacio Alzugaray, Paul HJ Kelly, and Andrew J Davison. Distributed simultaneous localisation and auto-calibration using gaussian belief propagation. *IEEE Robotics and Automation Letters*, 9(3):2136–2143, 2024. 3

[20] Richard A Newcombe, Steven J Lovegrove, and Andrew J Davison. Dtam: Dense tracking and mapping in real-time. In *2011 international conference on computer vision*, pages 2320–2327. IEEE, 2011. 2

[21] Joseph Ortiz, Mark Pupilli, Stefan Leutenegger, and Andrew J Davison. Bundle adjustment on a graph processor. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2416–2425, 2020. 2

[22] Joseph Ortiz, Talfan Evans, and Andrew J Davison. A visual introduction to gaussian belief propagation. *arXiv preprint arXiv:2107.02308*, 2021. 3

[23] Raluca Scona, Hidenobu Matsuki, and Andrew Davison. From scene flow to visual odometry through local and global regularisation in markov random fields. *IEEE Robotics and Automation Letters*, 7(2):4299–4306, 2022. 2

[24] Haley M So, Laurie Bose, Piotr Dudek, and Gordon Wetzstein. Pixelrnn: in-pixel recurrent neural networks for end-to-end-optimized perception with neural sensors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 25233–25244, 2024. 1

[25] Joan Sola, Jeremie Deray, and Dinesh Atchuthan. A micro lie theory for state estimation in robotics. *arXiv preprint arXiv:1812.01537*, 2018. 3

[26] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, et al. The replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019. 4

[27] Edgar Sucar, Shikun Liu, Joseph Ortiz, and Andrew J Davison. imap: Implicit mapping and positioning in real-time. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6229–6238, 2021. 4

[28] Zihan Zhu, Songyou Peng, Viktor Larsson, Weiwei Xu, Hujun Bao, Zhaopeng Cui, Martin R Oswald, and Marc Pollefeys. Nice-slam: Neural implicit scalable encoding for slam. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12786–12796, 2022. 4

[29] Zihan Zhu, Songyou Peng, Viktor Larsson, Zhaopeng Cui, Martin R Oswald, Andreas Geiger, and Marc Pollefeys. Nicer-slam: Neural implicit scene encoding for rgb slam. In *2024 International Conference on 3D Vision (3DV)*, pages 42–52. IEEE, 2024. 4

[30] Bingbing Zhuang, Loong-Fah Cheong, and Gim Hee Lee. Baseline desensitizing in translation averaging. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4539–4547, 2018. 2