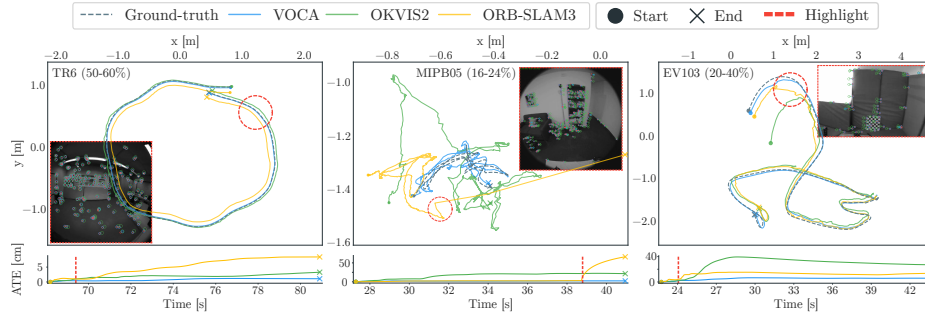


# VOCA: Visual Odometry with Codec Awareness

Nouri Alexander Hilscher<sup>\*1</sup>  Mateo de Mayo<sup>\*1,2</sup>  Dominik Muhle<sup>1,2</sup>   
 Christoph Otten genannt Hermes<sup>1</sup> Daniel Cremers<sup>1,2</sup> 

<sup>1</sup> Technical University of Munich, Munich, Germany

<sup>2</sup> Munich Center for Machine Learning, Munich, Germany



**Fig. 1: Visual Odometry on Compressed Videos.** We present VOCA, a novel Visual Odometry system that produces smoother, more stable trajectories than descriptor-based systems such as ORB-SLAM3 and OKVIS2, thanks to its codec-aware sparse optical-flow frontend. Our system enables Visual Odometry on data compressed by up to  $100\times$ . We visualize challenging segments with dashed-red markers, sampled from three different datasets. In the zoomed-in views, we show the **previous pixel** location that would be the prior in regular VO systems, the **motion-vector** prior that we introduce, and the optical flow **solution**. In most cases, motion-vector priors reduce the initialization distance to the solution to just a few pixels.

**Abstract.** Camera pose estimation from image streams is a critical component of spatial world models that integrate perception into planning and decision-making. Nearly all Visual Odometry (VO) and Simultaneous Localization and Mapping (SLAM) systems have focused on datasets containing raw, uncompressed videos. Many working systems instead use ubiquitous hardware units to efficiently compress and decode video streams, saving orders of magnitude in storage and bandwidth. However, this lossy compression introduces visual artifacts that hinder the performance of traditional tracking systems. We present VOCA, a causal stereo visual-odometry method that exploits codec information to improve tracking performance. We achieve state-of-the-art performance on causal VO for relative trajectory error, efficiency, and absolute trajectory error on compressed streams. This work highlights the potential of leveraging widely available video codec information for vision tasks.

**Keywords:** Visual Odometry · KLT-Tracking · Compressed Videos

<sup>\*</sup> equal contribution

## 1 Introduction

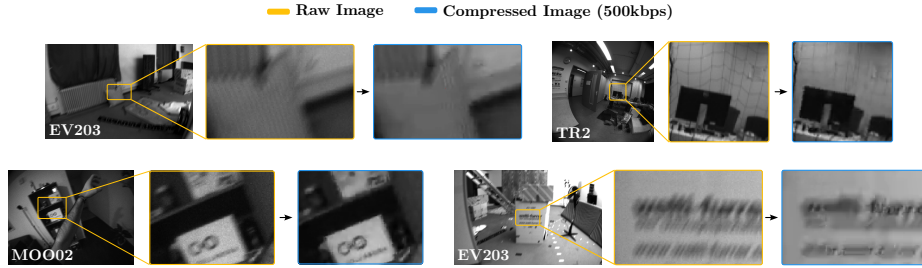
Spatial computing systems make their way into our lives in a variety of forms, from mixed reality devices to autonomous systems such as robotic assistants, drones, and self-driving cars. Cameras have proven to be a rich and cost-effective sensor, driven in part by the ubiquity of mobile and embedded vision in consumer devices [3, 63]. Current trends in computer vision, learning, and robotics show increasing overlap in research, with no sign of slowing [9, 27, 47]. Furthermore, immense potential is expected from learning-based models that leverage the vast amount of internet videos captured by these sensors for training [2, 9]. In this context, tracking camera poses has become imperative for many applications and research [11, 19, 21, 23, 70]. There cannot be action and planning in 3D without first understanding the current location of a moving agent [22].

The high potential for cameras comes at the cost of large volumes of data. A raw image stream from a common setup of stereo monochrome cameras (8-bit pixels) with a resolution of  $640 \times 480$  at 30 frames per second produces more than a gigabyte of data per minute. Since bandwidth and memory are expensive [72], practical systems rely on compression to transmit camera streams efficiently under the compress-then-analyze design paradigm [24]. Years of software and hardware development have gone into accelerating common encoders such as H.264, AV1, and VP9 [20, 42, 68]. Video codecs are now ubiquitous when working with camera streams [46].

Over the last 20 years, a variety of approaches, ranging from purely classical algorithms [7, 16, 18, 32, 43, 44, 50, 53, 64] to various learning-based systems [29, 34, 41, 57, 66, 67, 69] have been proposed in the literature for egocentric tracking. However, most VO and Visual SLAM (VSLAM) systems focus solely on raw images, with datasets and benchmarks in the field considering only fully uncompressed images [3, 39]. As we will show, common video encoders like H.264 can reduce the size of sequences by orders of magnitude. Unfortunately, this compression process is lossy and introduces visual artifacts that significantly degrade the performance of these state-of-the-art tracking systems [62] and other vision tasks [48]. This has been an often overlooked challenge in the field of VO and VSLAM, with only a few works discussing the topic [24, 73] and, to the best of our knowledge, only one, MoV-SLAM [62], attempting to actively improve tracking through codec information.

In this work, we show that a significant amount of the tracking performance loss due to compression can be recovered by properly exploiting the information available in the encoded data. In some cases, certain appearance changes introduced by compression can make feature tracking more stable (e.g., lower pixel noise since it is difficult to encode). In addition, the reduced bandwidth and memory requirements of compressed streams can improve runtime performance, since decoding can, in many cases, be more efficient than repeatedly accessing the much larger raw image sequences.

Following these observations, we present VOCA, a *causal* stereo VO method that leverages video codec information for tracking. VOCA is built on top of the popular odometry system Basalt [64] with modifications from [39]. It produces



**Fig. 2:** Video encoding can introduce artifacts that violate the photometric-constancy assumption used by most tracking algorithms. Examples from datasets used in this work: EV203 (from EuRoC [6]) shows blurred details, TR2 (from TUM-VI room [54]) exhibits reduced contrast in the thin net, and MOO02 (from MSD [39]) contains geometrically jagged edges/textures. See [35] for additional artifacts.

causal estimates, meaning it uses only past information for tracking, making it suitable for real-time use [32]. Our method outperforms Basalt on  $\sim 100\times$  compressed sequences. Furthermore, VOCA achieves state-of-the-art performance for causal stereo VO when compared to other top-scoring systems (see Fig. 1). The contributions of our work include:

- VOCA, a causal stereo VO method that can run on compressed videos and leverage its codec information for greater accuracy and efficiency.
- An extensive benchmark showing that VOCA achieves state-of-the-art performance, even in settings with heavy compression ( $\sim 100\times$ ).
- An ablation study that isolates the contribution of each component and provides insights for future methods for tracking encoded videos.
- A performance degradation analysis, showing tracking degradation for top-scoring systems under different compression settings

See VOCA’s project page at <https://tum-vision.github.io/voca>.

## 2 Related Work

The focus of this paper is VO for compressed images. Our discussion of related work will include an overview of the topics *VO* and *VSLAM*, with a focus on the most closely related work MoV-SLAM [62]. We include approaches that also use inertial measurements, although our work is primarily concerned with improvements in front-end tracking. Therefore, we will introduce feature tracking separately to discuss its challenges under video compression. We will also give a brief introduction to video compression.

### 2.1 Feature Tracking

Feature-based VO and SLAM approaches require tracking feature positions across multiple images to construct a (windowed) bundle adjustment optimization prob-

lem. To establish these correspondences, feature positions are extracted and descriptors of these feature positions are matched between images.

For classical VO, popular descriptors are SIFT/SURF [4, 37], ORB [49], and BRISK [33]. SuperPoint [15] presents a learned alternative. For compressed videos, MoV-SLAM [62] introduced EXPRESS features based on the Macro Blocks (MBs) of the H.264 video encoder. Each MB represents a potential feature, classified as *distinctive* or not, based on a linear-time operation along the patch diagonals. The features are scale-invariant but not rotation-invariant. They are matched using a binary descriptor distance, similar to ORB features.

Tracking feature points presents an alternative to feature matching, usually based on optical flow, such as KLT-Tracking [38, 56, 61]. The reliance on pixel intensities renders optical-flow-based approaches sensitive to image artifacts introduced by compression, as the photo constancy assumption is violated. In this work, we exploit information from video compression to reduce the impact of these artifacts on the feature tracking, improving its robustness.

## 2.2 Visual Odometry

Classical VO approaches can be broadly categorized into feature-based and direct methods. Whereas feature-based methods such as PTAM [30], MRO [12], ROBA [31], PNEC [40], and Basalt [64] use the aforementioned feature positions, direct methods such as DSO [16] and DM-VIO [65] operate directly on pixel intensities. Because they rely on the photo constancy assumption, direct methods are even more sensitive to image artifacts than sparse feature-tracking methods. More recently, purely learning-based approaches have enabled VO on uncalibrated videos including VGGT [66], PI3 [67], MapAnything [29], AnyCam [69], AnyMap [13], and DepthAnything3 [34]. Due to their large model size and scalability constraints for long videos, they are not yet viable for deployment on constrained hardware. Hybrid approaches like LEAP-VO [10] and Splat-SLAM [51] bridge the gap somewhat. We build our work on Basalt [64] as it is among the few VO systems that use KLT-Tracking.

## 2.3 Visual Simultaneous Localization and Mapping

Despite its rigorous mathematical definition in [58], we follow the commonly accepted approach in which SLAM extends VO with concepts such as loop closure [8]. Similar to VO, a family of direct SLAM methods exists around LSD-SLAM [17]. ORB-SLAM 1, 2, and 3 [7, 43, 44] build a system around matching ORB features and loop detection using Bag-of-Words. OKVIS2 [32] employs BRISK features and is optimized for drone usage. Snake-SLAM [50] decouples front-end processing and back-end optimization, which is further split into IMU state estimation and bundle adjustment to improve efficiency. Similar to VO, end-to-end learning approaches [14, 45, 60] have been introduced in recent years. More closely related to our approach is MoV-SLAM [62]. MoV-SLAM leverages the EXPRESS features discussed previously for front-end correspondence estimation with optional stereo matching using Lucas-Kanade optical flow. The

back-end is based on PnP pose estimation using MAGSAC++. Despite its name, MoV-SLAM does not use loop closure for drift correction.

## 2.4 Video Compression

Early digital video coding was first standardized for low-bitrate transmission (e.g., ITU-T H.261 [36]), and later for broadcast and storage (MPEG-2 [26]), which combined block transforms with motion-compensated inter-frame prediction [52]. With the rise of videos on the internet, H.264/AVC [68] became the most widely adopted codec by major platforms [25, 46, 55]. Modern video codecs (e.g., MPEG-2/H.262, H.264/AVC, H.265/HEVC, VP9, AV1, and H.266/VVC [5, 20, 26, 42, 59, 68]) largely follow a hybrid block-based architecture: Each frame is partitioned into blocks, and for each block the codec forms a predictor via intra- or inter-prediction. The former leverages previously reconstructed neighboring blocks within the same frame, while the latter uses motion-compensated prediction from one or more previously decoded reference frames, either causally or not, depending on encoder settings. The block is then transformed, quantized, and entropy-coded, typically with in-loop filtering to improve reference quality.

For inter-frame prediction, the encoder selects one (or more) previously reconstructed reference frames and estimates per-block motion vectors that point to displaced regions in the reference; the decoder uses the corresponding displaced (and, if needed, sub-pixel interpolated) reference block as an initial prediction and reconstructs the final block by adding the decoded (inverse-quantized/inverse-transformed) residual. Although these per-block motion vectors are primarily optimized for rate-distortion efficiency in inter-frame coding, prior work shows that they roughly resemble optical flow (see Fig. 3) and can be reused as a warm-start/initialization for optical flow estimation pipelines [28, 71, 73, 74].

Despite newer codecs improving compression efficiency, H.264/AVC remains the most widely adopted codec on commodity devices [46]. Moreover, recent fixed-function encoder benchmarks show that H.264 encoding throughput often matches or even exceeds that of newer codecs [1], owing to its mature hardware-accelerated implementations. Accordingly, we focus on H.264/AVC as our primary codec throughout the paper and refer the reader to the supplementary material for preliminary positive results of the method on the AV1 codec.

## 3 Method

We first explain why video compression degrades KLT-Tracking, and hence why systems like Basalt [64] struggle under these conditions, then show how the compression itself can initialize tracking to overcome these limitations. Throughout, we assume a causal configuration in which encoding order matches capture order and inter-frame prediction references only past frames. This matches both our target streaming setting and all experiments (Sec. 4). We begin with the basics of video compression and optical flow.

### 3.1 A Primer on Video Compression

Let  $\mathbf{I}_t : \Omega \rightarrow \mathbb{R}^C$  be the image at time  $t$  defined on the pixel lattice  $\Omega = \{1, \dots, H\} \times \{1, \dots, W\}$  with  $C$  color channels. In video coding,  $\mathbf{I}_t$  is typically represented in a color space that separates the luminance, like YCbCr, and the encoding pipeline operates on each component separately. For notational simplicity and because the evaluated datasets are monochromatic, we describe the process for a single scalar component  $Y_t : \Omega \rightarrow \mathbb{R}$ .

Let  $\{B_k\}_{k=1}^K$  be a frame-specific partition of  $\Omega$  into  $K$  rectangular blocks, indexed by  $k$  alone since  $t$  is fixed throughout. Each block  $B_k = \{m_k, \dots, m_k + H_k - 1\} \times \{n_k, \dots, n_k + W_k - 1\} \subseteq \Omega$  has height  $H_k$ , width  $W_k$ , and a top-left anchor coordinate  $(m_k, n_k)$ . We define the block of component values associated with  $B_k$  as  $\mathbf{y}_{t,k} = Y_t(B_k) \in \mathbb{R}^{H_k \times W_k}$ .

The encoder aims to represent each block  $\mathbf{y}_{t,k}$  as accurately as possible while minimizing the number of required bits. Within a frame, blocks are processed in encoding order; when processing  $B_k$ , previously reconstructed blocks  $\hat{\mathbf{y}}_{t,1}, \dots, \hat{\mathbf{y}}_{t,k-1}$  within the current frame and reconstructed frames  $\hat{Y}_{t'}, t' < t$ , are available. The encoder forms a predicted block  $\tilde{\mathbf{y}}_{t,k}$  either by *intra-frame* prediction from neighboring boundary samples or by *inter-frame* prediction, in which a reference frame  $\hat{Y}_{t'}$  and a displacement vector, often called a *motion vector*,  $\mathbf{d}_k \in \mathbb{R}^2$ , likewise indexed only by  $k$ , are selected by minimizing a rate-distortion cost, yielding  $\tilde{\mathbf{y}}_{t,k} = \hat{Y}_{t'}(B_k + \mathbf{d}_k)$ . The prediction residual is then computed, transformed, and quantized as

$$\mathbf{r}_{t,k} = \mathbf{y}_{t,k} - \tilde{\mathbf{y}}_{t,k}, \quad \mathbf{X}_{t,k} = \mathcal{Q}(\mathcal{T}(\mathbf{r}_{t,k})). \quad (1)$$

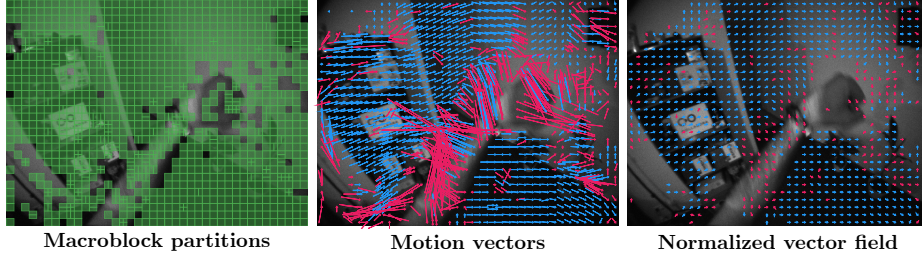
Since quantization is a lossy, many-to-one mapping, the decoder can only recover an approximation  $\hat{\mathbf{r}}_{t,k} = \mathcal{T}^{-1}(\mathcal{Q}^{-1}(\mathbf{X}_{t,k})) \neq \mathbf{r}_{t,k}$ , which leads to artifacts as shown in Fig. 2. The quantized coefficients  $\mathbf{X}_{t,k}$ , together with the prediction parameters (prediction mode, motion vector, and reference index), are entropy coded into the bitstream. The choice of admissible prediction types is governed at the frame level: within so-called *I-frames*, all blocks are restricted to intra-prediction, making the frame independently decodable and serving as a random-access point; within so-called *P-frames*, each block may choose between intra and inter prediction, and blocks predicted in intra mode do not carry motion vectors. The large majority of frames, however, are encoded as P-frames.

In H.264 [68], the blocks  $B_k$  correspond to so-called *macroblocks* of fixed size  $16 \times 16$  or (sub)-partitions of macroblocks with sizes ranging from  $16 \times 8$ ,  $8 \times 8$ ,  $4 \times 8$ ,  $8 \times 4$ , down to  $4 \times 4$ , each carrying its own motion vector (see Fig. 3).

### 3.2 From Video Compression to Lucas-Kanade Optical Flow

We denote with  $I_t : \Omega \rightarrow [0, 1]$  the gray-scale intensity of the image frame  $\mathbf{I}_t$ . The Lucas-Kanade [38] method assumes *brightness constancy* along the trajectory  $\mathbf{p}(t)$  of a moving point in an image sequence  $I(\mathbf{p}, t)$ ,

$$I(\mathbf{p}(t), t) = \text{const}, \quad \forall t, \quad (2)$$



**Fig. 3: Encoded information.** Video codecs assign motion vectors to macroblock partitions and sub-partitions. They encode local block motion as a displacement to an intensity-matched block in a reference frame. The figure shows a frame with macroblock partitions (left), its motion vectors (center), and the normalized discrete vector field they induce (right). While this field is correlated with optical flow, it is considerably noisier: the encoder is free to match any region of the reference frame to improve compression, regardless of whether the resulting vector reflects true scene motion. Vectors likely deviating from optical flow are highlighted in red via median thresholding.

which, via first-order linearization, yields the optical-flow constraint

$$\frac{d}{dt}I(\mathbf{p}(t), t) = \nabla I(\mathbf{p}, t)^\top \mathbf{v}(\mathbf{p}, t) + \frac{\partial}{\partial t}I(\mathbf{p}, t) = 0, \quad (3)$$

where  $\mathbf{v} = (u, v)^\top$  denotes the (instantaneous) optical flow and  $I(\mathbf{p}, t) \equiv I_t(\mathbf{p})$ .

Evaluated between two consecutive frames (i.e., discretizing with  $\Delta t = 1$ ), the brightness constancy assumption takes the form

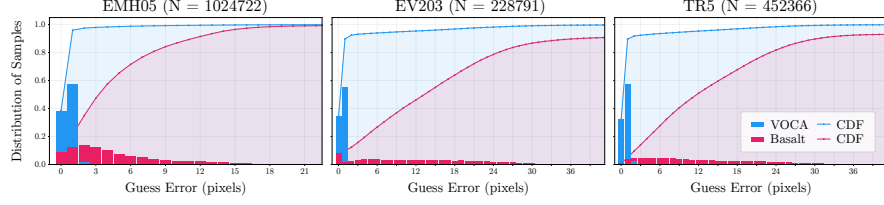
$$I_{t-1}(\tilde{\mathbf{p}}) \approx I_t(\tilde{\mathbf{p}} + \mathbf{v}). \quad (4)$$

Lucas-Kanade further assumes locally constant motion, i.e.,  $\mathbf{v}(\tilde{\mathbf{p}}, t) \approx \mathbf{v}(\mathbf{p}, t)$  for all  $\tilde{\mathbf{p}}$  in a neighborhood  $\mathcal{N}(\mathbf{p})$ .

This local-constancy assumption mirrors the structure of block-based inter-frame prediction in video coding: identifying  $\mathcal{N}(\mathbf{p})$  with the block  $B_k$  containing  $\mathbf{p}$ , the encoder assigns a motion vector  $\mathbf{d}_k$  that predicts  $B_k$  by sampling a displaced region from a reference frame, yielding the relation

$$I_t(\tilde{\mathbf{p}}) \approx I_{t-1}(\tilde{\mathbf{p}} + \mathbf{d}_k), \quad \forall \tilde{\mathbf{p}} \in B_k. \quad (5)$$

Comparing with Eq. (4), since the motion vector  $\mathbf{d}_k$  points from the current frame back into the reference frame, whereas  $\mathbf{v}$  points forward in time,  $-\mathbf{d}_k$  acts as a coarse, block-wise approximation of sparse optical flow, i.e.,  $-\mathbf{d}_k \approx \mathbf{v}$ . However, as Fig. 3 illustrates, this correspondence can break down: since the encoder is free to reference any region that improves compression, regardless of whether the resulting vector reflects true scene motion, decoded motion vectors may deviate substantially from optical flow, particularly in areas with little texture, repetitive patterns, or strong motion blur. We therefore maintain a distinction between the *true optical flow*  $\mathbf{v}$  and the decoded *motion vectors*  $\mathbf{d}_k$  throughout.



**Fig. 4: Proximity to the minimum.** VOCA uses motion vectors as priors for optical flow, which significantly reduces the distance to the ground-truth solution, possibly improving convergence times and the likelihood that the initial state lies in the convergence basin of the non-linear problem. This figure shows the error distributions of pixel distances between the prior guesses (initialization) and the final tracked results.

### 3.3 Using Motion Vectors as Priors for KLT-Tracking

While video-codec motion vectors provide purely translational displacements per block, KLT-Tracking estimates correspondences more accurately by optimizing over a richer motion model for image patches. Specifically, the Basalt [64] implementation performs tracking on the gray-scale intensity  $I_t$  and optimizes an SE(2) transform  $\mathbf{T}_{2D}$  per patch  $\mathcal{P} \equiv \mathcal{N}(\mathbf{p})$  centered on a keypoint  $\mathbf{p}$

$$\mathbf{T}_{2D} = \begin{pmatrix} \mathbf{R}_{2D} & \mathbf{t}_{2D} \\ \mathbf{0} & 1 \end{pmatrix}, \quad (6)$$

by minimizing the photometric error

$$\mathbf{R}_{2D}, \mathbf{t}_{2D} = \arg \min_{\mathbf{R}_{2D}, \mathbf{t}_{2D}} \sum_{\tilde{\mathbf{p}} \in \mathcal{P}} \left( \frac{I_{t-1}(\tilde{\mathbf{p}})}{\bar{I}_{t-1}} - \frac{I_t(\mathbf{R}_{2D}\tilde{\mathbf{p}} + \mathbf{t}_{2D})}{\bar{I}_t} \right)^2, \quad (7)$$

where  $\bar{I}_{t-1}$  and  $\bar{I}_t$  are mean patch intensities that normalize for illumination change. The translation  $\mathbf{t}_{2D}$  corresponds to an estimate of the optical flow  $\mathbf{v}$ , while the rotational component  $\mathbf{R}_{2D}$  captures local image rotation that a purely translational model, such as a codec motion vector, cannot represent.

Directly replacing KLT-Tracking with decoded motion vectors is not a viable alternative: as discussed above, motion vectors are optimized for compression rather than motion fidelity, making them noisy estimates of true scene motion. Moreover, they are purely translational and cannot represent the in-plane rotation captured by the  $\mathbf{R}_{2D}$  component of the SE(2) model.

However, motion vectors can bridge displacements far larger than what image pyramids alone can recover, making them well suited for use as initialization priors. Since  $\mathbf{t}_{2D}$  points forward in time (matching the convention of  $\mathbf{v}$ ) and  $\mathbf{v} \approx -\mathbf{d}_k$ , we initialize  $\mathbf{t}_{2D}$  with the negated motion vector  $-\mathbf{d}_k$ , while retaining the identity for  $\mathbf{R}_{2D}$ , and let the KLT optimizer refine both from this starting point. For backward tracking from frame  $t$  to  $t-1$ , used for cycle-consistency verification, the initialization is  $\mathbf{d}_k$  directly.

As Fig. 4 shows, for successful tracks, this initialization reduces the distances from the initialization points to the final converged positions when using



motion vectors; the majority of motion vectors lie within 1–2 pixels of the converged solution, confirming their effectiveness as initialization priors. Note that video compression additionally introduces artifacts into the decoded images (see Fig. 2), adding further noise to the optimization, thereby reinforcing the need for a full KLT refinement rather than relying on motion vectors directly.

### 3.4 Bridging I-Frames

As established in Sec. 3.1, I-frames carry no inter-frame prediction and therefore yield no motion vectors. This is particularly challenging because I-frames tend to occur precisely when scene content changes substantially, the same conditions under which establishing feature correspondences is most difficult. To maintain motion-vector initialization across I-frame boundaries, we reuse the motion vectors  $\mathbf{d}_k$  decoded from the last available P-frame, assuming constant motion. While these vectors do not reflect the true displacement at the I-frame, they provide a temporally proximate estimate that, in many empirical cases, lies sufficiently close to the true optical flow to place the KLT optimizer within its convergence basin.

### 3.5 Outliers through False Positives and Dynamic Objects

As the motion-vector initialization can cover large displacements, the tracking is not constrained to the same locality as the previous-pixel prior baseline. This has the potential to introduce two kinds of false positive tracks:

- A wrong motion vector can point in the wrong direction, but represent visually similar image regions (e.g., wall area of Fig. 3)
- A correct motion vector is associated with a dynamic object (see Fig. 3 on the arm). These features are difficult to track because of their large displacement.

If, in both of these scenarios, the forward tracking is already close to the minima, then the backward track will be initialized close to its origin. As the motion vectors from the video encoder represent photometrically similar regions, the traditional forward-backward consistency of KLT-Tracking is not reliable in filtering out these outliers. These correspondences badly influence the bundle adjustment in the backend and cause drift or divergence in the trajectory.

To reduce their influence, we propose the following strategy: track each point *with* and *without* motion priors. If a point is tracked only once, we treat it as a success. If both approaches successfully track the point, we check whether they are consistent and reject any that are inconsistent. This allows us to identify possible outliers in the tracks.

### 3.6 Implementation Details

As discussed in Sec. 1, KLT-Tracking has strong assumptions and is not well suited for compressed data. In consequence, this also holds for our foundation,

**Table 1: EuRoC dataset.** Following [39], we use thresholds for ATE and RTE ( $\Delta = 6$  frames) of 10 m and 10 cm, respectively, and mark runs that exceed these thresholds as divergent with  $\infty$ . Incomplete runs (crashes) are marked as  $\times$ . Systems that are unable to estimate more than 50% of the trajectory without resetting are marked as R. We show median (MED) and average (AVG). For avoiding outliers skewing the AVG, we exclude sequences that fail in any of the systems with reasonable metrics. We mark these sequences in **gray**. MoV-SLAM does not produce reasonable ATE and RTE, and ORB-SLAM3 does not produce reasonable RTE. Only on easy sequences does Basalt show slightly better performance than VOCA. On EuRoC, our method improves the mean RTE by  $\approx 15\%$  and the mean ATE by  $\approx 10\%$ , while being more robust, as shown by the number of successfully completed sequences.

|       | ATE [cm] (SE3 aligned) |              |              |              |              | RTE [cm] ( $\Delta = 6$ frames) |           |              |              |              |
|-------|------------------------|--------------|--------------|--------------|--------------|---------------------------------|-----------|--------------|--------------|--------------|
|       | MoV-SLAM               | ORB-SLAM3    | OKVIS2       | Basalt       | VOCA OURS    | MoV-SLAM                        | ORB-SLAM3 | OKVIS2       | Basalt       | VOCA OURS    |
| EMH01 | $\infty$               | <b>3.80</b>  | <u>12.10</u> | 16.20        | 17.50        | $\infty$                        | $\infty$  | 1.371        | <b>0.595</b> | 0.637        |
| EMH02 | $\infty$               | <b>5.30</b>  | <u>13.60</u> | <u>10.60</u> | 13.30        | $\infty$                        | $\infty$  | 1.528        | <u>0.621</u> | <b>0.604</b> |
| EMH03 | 373.90                 | <b>4.70</b>  | <u>17.60</u> | <u>20.00</u> | 20.60        | $\infty$                        | $\infty$  | 2.950        | <u>1.504</u> | <b>1.487</b> |
| EMH04 | 698.50                 | 949.30       | $\infty$     | <u>21.90</u> | <u>22.90</u> | $\infty$                        | $\infty$  | $\infty$     | <u>2.095</u> | 2.075        |
| EMH05 | $\times$               | <b>12.30</b> | 20.50        | <u>24.30</u> | <u>20.10</u> | $\times$                        | $\infty$  | 4.041        | <b>1.603</b> | <u>1.656</u> |
| EV101 | 681.90                 | <b>9.40</b>  | 15.60        | <u>10.10</u> | 11.30        | $\infty$                        | $\infty$  | 1.899        | <u>1.423</u> | <b>1.405</b> |
| EV102 | 484.40                 | R            | <u>20.60</u> | $\times$     | 8.30         | $\infty$                        | R         | <u>3.176</u> | $\times$     | 1.549        |
| EV103 | 245.00                 | 138.30       | 66.80        | <u>51.40</u> | <b>38.30</b> | $\infty$                        | $\infty$  | 8.228        | 6.185        | <b>4.085</b> |
| EV201 | 781.50                 | 42.40        | 11.00        | <b>7.60</b>  | 8.10         | $\infty$                        | $\infty$  | <b>1.582</b> | <u>1.981</u> | 2.031        |
| EV202 | 549.50                 | <u>12.50</u> | 20.20        | 13.70        | <b>9.60</b>  | $\infty$                        | $\infty$  | 3.073        | <u>1.667</u> | <b>1.446</b> |
| EV203 | 203.30                 | R            | 533.20       | $\infty$     | 185.60       | $\infty$                        | R         | $\infty$     | $\infty$     | $\infty$     |
| AVG   | $\infty$               | 28.59        | 22.18        | <u>19.24</u> | <b>17.35</b> | $\infty$                        | $\infty$  | 3.084        | <u>1.947</u> | <b>1.669</b> |
| MED   | 681.90                 | <b>12.50</b> | 20.20        | 20.00        | <u>17.50</u> | $\infty$                        | $\infty$  | 3.073        | <u>1.667</u> | <b>1.549</b> |

Basalt [64], with updates from [39]. In addition to motion-vector initialization, we adapt the original system in two ways: First, we relax the threshold for the forward-backward consistency, allowing for more tracks over the whole sequence. Second, we adapt the patch  $\mathcal{P}$  to cover a larger area. This captures more of the image structure, increasing robustness. The reported numbers for Basalt include these changes.

## 4 Experiments

We evaluate on three different VSLAM datasets representative of use cases with limited compute due to hardware constraints. EuRoC [6] represents drone data captured in two different scenarios, a machine hall and a standard room. TUM-VI [54] represents handheld motion commonly found in smartphone or body-cam footage. The Monado SLAM dataset (MSD) [39] is a recent dataset captured with a head-mounted sensor setup for applications such as VR and humanoid robotics, and is much longer than the previous two.

We use sequence acronyms with the first letter representing the dataset; e.g., EMH04 for MH\_04\_difficult from EuRoC, TR2 for room2 from TUM-VI, and MIPB05 for MIPB05\_beatsaber\_fitbeat\_360\_expert from MSD.

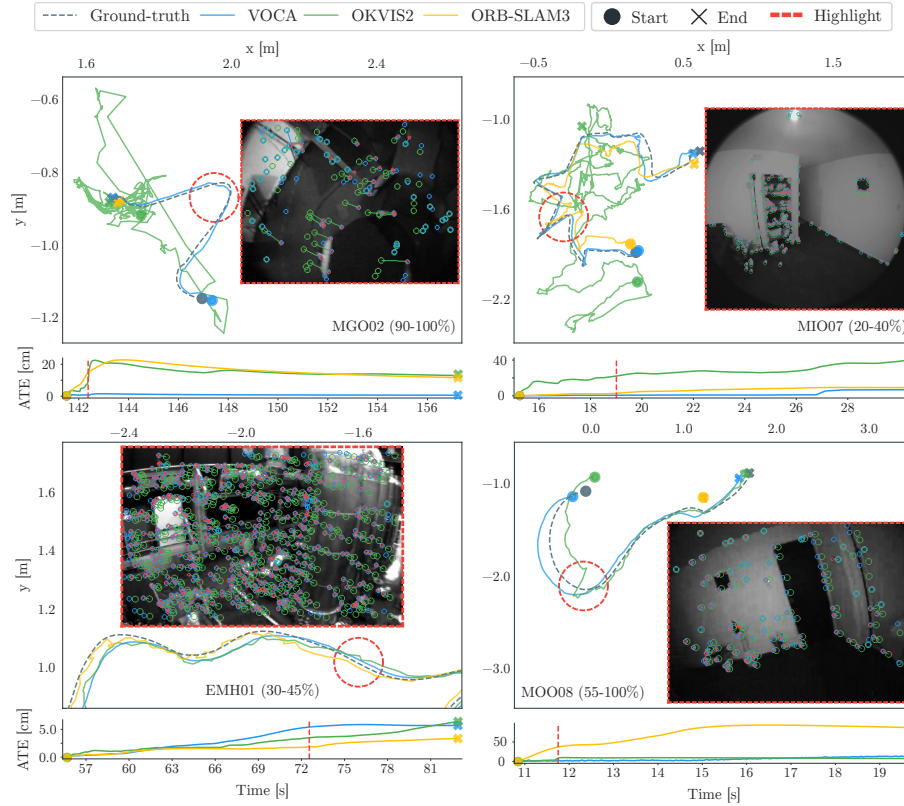
We compare against the following baselines: **1. ORB-SLAM3** [7] is a state-of-the-art descriptor-based visual SLAM system that is not designed with com-

**Table 2: TUM-VI dataset.** We restrict the evaluation to the room sequences, as they are the only ones with full ground truth. For almost all sequences, VOCA clearly outperforms the baseline Basalt [64]. The difference is more pronounced on the most challenging sequence, TR5, with an ATE improvement of  $\approx 85\%$ . Only ORB-SLAM3 achieves comparable performance in global consistency (ATE) but exceeds the RTE threshold across all sequences. Symbols and shading follow the conventions of Tab. 1.

|     | ATE [cm] (SE3 aligned) |          |             |             | RTE [cm] ( $\Delta = 6$ frames) |          |              |              |
|-----|------------------------|----------|-------------|-------------|---------------------------------|----------|--------------|--------------|
|     | ORB-SLAM3              | OKVIS2   | Basalt      | VOCA OURS   | ORB-SLAM3                       | OKVIS2   | Basalt       | VOCA OURS    |
| TR1 | 7.70                   | $\times$ | 6.30        | 6.80        | $\infty$                        | $\times$ | 1.010        | 0.896        |
| TR2 | <b>7.60</b>            | 12.00    | 12.80       | 12.70       | $\infty$                        | 1.489    | <b>0.941</b> | <b>0.941</b> |
| TR3 | <b>12.30</b>           | 21.30    | 15.30       | 15.90       | $\infty$                        | 2.028    | 0.910        | <b>0.853</b> |
| TR4 | 8.10                   | 8.70     | 14.40       | <b>7.10</b> | $\infty$                        | 2.085    | 4.458        | <b>0.733</b> |
| TR5 | 7.80                   | 54.80    | 54.50       | <b>7.70</b> | $\infty$                        | 3.907    | 7.287        | <b>1.046</b> |
| TR6 | 7.80                   | 5.80     | <b>4.70</b> | 4.80        | $\infty$                        | 1.091    | <b>0.502</b> | 0.503        |
| AVG | <b>8.72</b>            | 20.52    | 20.34       | 9.64        | $\infty$                        | 2.120    | 2.820        | <b>0.815</b> |
| MED | 7.80                   | 16.65    | 13.60       | <b>7.40</b> | $\infty$                        | 2.057    | 0.975        | <b>0.874</b> |

pressed videos in mind. Similarly, **2. OKVIS2** [32] presents another SLAM system with strong performance. **3. MoV-SLAM** [62] presents the only work in the literature that is specifically designed for compressed videos. **4. Basalt** [64] is the baseline upon which we build VOCA and is therefore conceptually the closest, as it also uses KLT-Tracking correspondences. We evaluate our method in a purely stereo VO setting. To simulate real-time usage, we report causal estimates only. We disable loop closure for ORB-SLAM3 and OKVIS2 to isolate the front-end tracking.

Whenever possible, we provide the same input data. However, MoV-SLAM and ORB-SLAM3 require stereo-rectified images for certain camera models, introducing computational overhead. As TUM-VI and MSD employ fisheye cameras with large distortions, MoV-SLAM is not reported, as it cannot work without rectification and would be heavily disadvantaged. We run the official MoV-SLAM implementation on EuRoC and report those metrics since, despite our best efforts, we were unable to reproduce the numbers claimed in [62]. We compress each camera separately with a maximum bitrate of 500 kbps when not otherwise specified, using the popular tool `ffmpeg`. We provide usage examples in the supplementary material. We adapt ORB-SLAM3, OKVIS2, and Basalt to support video input. MoV-SLAM expects a video of alternating left and right frames. This allows it to compute not only temporal but also stereo correspondences. To simulate real-world live applications, we *do not* use bidirectional encoding for video compression, giving us purely causal data. Furthermore, we evaluate the causal pose predictions of the methods, i.e., the poses of each frame after its first optimization, before future frames become available. Qualitative examples of trajectories are shown in Fig. 1 and Fig. 5. We evaluate on the common *absolute trajectory error* (ATE) and *relative trajectory error* (RTE) metrics. Details on the metrics can be found in the supplementary material.



**Fig. 5: Qualitative trajectory examples.** To showcase the quality of VOCA trajectories, in addition to Fig. 1, we provide segments covering more devices and scenarios. We highlight specific frames with dashed red outlines. Inside each frame, we draw the **optical-flow** solutions, if any, the **feature location** in the previous frame used by standard optical-flow-based feature trackers, and the **motion-vector** prior extracted from the H.264 decoder used by VOCA. As can be seen in the images, our prior is closer to the solution in almost all cases, even for large displacements such as those seen in MGO02 (from MSD).

#### 4.1 EuRoC

On EuRoC (see Tab. 1), we show competitive performance against established VO systems. Especially on the hard sequences (EMH04, EMH05, EV103, EV203), we show significant improvements over most baselines. In the RTE, we outperform the closest baseline, Basalt, by  $\approx 15\%$ , demonstrating superior relative tracking performance. VOCA recovers EV102, whereas most systems fail.

#### 4.2 TUM-VI

On the TUM-VI room sequences (Tab. 2), we show consistent improvement over the baselines. We exclude MoV-SLAM due to the previously discussed prob-

**Table 3: MSD dataset.** Given MSD’s small room environment, we reduce the divergence threshold for ATE to 1 m. The high reset count of ORB-SLAM3 (see Fig. 6) makes it necessary to exclude it from the AVG aggregate. We report the mean, median, and percentage of successful sequences (success rate, SR) over the whole dataset. Rankings for individual sequences can be found in Fig. 6, and the quantitative results can be found in Fig. 5. MSD is the most challenging dataset, with a combination of difficult movements and dynamic objects. This is the only dataset that exhibits consistent failures across all methods. Compared to the second-best method, VOCA improves median ATE by 37%, median RTE by 40%, and ATE SR by 13 percentage points (36% relative improvement).

| ATE [cm] (SE3 aligned) |           |        |        |              | RTE [cm] ( $\Delta = 6$ frames) |        |        |              |
|------------------------|-----------|--------|--------|--------------|---------------------------------|--------|--------|--------------|
|                        | ORB-SLAM3 | OKVIS2 | Basalt | VOCA OURS    | ORB-SLAM3                       | OKVIS2 | Basalt | VOCA OURS    |
| AVG                    | R         | 38.17  | 5.51   | <b>4.46</b>  | R                               | 5.223  | 1.478  | <b>0.755</b> |
| MED                    | R         | 124.70 | 479.20 | <b>78.30</b> | R                               | 8.692  | 29.124 | <b>5.212</b> |
| SR                     | 34.4%     | 31.2%  | 34.4%  | <b>46.9%</b> | 21.9%                           | 48.8%  | 39.1%  | <b>53.1%</b> |

lems with fisheye cameras. Unlike Basalt, VOCA performs consistently across sequences. The strong prior information of the motion vectors allows VOCA to overcome difficult sections like the ones in TR5, reducing both ATE and RTE.

### 4.3 Monado SLAM Dataset

Similar to TUM-VI, the reduced camera overlap means stereo-rectification would produce extremely narrow fields of view, penalizing performance disproportionately for MoV-SLAM [62], so it is also excluded from this evaluation. Overall, the dataset is significantly harder than EuRoC and TUM-VI due to regular dynamic occlusions, strong accelerations, and fast rotations. In Tab. 3, we report metrics over all sequences; all methods show a significantly worse RTE compared to other datasets. We show an overview of each sequence in Fig. 6 with the full numbers reported in the supplementary material. It is also the only dataset where methods either do not complete a sequence or produce results that are beyond the selected thresholds. Due to this greater difficulty, in Tab. 3, our work shows significant improvements when compared to the second-best method in each metric. Over the whole dataset, the median ATE and RTE improve by around 37% and 40%. In terms of the number of successfully estimated sequences (success rate) for non-diverging ATE results, VOCA shows an improvement of 36%.

### 4.4 Performance vs. Compression

In Fig. 7, we investigate how different compression bitrates affect the tracking systems on EuRoC and TUM-VI. OKVIS2 suffers especially heavily from compression. Overall, VOCA is the most stable VO system across both datasets. VOCA’s performance is relatively stable up to 500 kbps, which represents about  $\times 100$  compression for most datasets. RTE, in particular, shows quite stable results.

|      |           |                                 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|------|-----------|---------------------------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
|      |           | ATE [cm] (SE3 aligned)          |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| VOCA | Basalt    | ∞                               | ∞ | ∞ | ∞ | 1 | ∞ | 1 | 1 | 1 | 1 | ∞ | ∞ | 1 | 1 | ∞ | 3 |
|      | OKVIS2    | ∞                               | ∞ | ∞ | ∞ | ∞ | ∞ | 4 | 2 | 4 | ∞ | 4 | ∞ | ∞ | ∞ | ∞ | 2 |
|      | ORB-SLAM3 | ∞                               | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | 2 |
|      | ORB-SLAM3 | ∞                               | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | 2 |
|      |           | RTE [cm] ( $\Delta = 6$ frames) |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| VOCA | Basalt    | ∞                               | ∞ | ∞ | ∞ | 1 | ∞ | 1 | 1 | 1 | 1 | 1 | 2 | 1 | 2 | ∞ | 1 |
|      | OKVIS2    | ∞                               | ∞ | ∞ | ∞ | ∞ | ∞ | 4 | ∞ | 4 | ∞ | ∞ | 1 | 4 | 1 | ∞ | 2 |
|      | ORB-SLAM3 | ∞                               | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | 3 |
|      | ORB-SLAM3 | ∞                               | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | 3 |
|      |           | ATE [cm] (SE3 aligned)          |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| VOCA | Basalt    | ∞                               | ∞ | ∞ | ∞ | 1 | ∞ | 2 | 1 | 1 | 1 | 1 | ∞ | ∞ | ∞ | ∞ | ∞ |
|      | OKVIS2    | ∞                               | ∞ | ∞ | ∞ | ∞ | ∞ | 3 | 4 | 3 | 4 | 4 | 4 | ∞ | ∞ | ∞ | 1 |
|      | ORB-SLAM3 | ∞                               | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | 2 |
|      | ORB-SLAM3 | ∞                               | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | 2 |
|      |           | RTE [cm] ( $\Delta = 6$ frames) |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| VOCA | Basalt    | ∞                               | ∞ | ∞ | ∞ | 1 | 1 | 2 | 1 | 1 | 1 | 1 | ∞ | ∞ | ∞ | ∞ | ∞ |
|      | OKVIS2    | ∞                               | ∞ | ∞ | ∞ | ∞ | ∞ | 3 | 3 | 3 | 3 | 3 | ∞ | ∞ | ∞ | ∞ | 1 |
|      | ORB-SLAM3 | ∞                               | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | 4 |
|      | ORB-SLAM3 | ∞                               | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | 4 |

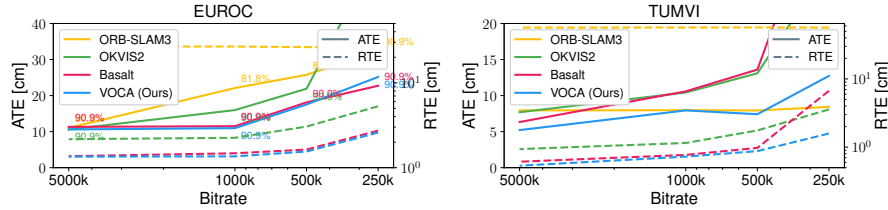
**Fig. 6: MSD overview.** We indicate divergences and resets with gray  $\infty$  and  $\odot$ , respectively. First-place ranks for each sequence are shown in green, and non-first ranks are shown in blue. See the supplementary for the detailed numbers of each run.

#### 4.5 Ablation Study

We consider three strategies for integrating motion vectors (MV) into tracking, balancing the strength of the MV prior against the risk of propagating outliers:

- (A) **OF**  $\Rightarrow$  **MV** tracks first with the base same-pixel prior, falling back to MV priors for failed points, assuming most points are easy to track and MV priors are only needed in difficult sections.
- (B) **MV**  $\Rightarrow$  **OF** reverses this order, placing high confidence in the MV prior and resorting to prior-free tracking only when it fails.
- (C) **MV**  $\parallel$  **OF** is our default strategy from Sec. 3.5. Both modes run in parallel; tracks successful in a single mode are kept; tracks successful in both modes are kept only if mutually consistent.

For (C), we additionally evaluate bridging I-frame keyframes (+ IF), as described in Sec. 3.4. Tab. 4 shows that (C) MV  $\parallel$  OF achieves the best overall performance, with (A) OF  $\Rightarrow$  MV close behind. Intuitively, most data represents easy motion where points are straightforward to track. MV priors become essential only in difficult sections. Adding I-frame bridging further improves (C), yielding the best median ATE on TUM-VI and competitive RTE across both datasets.



**Fig. 7: Median ATE/RTE vs. Bitrate.** Lower bitrates lead to smaller file sizes. Numbers indicate a success rate of less than 100% for ATE. VOCA shows stable performance even at low bitrates. As a tracking-based system, Basalt [64] suffers from compression artifacts, especially on the TUM-VI dataset. ORB-SLAM3 [7] shows steady performance in ATE on TUM-VI but delivers poor performance in RTE. This indicates good global consistency but poor frame-to-frame tracking.

**Table 4: Ablation study.** We evaluate three motion-vector (MV) integration strategies in combination with optical flow (OF) on EuRoC and TUM-VI: (A) OF  $\Rightarrow$  MV tracks without priors first and falls back to MV priors; (B) reverses this order; (C) MV || OF runs both in parallel, keeping single-mode tracks directly and filtering dual-mode tracks by consistency. + IF denotes I-frame bridging. (C) + IF is our method, VOCA.

|        |     | ATE [cm] (SE3 aligned) |              |              |             |             | RTE [cm] ( $\Delta = 6$ frames) |       |       |              |              |
|--------|-----|------------------------|--------------|--------------|-------------|-------------|---------------------------------|-------|-------|--------------|--------------|
|        |     | Basalt                 | (A)          | (B)          | (C)         | (C)<br>+ IF | Basalt                          | (A)   | (B)   | (C)          | (C)<br>+ IF  |
| EuRoC  | AVG | 19.50                  | <b>17.20</b> | 18.40        | 18.70       | 18.00       | 1.964                           | 1.871 | 2.044 | <b>1.710</b> | 1.714        |
|        | MED | 20.00                  | <b>12.80</b> | <u>16.80</u> | 17.30       | 17.50       | 1.667                           | 1.608 | 1.656 | <u>1.552</u> | <b>1.549</b> |
| TUM-VI | AVG | 18.00                  | 10.40        | 10.70        | <b>9.10</b> | <u>9.20</u> | 2.518                           | 1.380 | 1.121 | <u>0.919</u> | <b>0.829</b> |
|        | MED | 13.60                  | 10.30        | 10.60        | <u>8.90</u> | <b>7.40</b> | 0.975                           | 1.234 | 0.968 | <b>0.872</b> | <u>0.874</u> |

## 5 Conclusion

We present VOCA, a Visual Odometry system with codec awareness. By leveraging motion vectors and I-frame structure from the video codec to aid KLT-Tracking, VOCA overcomes difficult sections, achieving more accurate and stable trajectories. This is particularly apparent in RTE, a metric critical for mixed reality, where local smoothness directly impacts user experience. Codec-aware tracking is equally relevant for autonomous systems where bandwidth and memory constraints make compression unavoidable, such as drones, wearables, and robotic assistants under a compress-then-analyze design [24]. Unlike conventional SLAM systems, VOCA remains robust under heavy compression. With the vast majority of video data on the internet being codec-compressed, codec-aware methods like VOCA are essential for efficiently unlocking large amounts of data for downstream vision and learning tasks. We hope this work inspires further research on leveraging information from codecs in computer vision tasks.

**Acknowledgments.** This work was supported by the European Research Council (ERC) Advanced Grant SIMULACRON, by the DFG project CR 250/26-1 “4D-YouTube”, by the GNI Project “AI4Twinning”, and by the Munich Center for Machine Learning.

## References

1. Arunruangsirilert, K., Katto, J.: Evaluation of hardware-based video encoders on modern gpus for uhd live-streaming. In: 2024 33rd International Conference on Computer Communications and Networks (ICCCN). pp. 1–9 (2024). <https://doi.org/10.1109/ICCCN61486.2024.10637525>
2. Bahl, S., Mendonca, R., Chen, L., Jain, U., Pathak, D.: Affordances from Human Videos as a Versatile Representation for Robotics. In: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 01–13. IEEE, Vancouver, BC, Canada (Jun 2023). <https://doi.org/10.1109/CVPR52729.2023.01324>
3. Banerjee, P., Shkodrani, S., Moulon, P., Hampali, S., Han, S., Zhang, F., Zhang, L., Fountain, J., Miller, E., Basol, S., Newcombe, R., Wang, R., Engel, J.J., Hodan, T.: HOT3D: Hand and Object Tracking in 3D from Egocentric Multi-View Videos. In: 2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 7061–7071 (Jun 2025). <https://doi.org/10.1109/CVPR52734.2025.00662>, <https://ieeexplore.ieee.org/document/11092663>
4. Bay, H., Tuytelaars, T., Van Gool, L.: Surf: Speeded up robust features. In: European conference on computer vision. pp. 404–417. Springer (2006)
5. Bross, B., Wang, Y.K., Ye, Y., Liu, S., Chen, J., Sullivan, G.J., Ohm, J.R.: Overview of the versatile video coding (vvc) standard and its applications. *IEEE Transactions on Circuits and Systems for Video Technology* **31**(10), 3736–3764 (2021). <https://doi.org/10.1109/TCSVT.2021.3101953>
6. Burri, M., Nikolic, J., Gohl, P., Schneider, T., Rehder, J., Omari, S., Achtelik, M.W., Siegwart, R.: The euroc micro aerial vehicle datasets. *The International Journal of Robotics Research* **35**(10), 1157–1163 (2016)
7. Campos, C., Elvira, R., Rodríguez, J.J.G., Montiel, J.M., Tardós, J.D.: Orb-slam3: An accurate open-source library for visual, visual-inertial, and multimap slam. *IEEE transactions on robotics* **37**(6), 1874–1890 (2021)
8. Carlone, L., Kim, A., Barfoot, T., Cremers, D., Dellaert, F.: Slam handbook: From localization and mapping to spatial intelligence (2025)
9. Chen, H., Sun, B., Zhang, A., Pollefeys, M., Leutenegger, S.: VidBot: Learning generalizable 3d actions from in-the-wild 2d human videos for zero-shot robotic manipulation. In: Proceedings of the Computer Vision and Pattern Recognition Conference (2025)
10. Chen, W., Chen, L., Wang, R., Pollefeys, M.: Leap-vo: Long-term effective any point tracking for visual odometry. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 19844–19853 (2024)
11. Chi, Y., Sommer, L., Dünkler, O., Muhle, D., Cremers, D., Theobalt, C., Kortylewski, A.: C3po: Canonicalization of 3d pose from partial views with generalizable correspondence features. In: 2026 International Conference on 3D Vision (3DV). pp. 587–597. IEEE (2026)
12. Chng, C.K., Parra, A., Chin, T.J., Latif, Y.: Monocular rotational odometry with incremental rotation averaging and loop closure. In: 2020 Digital Image Computing: Techniques and Applications (DICTA). pp. 1–8. IEEE (2020)
13. Cin, A.P.D., Dikov, G., Ju, J., Ghafoorian, M.: Anymap: Learning a general camera model for structure-from-motion with unknown distortion in dynamic scenes. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 16674–16684 (2025)
14. Deng, K., Ti, Z., Xu, J., Yang, J., Xie, J.: Vggt-long: Chunk it, loop it, align it—pushing vgg’s limits on kilometer-scale long rgb sequences. *arXiv preprint arXiv:2507.16443* (2025)



15. DeTone, D., Malisiewicz, T., Rabinovich, A.: Superpoint: Self-supervised interest point detection and description. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops. pp. 224–236 (2018)
16. Engel, J., Koltun, V., Cremers, D.: Direct sparse odometry. *IEEE transactions on pattern analysis and machine intelligence* **40**(3), 611–625 (2017)
17. Engel, J., Schöps, T., Cremers, D.: Lsd-slam: Large-scale direct monocular slam. In: European conference on computer vision. pp. 834–849. Springer (2014)
18. Geneva, P., Eickenhoff, K., Lee, W., Yang, Y., Huang, G.: Opencvins: A research platform for visual-inertial estimation. In: 2020 IEEE International Conference on Robotics and Automation (ICRA). pp. 4666–4672. IEEE (2020)
19. Gross, M., Fahmy, A., Niwattananan, D., Muhle, D., Song, R., Cremers, D., Meeß, H.: Ipformer: Visual 3d panoptic scene completion with context-adaptive instance proposals. *Advances in Neural Information Processing Systems* **38**, 4989–5014 (2026)
20. Han, J., Li, B., Mukherjee, D., Chiang, C.H., Grange, A., Chen, C., Su, H., Parker, S., Deng, S., Joshi, U., Chen, Y., Wang, Y., Wilkins, P., Xu, Y., Bankoski, J.: A technical overview of av1. *Proceedings of the IEEE* **109**(9), 1435–1462 (2021). <https://doi.org/10.1109/JPROC.2021.3058584>
21. Han, K., Muhle, D., Wimbauer, F., Cremers, D.: Boosting self-supervision for single-view scene completion via knowledge distillation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9837–9847 (2024)
22. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press, Cambridge, 2 edn. (2004). <https://doi.org/10.1017/CB09780511811685>
23. Hayler, A., Wimbauer, F., Muhle, D., Rupperecht, C., Cremers, D.: S4c: Self-supervised semantic scene completion with neural fields. In: 2024 International Conference on 3D Vision (3DV). pp. 409–420. IEEE (2024)
24. Hofer, J., et al.: H.264 Compress-Then-Analyze Transmission in Edge-Assisted Visual SLAM. In: European Wireless 2023; 28th European Wireless Conference. pp. 130–135 (2023)
25. Hsiao, Y.M., Lee, J.F., Chen, J.S., Chu, Y.S.: Review: H.264 video transmissions over wireless networks: Challenges and solutions. *Comput. Commun.* **34**(14), 1661–1672 (Sep 2011). <https://doi.org/10.1016/j.comcom.2011.03.016>
26. International Telecommunication Union: ITU-T Recommendation H.262: Information technology – Generic coding of moving pictures and associated audio information: Video. <https://www.itu.int/rec/T-REC-H.262> (Jan 2021), accessed: 2026-02-08
27. Ji, Y., Tan, H., Shi, J., Hao, X., Zhang, Y., Zhang, H., Wang, P., Zhao, M., Mu, Y., An, P., Xue, X., Su, Q., Lyu, H., Zheng, X., Liu, J., Wang, Z., Zhang, S.: Robobrain: A unified brain model for robotic manipulation from abstract to concrete. 2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 1724–1734 (2025)
28. Kantorov, V., Laptev, I.: Efficient feature extraction, encoding, and classification for action recognition. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition. pp. 2593–2600 (2014). <https://doi.org/10.1109/CVPR.2014.332>
29. Keetha, N., Müller, N., Schönberger, J., Porzi, L., Zhang, Y., Fischer, T., Knapitsch, A., Zaus, D., Weber, E., Antunes, N., et al.: Mapanything: Universal feed-forward metric 3d reconstruction; map-anything. *github. io*. In: 2026 International Conference on 3D Vision (3DV). pp. 499–509. IEEE (2026)

30. Klein, G., Murray, D.: Parallel tracking and mapping for small ar workspaces. In: 2007 6th IEEE and ACM international symposium on mixed and augmented reality. pp. 225–234. IEEE (2007)
31. Lee, S.H., Civera, J.: Rotation-only bundle adjustment. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 424–433 (2021)
32. Leutenegger, S.: Okvis2: Realtime scalable visual-inertial slam with loop closure. arXiv preprint arXiv:2202.09199 (2022)
33. Leutenegger, S., Chli, M., Siegwart, R.Y.: Brisk: Binary robust invariant scalable keypoints. In: 2011 International conference on computer vision. pp. 2548–2555. Ieee (2011)
34. Lin, H., Chen, S., Liew, J., Chen, D.Y., Li, Z., Shi, G., Feng, J., Kang, B.: Depth anything 3: Recovering the visual space from any views. arXiv preprint arXiv:2511.10647 (2025)
35. Lin, L., Yu, S., Zhou, L., Chen, W., Zhao, T., Wang, Z.: Pea265: Perceptual assessment of video compression artifacts. *IEEE Transactions on Circuits and Systems for Video Technology* **30**(11), 3898–3910 (2020)
36. Liou, M.: Overview of the p×64 kbit/s video coding standard. *Commun. ACM* **34**(4), 59–63 (Apr 1991). <https://doi.org/10.1145/103085.103091>
37. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. *International journal of computer vision* **60**(2), 91–110 (2004)
38. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: Hayes, P.J. (ed.) *Proceedings of the 7th International Joint Conference on Artificial Intelligence, IJCAI '81, Vancouver, BC, Canada, August 24-28, 1981*. pp. 674–679. William Kaufmann (1981)
39. de Mayo, M., Cremers, D., Pire, T.: The monado slam dataset for egocentric visual-inertial tracking. In: 2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 13111–13118. IEEE (2025)
40. Muhle, D., Koestler, L., Demmel, N., Bernard, F., Cremers, D.: The probabilistic normal epipolar constraint for frame-to-frame rotation optimization under uncertain feature positions. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 1819–1828 (2022)
41. Muhle, D., Koestler, L., Jatavallabhula, K.M., Cremers, D.: Learning correspondence uncertainty via differentiable nonlinear least squares. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 13102–13112 (2023)
42. Mukherjee, D., Bankoski, J., Grange, A., Han, J., Koleszar, J., Wilkins, P., Xu, Y., Bultje, R.: The latest open-source video codec vp9 - an overview and preliminary results. In: 2013 Picture Coding Symposium (PCS). pp. 390–393 (2013). <https://doi.org/10.1109/PCS.2013.6737765>
43. Mur-Artal, R., Montiel, J.M.M., Tardós, J.D.: Orb-slam: A versatile and accurate monocular slam system. *IEEE Transactions on Robotics* **31**(5), 1147–1163 (2015). <https://doi.org/10.1109/TR0.2015.2463671>
44. Mur-Artal, R., Tardós, J.D.: Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics* **33**(5), 1255–1262 (2017). <https://doi.org/10.1109/TR0.2017.2705103>
45. Murai, R., Dexheimer, E., Davison, A.J.: Mast3r-slam: Real-time dense slam with 3d reconstruction priors. In: *Proceedings of the Computer Vision and Pattern Recognition Conference*. pp. 16695–16705 (2025)

46. Peroni, L., Gorinsky, S.: An end-to-end pipeline perspective on video streaming in best-effort networks: A survey and tutorial. *ACM Computing Surveys* **57**(12), 1–47 (Jul 2025). <https://doi.org/10.1145/3742472>, <http://dx.doi.org/10.1145/3742472>
47. Qian, S., Mo, K., Blukis, V., Fouhey, D.F., Fox, D., Goyal, A.: 3D-MVP: 3D Multiview Pretraining for Manipulation. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2025)
48. Reich, C., Hahn, O., Cremers, D., Roth, S., Debnath, B.: A perspective on deep vision performance with standard image and video codecs. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 5712–5721 (2024)
49. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: Orb: An efficient alternative to sift or surf. In: *2011 International Conference on Computer Vision*. pp. 2564–2571 (2011). <https://doi.org/10.1109/ICCV.2011.6126544>
50. Rückert, D., Stamminger, M.: Snake-slam: Efficient global visual inertial slam using decoupled nonlinear optimization. In: *2021 International conference on unmanned aircraft systems (ICUAS)*. pp. 219–228. IEEE (2021)
51. Sandström, E., Zhang, G., Tateno, K., Oechsle, M., Niemeyer, M., Zhang, Y., Patel, M., Van Gool, L., Oswald, M., Tombari, F.: Splat-slam: Globally optimized rgb-only slam with 3d gaussians. In: *Proceedings of the Computer Vision and Pattern Recognition Conference*. pp. 1680–1691 (2025)
52. Schafer, R., Sikora, T.: Digital video coding standards and their role in video communications. *Proceedings of the IEEE* **83**(6), 907–924 (1995). <https://doi.org/10.1109/5.387092>
53. Schonberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 4104–4113 (2016)
54. Schubert, D., Goll, T., Demmel, N., Usenko, V., Stückler, J., Cremers, D.: The tum vi benchmark for evaluating visual-inertial odometry. In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. pp. 1680–1687. IEEE (2018)
55. Seufert, M., Egger, S., Slanina, M., Zinner, T., Hoßfeld, T., Tran-Gia, P.: A survey on quality of experience of http adaptive streaming. *IEEE Communications Surveys & Tutorials* **17**(1), 469–492 (2015). <https://doi.org/10.1109/COMST.2014.2360940>
56. Shi, J., Tomasi, C.: Good features to track. In: *Conference on Computer Vision and Pattern Recognition, CVPR 1994, 21-23 June, 1994, Seattle, WA, USA*. pp. 593–600. IEEE (1994). <https://doi.org/10.1109/CVPR.1994.323794>, <https://doi.org/10.1109/CVPR.1994.323794>
57. Smith, C., Charatan, D., Tewari, A., Sitzmann, V.: Flowmap: High-quality camera poses, intrinsics, and depth via gradient descent. In: *2025 International Conference on 3D Vision (3DV)*. pp. 389–400. IEEE (2025)
58. Stachniss, C., Leonard, J.J., Thrun, S.: *Simultaneous localization and mapping*. In: *Springer handbook of robotics*, pp. 1153–1176. Springer (2016)
59. Sullivan, G.J., Ohm, J.R., Han, W.J., Wiegand, T.: Overview of the high efficiency video coding (hevc) standard. *IEEE Transactions on Circuits and Systems for Video Technology* **22**(12), 1649–1668 (2012). <https://doi.org/10.1109/TCSVT.2012.2221191>
60. Teed, Z., Deng, J.: Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. *Advances in neural information processing systems* **34**, 16558–16569 (2021)

61. Tomasi, C., Kanade, T.: Detection and tracking of point features. Tech. rep., International Journal of Computer Vision (1991)
62. Turner, R.N., Banerjee, N.K., Banerjee, S.: Mov-slam: Using motion vectors for real-time single-cpu visual slam. In: 2023 Seventh IEEE International Conference on Robotic Computing (IRC). pp. 51–58. IEEE (2023)
63. Ungureanu, D., Bogu, F., Galliani, S., Sama, P., Duan, X., Meekhof, C., Stühmer, J., Cashman, T.J., Tekin, B., Schönberger, J.L., Olszta, P., Pollefeys, M.: HoloLens 2 Research Mode as a Tool for Computer Vision Research (Aug 2020). <https://doi.org/10.48550/arXiv.2008.11239>, <http://arxiv.org/abs/2008.11239>
64. Usenko, V., Demmel, N., Schubert, D., Stückler, J., Cremers, D.: Visual-inertial mapping with non-linear factor recovery. *IEEE Robotics Autom. Lett.* **5**(2), 422–429 (2020). <https://doi.org/10.1109/LRA.2019.2961227>, <https://doi.org/10.1109/LRA.2019.2961227>
65. Von Stumberg, L., Cremers, D.: Dm-vio: Delayed marginalization visual-inertial odometry. *IEEE Robotics and Automation Letters* **7**(2), 1408–1415 (2022)
66. Wang, J., Chen, M., Karaev, N., Vedaldi, A., Rupprecht, C., Novotny, D.: Vggt: Visual geometry grounded transformer. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 5294–5306 (2025)
67. Wang, Y., Zhou, J., Zhu, H., Chang, W., Zhou, Y., Li, Z., Chen, J., Pang, J., Shen, C., He, T.:  $\pi^3$ : Permutation-equivariant visual geometry learning. arXiv preprint arXiv:2507.13347 (2025)
68. Wiegand, T., Sullivan, G., Bjontegaard, G., Luthra, A.: Overview of the h.264/avc video coding standard. *IEEE Transactions on Circuits and Systems for Video Technology* **13**(7), 560–576 (2003). <https://doi.org/10.1109/TCSVT.2003.815165>
69. Wimbauer, F., Chen, W., Muhle, D., Rupprecht, C., Cremers, D.: Anycam: Learning to recover camera poses and intrinsics from casual videos. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 16717–16727 (2025)
70. Wimbauer, F., Yang, N., Rupprecht, C., Cremers, D.: Behind the scenes: Density fields for single view reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9076–9086 (2023)
71. Zhou, S., Jiang, X., Tan, W., He, R., Yan, B.: Mvflow: Deep optical flow estimation of compressed videos with motion vector prior. In: Proceedings of the 31st ACM International Conference on Multimedia. pp. 1964–1974 (2023)
72. Zhu, Z., Akkaya, I.B., Waeijen, L., Bondarev, E., Pourtaherian, A., Moreira, O.: MEET: Towards Memory-Efficient Temporal Sparse Deep Neural Networks. In: 2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 29309–29320 (Jun 2025). <https://doi.org/10.1109/CVPR52734.2025.02729>, <https://ieeexplore.ieee.org/document/11092745>
73. Zouein, J., Javidnia, H., Pitié, F., Kokaram, A.: Leveraging AV1 Motion Vectors for Fast and Dense Feature Matching. In: 2025 IEEE 4th International Conference on Intelligent Reality (ICIR). pp. 1–4. <https://doi.org/10.1109/ICIR68135.2025.11361611>
74. Zouein, J., Vibhoothi, V., Kokaram, A.: AV1 Motion Vector Fidelity and Application for Efficient Optical Flow. In: 2025 Picture Coding Symposium (PCS). pp. 1–5. <https://doi.org/10.1109/PCS65673.2025.11417638>