frontiers

A Dataset for Visual Navigation with Neuromorphic Methods

Francisco Barranco [1,2,*], **Cornelia Fermüller** [2] **Yiannis Aloimonos** [2], **Tobi Delbruck** [3]

[1] *CITIC, Department of Computer Architecture and Computer Technology, University of Granada, Granada, Spain*
[2] *Computer Vision Lab, UMIACS, Department of Computer Science, University of Maryland, College Park, Maryland, USA*
[3] *Department of Information Technology and Electrical Engineering, Institute of Neuroinformatics, ETH Zurich and University of Zurich, Zurich, Switzerland*

Correspondence*:
corresponding Author
UMIACS, A.V. Williams Bldg, University of Maryland, 20740, College Park, Maryland, USA, barranco@umiacs.umd.edu

Benchmarks and Challenges for Neuromorphic Engineering

ABSTRACT

Standardized benchmarks in Computer Vision have greatly contributed to the advance of approaches to many problems in the field. If we want to enhance the visibility of event-driven vision and increase its impact, we will need benchmarks that allow comparison among different neuromorphic methods as well as comparison to Computer Vision conventional approaches. We present datasets to evaluate the accuracy of frame-free and frame-based approaches for tasks of visual navigation. Similar to conventional Computer Vision datasets, we provide synthetic and real scenes, with the synthetic data created with graphics packages, and the real data recorded using a mobile robotic platform carrying a dynamic and active pixel vision sensor (DAVIS) and an RGB+Depth sensor. For both datasets the cameras move with a rigid motion in a static scene, and the data includes the images, events, optic flow, 3D camera motion, and the depth of the scene, along with calibration procedures. Finally, we also provide simulated event data generated synthetically from well-known frame-based optical flow datasets.

Keywords: Event-driven methods, Frame-free sensors, Visual navigation, Dataset, Calibration

1 INTRODUCTION

Asynchronous frame-free vision sensors have gained popularity among vision researchers in recent years. The most prominent of these sensors are the temporal change threshold imager (**Mallik et al.**, 2005), the DVS (**Lichtsteiner et al.**, 2008), the ATIS (**Posch et al.**, 2011), and the DAVIS (**Brandli et al.**, 2014). Inspiration for their design comes from the transient pathway of primate vision, which processes information due to luminance changes in the scene (**Lichtsteiner et al.**, 2008; **Liu et al.**, 2015). Their properties, such as the high temporal resolution (triggering temporal contrast events with a resolution of a few microseconds), low-bandwidth, low-computational resource requirements, low-latency, and real-time performance, make them interesting for many applications of motion perception. While conventional cameras record image luminance at fixed time intervals, frame-free vision sensors record asynchronously the time and location, where changes in the luminance occur.

Visual motion analysis for navigation is about relating the observed intensity changes on the imaging device to the 3D scene geometry and the 3D motion of the observer (or imaging device) relative to the scene. The computational analysis involves two distinct processes: the estimation of observed image motion on the imaging surface due to the movement of scene points, in Computer Vision usually called *optical flow*, and the estimation of the geometry and dynamics of the scene on the basis of image motion. Visual navigation, in general, involves moving cameras in environments that can be dynamic as well, and it refers to a set of tasks ranging from obstacle avoidance, over object tracking, 3D motion estimation and scene segmentation, to map making. Currently, however, our dataset has static scenes only. We provide the raw data along with the 3D motion and the scene geometry, and this data allows for evaluating algorithms concerned with the classic *structure from motion* problems of image motion estimation, 3D motion estimation, reconstruction, and segmentation by depth.

Evaluation datasets drive applications and challenge researchers to develop techniques that are widely applicable, consider diverse scenarios, and have high accuracy. The Computer Vision community has realized their importance for many years, and has provided datasets for many applications, including visual navigation. Among the best known datasets for image motion one can find Middlebury (**Baker et al.**, 2011), MPI Sintel (**Butler et al.**, 2012), and KITTI (**Geiger et al.**, 2012). Middlebury, a benchmark that also provides a creative ranking of methods, has been the standard until the last few years. The more recent MPI Sintel and KITTI datasets include scenarios of greater complexity and much larger image motion. The former consists of synthetic sequences and has many challenging cases such as transparencies, blurring, or variations in illumination. The latter has sequences from real-world driving scenarios, and provides besides optical flow also ground-truth for 3D motion, structure, and the tracking of objects. Other well-known data sets for 3D motion and structure include the CMU dataset (**Badino et al.**, 2015), the TUM dataset (**Sturm et al.**, 2012), as well as the KITTI dataset (**Geiger et al.**, 2012). These datasets were designed for evaluation of navigation and localization algorithms.

Along with datasets, we also need metrics to evaluate the techniques. The metrics of Computer Vision focused mostly on accuracy. Image motion is usually evaluated by the average error of either the flow vectors (**Otte and Nagel**, 1994), or their directions (**Fleet and Jepson**, 1990). 3D camera motion is evaluated by the average error in the direction of the rotation axis, the angular velocity, and the direction of translation (see (**Raudies and Neumann**, 2012)). Clearly, the average error does not capture fully the quality of a method, given the heterogeneity of sequences in the different datasets. In (**Sun et al.**, 2014), statistical significance tests provide a way to cope with this problem.

A few of the methods published in the event-based literature included evaluations. Several methods evaluated the accuracy of image motion estimation methods, e.g. (**Tschechne et al.**, 2014; **Benosman et al.**, 2014; **Barranco et al.**, 2014; **Orchard and Etienne-Cummings**, 2014), and (**Censi and Scaramuzza**, 2014) evaluated odometry estimation. However, all these methods used their own datasets. Therefore, so far there is a lack of comparisons between different event-based methods and comparisons to Computer Vision methods. Another paper, which is part of this special issue (**Ruckauer and Delbruck**, 2015) provides a dataset for the evaluation of event-based flow methods and also releases codes for the evaluated methods. However, this work is the first to present a dataset that facilitates comparison of event-based and frame-based methods for 2D and 3D visual navigation tasks.

Our real-time dataset was collected with a mobile platform carrying a DAVIS sensor (**Brandli et al.**, 2014) and an RGB-D sensor (RGB + Depth sensor). The DAVIS sensor provides asynchronous streams of events called DVS events, and synchronous sequences of image frames called APS frames. From the RGB-D sensor we obtain the depth maps of the scene and from the odometry of the platform we obtain the 3D motion. Using the 3D motion and depth, we compute the image motion. In addition to the data, we also provide the code for the calibration of the DAVIS sensor with respect to the RGB-D sensor (using the synchronous frames of the DAVIS), and the calibration between the robotic platform and the DAVIS sensor. We use the same metrics as in conventional methods to evaluate the accuracy of event-driven methods. To account for the sparseness of the event data, we also include a measure of the data density.

The paper is structured as follows: §2 describes current datasets of visual navigation from Computer Vision. Next, §3 describes how we created the event-based dataset. §4 reviews different metrics for evaluation and §5 presents some of the sequences of our dataset. Finally, §6 concludes the work.

2 DATASETS IN COMPUTER VISION

Benchmarks, datasets and quantifiable metrics to estimate accuracy are very common in the Computer Vision literature. They have greatly influenced the development of Computer Vision techniques for different applications, and contributed to market solutions in demanding fields such as medical image analysis, autonomous driving, and robotics.

There are a number of benchmarks for visual navigation. Barron et al. (**Barron et al.**, 1994) were the first to propose a benchmark and quantitative evaluation of optical flow methods. This dataset of synthetic scenes was then replaced by the Middlebury database (**Baker et al.**, 2011), which contains much more challenging datasets of synthetic and real scenes with objects at different depth causing motion discontinuities. The success of Middlebury may be partly due to its evaluation platform: through a web interface one can upload the results of a motion estimation method so it will be compared to the state-of-the-art. Half of the example sequences are provided with the ground-truth as training set to allow users to tune their methods. For evaluation, authors are instructed to estimate the motion for the remainder of the sequences (the test set) whose ground-truths are not provided, and to submit them through the web application. Then, the methods are ranked according to different error metrics: endpoint error, angular error, interpolation error, and normalized interpolation error. The most recent prominent datasets, MPI Sintel (**Butler et al.**, 2012) and KITTI (**Geiger et al.**, 2012) are much more challenging. They provide long video sequences at high spatial resolution, and the image motion between frames spans a large range of values (even exceeding 100 pixels), and thus actually the video frames in these sets are closer to stereo than image motion configurations. The sequences include deformable objects and introduce very complex problems such as transparencies, shadows, smoke and lighting variations. Masks for motion boundaries and for unmatched pixels are included, and new metrics are described to measure the image motion accuracy in these areas. MPI Sintel, which is generated with a computer graphic model, provides different variations of its sequence, such as with and without motion blur.

Several other datasets provide benchmarks for 3D position and pose estimation. Usually they include sequences of image frames and the corresponding six parameters of the camera motion defined by the rotation and the translation. Some of these datasets also provide corresponding sequences of depth maps and image motion fields. (**Raudies and Neumann**, 2009) used the earlier created *Yosemite* sequence, a synthetic fly-through sequence over the so-named valley, and created the synthetic *Fountain* sequence with a curvilinear motion for a patio sequence. KITTI (**Geiger et al.**, 2012) provides a dataset for 3D visual navigation, specifically created for autonomous driving. It includes data from a stereo camera rig, a laser scanner, and GPS/IMU signals. The CMU dataset, available at (**Badino et al.**, 2015), uses the same sensors also mounted on a car. The data of the TUM dataset (**Sturm et al.**, 2012) includes images and depth frames captured with an RGB-D sensor (Microsoft Kinect). The ground-truth odometry was estimated from the external camera-based tracking system and the RGB-D sensor data.

3 DATASET DESIGN

Event-based sensors and frame-based cameras record very different kinds of data streams, and thus to create a benchmark for their comparison is quite challenging. While conventional frame-based sensors record scene luminance, which is static scene information, event-based sensors record changes in the luminance, which is dynamic scene information. Conventional cameras have a higher spatial resolution than event-based sensors, but their temporal resolution is fixed, usually up to approximately 60 fps (frames per second). In contrast, for frame-free sensors there is no fixed sampling period, which can be as small

as a few microseconds. To compare static images to events, a few works (such as (**Pérez-Carrasco et al.**, 2013)) were shaking the sensor. This technique, however, is not applicable for visual navigation, as it would introduce too much additional noise. Indeed, we require a conventional sensor and a frame-free sensor collecting data of the same scene. For our dataset we used the DAVIS sensor, which collects both asynchronous brightness-change events and synchronous frames.

The synthetic data in our benchmark was created from existing Computer Vision datasets (Section §3.1), and includes two sets. First, we generated events (**Barranco et al.**, 2014) for the optic flow sequences provided in (**Baker et al.**, 2011) and (**Barron et al.**, 1994). The such created dataset allows comparison to the large number of existing optic flow techniques in the Computer Vision literature, but it is not accurate due to the lack of ground-truth information (in the original optical flow sequences) in areas occluded between consecutive frames and ambiguities in the depth discontinuities. This problem was overcome in a second dataset which was built from a graphics-generated 3D scene model (**Barranco et al.**, 2015). The real data in our benchmark was collected with a mobile robot carrying a rig on which we mounted a DAVIS sensor and an RGB-D sensor (RGB images plus Depth) (Section §3.2). By calibrating the DAVIS sensor with the depth sensor, we obtained the data required for reconstructing the 3D scene model. The simple odometry system, consisting of a gyroscope and an accelerometer, provided the 3D motion ground-truth.

Note, that we computed the motion of the sensor using the odometry of our platform. An alternative, much easier approach to obtain 3D sensor estimates, would be to use an external motion capture system (**Voigt et al.**, 2011). However, motion capture systems are expensive and cannot be used for outdoor scenarios.

Our dataset is available at `http://atcproyectos.ugr.es/realtimeasoc/protected/evbench.html` (user: *reviewer* and pass: *frontiers2015*). It includes the DAVIS sequences (DVS events and APS frames), the Kinect data (RGB images and depth maps), the generated motion flow fields, and the 3D camera motion (translation and rotation). The code for the different calibration procedures, registrations, and for computing the evaluation metrics, described in the next sections, are available at the software repository `https://github.com/fbarranco/eventVision-evbench`.

3.1 SIMULATED EVENTS FROM CURRENT COMPUTER VISION DATASETS

The first dataset was created from the sequences in Middlebury (**Baker et al.**, 2011) by simulating the events on the basis of the ground truth optic flow (**Barranco et al.**, 2014, 2015). Real frame-free sensors trigger an event when the intensity difference at a point exceeds a predetermined value (more exactly when the change in log Intensity exceeds a threshold). To simulate this, we first interpolate image frames in time using the optic flow information. Assuming a frame rate of 20 fps (frames per second) the optic flow sequences, we interpolate 50000 samples between pairs of consecutive frames to achieve a simulated temporal resolution of 1 μs in the DVS. Then events (with exact timestamp) are created, by checking at every position for changes greater than the threshold. However, this simulation only works at image regions due to smooth surfaces, but not at occlusion regions, where usually ground-truth flow is not provided. To perform reconstruction, a 3D model of the scene is required. In its absence we generated our data using the following approximation: we differentiate between occluded regions, which are pixels visible in the previous frame but not the current, and dis-occluded regions, which are pixels not visible in the previous frame, but uncovered in the current frame. Intensity values of occluded regions are obtained from the previous frame and those of dis-occlusions from the subsequent frame. For non-static regions, we assume the same motion for the background and the region. More complex scenarios, including non-regular motion patterns or occluded objects with different motions, are discarded.

The second dataset was created in a way similar to the MPI Sintel dataset (**Butler et al.**, 2012). Using a 3D graphics model of the scene and information on the 3D motion and 3D pose of the camera, we reconstructed the motion flow field and stream of events (**Barranco et al.**, 2015). Specifically, we used the 3D model, the textures, and the 3D motion ground-truth provided by (**Mac Aodha et al.**, 2013), which were created using the 3D software and modeling tool Maya (see `http://www.autodesk.com/`

products/maya). We note that for a more realistic simulation, one could additionally add simulated noise on the events using appropriate probability distributions.

3.2 DAVIS MOUNTED ON A MOBILE PLATFORM

The DAVIS sensor (**Brandli et al.**, 2014) and a Microsoft Kinect Sensor (providing an RGB image and depth map) are mounted on a stereo rig and the stereo rig is mounted on a Pan Tilt Unit (PTU-46-17P70T by FLIR Motion Control Systems) on-board a Pioneer 3DX Mobile Robot. The motion is due to the rotation of the PTU defined by pan and tilt angles and angular velocities, and the translation of the Pioneer 3DX Mobile Robot defined by the direction of translation and the speed. ROS (Robot Operating System) packages are available for both the PTU and the Pioneer 3DX mobile robot. Fig. 1 shows the Pan Tilt Unit on the left, the Pioneer 3DX mobile robot in the center, and the DAVIS sensor (a DAVIS240b by Inilabs) on the right.

Our dataset provides the following:

- *The 3D motion parameters: 3D translation and 3D pose of the camera*. These are provided by the PTU and the Pioneer Mobile Robot. Calibration of the PTU with respect to the platform, and calibration of the DAVIS with respect to the PTU are required.
- *The image depth* in the coordinate system of the DAVIS. Depth is obtained by the Microsoft Kinect Sensor (RGB-D sensor). A stereo calibration registering the Kinect depth to the DAVIS camera coordinates is required.
- *The 2D motion flow field*. Using the 3D motion and depth, the 2D motion flow field in the DAVIS coordinate system is computed.

DAVIS AND RGB-D SENSOR CALIBRATION

The RGB-D sensor provides the depth of the scene. This depth needs to be transformed to the coordinate system of the DAVIS. In our procedure, we first calibrate the two cameras individually, both for intrinsic and extrinsic parameters. Next, since the spatial resolutions of the two cameras are very different, we compute the transformation of the depth by creating an intermediate 3D model from the Kinect data, which subsequently is projected to the DAVIS coordinate system.

In the very first step the RGB data and the Depth of the Kinect, which internally are captured by two separate sensors, are aligned to each other using the Kinect SDK. Next, the Kinect intrinsic and extrinsic sensor camera parameters are obtained using conventional image camera calibration on RGB data. Similarly, the DAVIS intrinsic and extrinsic camera parameters are obtained using conventional image camera calibration on the APS frames of the DAVIS (the APS frames and the DVS events in the DAVIS are geometrically calibrated). However, we note that the DVS event signal of the DAVIS, may also be calibrated by itself using a calibration grid of flashing LEDs (**Mueggler et al.**, 2015). Such a procedure can be used if only a DVS (but not a DAVIS) is available. We can use the procedure of (**Mueggler et al.**, 2015), which consists of two steps: first it adjusts the focus, then is computes the intrinsic parameters. The code is based on ROS, and the calibration uses OpenCV functions.

The second step involves first a stereo calibration between the RGB-D sensor and the DAVIS, which provides the rotation and translation of the two sensors with respect to each other. Then the depth between the two cameras is registered via a 3D world model. In detail, the procedure involves the following transformations.

First, the Kinect 2D image coordinates are compensated for radial distortion as:

$$\mathbf{x'} = \mathbf{x}(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \quad (1)$$

where k_1, k_2, k_3 are the radial distortion coefficients, $\mathbf{x}$ and $\mathbf{x'}$ are the distorted and undistorted image coordinates respectively, and $r = \|\mathbf{x}\|$.

Next the 3D world coordinates $\mathbf{X_w} = (\mathbf{x_w}, z_w)$ are obtained from the 2D image coordinates, $\mathbf{x'}$, as:

$$\begin{bmatrix} \mathbf{X_w} \\ z_w \end{bmatrix} = \begin{bmatrix} -(\mathbf{x'} - \mathbf{c}) z \frac{1}{\mathbf{f}} \\ z \end{bmatrix} \tag{2}$$

where $\mathbf{c}$ denotes the principal point, f the focal length of the Kinect camera, and z the depth.

The 3D point cloud is then transformed using the geometric transformation between the sensors, given by the 3×1 translation $\mathbf{t}$ and 3×3 rotation R obtained by the stereo calibration. The transformation is formulated as $\mathbf{X'_w} = R\mathbf{X_w} + \mathbf{T}$, where $\mathbf{X'_w}$ is the new point cloud model in the 3D world.

Lastly, the point cloud $\mathbf{X'_w}$ is projected onto the 2D sensor plane of the DAVIS to obtain the sensor coordinates $\mathbf{x_D}$ as:

$$\mathbf{x_D} = \mathbf{x_w} \frac{\mathbf{f_D}}{z_w} + \mathbf{c_D} \tag{3}$$

where $\mathbf{c_D}$ denotes the principal point and f_D the focal length of the DAVIS sensor. The depth for each image coordinate in the DAVIS image plane is registered using the Z-buffer. Any holes or ambiguities in the new registered depth are filled in using the inpainting method in (**Janoch et al.**, 2013), which assumes second order smoothness, minimizing the curvature in a least-squares manner. An example of the result of this calibration is shown in Fig. 3.

DAVIS SENSOR AND PTU CALIBRATION

This section explains how to obtain an analytic expression for the rotation R_α and translation T_α of the DAVIS sensor (in its coordinate system) corresponding to a pan or tilt angle α of the PTU. This is a non-trivial task. The procedure is as follows: We first derive the translation and rotation for a number of pan-tilt combinations with respect to a base pose (pan = 0°, tilt = 0°) in the DAVIS camera. Then, we use these derived values to compute the (fixed) transformation between the DAVIS coordinate system and the PTU coordinate system. The parameters involved are the translation $\mathbf{u}$ between the coordinate systems, the rotation axis $\mathbf{r}$ of the pan-tilt unit, and the rotation axis $\mathbf{s}$ of the camera (see Fig. 2).

First, we derive the translation and rotation of the DAVIS corresponding to various pan (rotation in the horizontal plane) and tilt (rotation in the vertical plane) combinations. In order to do that, we capture APS images with the DAVIS sensor for a number of pan tilt combinations, and perform a stereo calibration for each set of images with respect to the baseline (pan = 0° and tilt = 0°). We use as angle rotations for pan and tilt the values [-5°, -4°, -3°, -2°, -1°, 0°, 1°, 2°, 3°, 4°, 5°]. Since the transformation for pan and tilt can be applied independently, we do not need different combinations of pan and tilt. Thus, we have 11 pan combinations (0° tilt, including the base-pose, pan = 0° and tilt = 0°) and 10 tilt combinations (0° pan). For every combination, we take 10 images for the calibration, each with a different pose and position of the calibration pattern. The calibration provides the extrinsic rotation and translations of the DAVIS coordinate system with respect to the base-pose.

Let us now compute the *translation* of the DAVIS sensor center with respect to the PTU center. Consider the center of the coordinate system of the DAVIS for the baseline position O^D. The position of the coordinate center for a combination of pan and tilt $O^{D^{rt}}$ is described by a translation $\mathbf{t}$ with respect to the center of coordinates of the baseline O^D. This translation $\mathbf{t}$ corresponds to the extrinsic translation estimated in the calibration of a pan-tilt-combination with respect to the baseline (explained in the previous paragraph). The camera center O^D is described by a translation $\mathbf{u}$ with respect to the PTU coordinate

center, and a rotation R moves it to position $O^{D^{rt}}$ (see Fig. 2). Thus we have in the coordinate system of the PTU that:

$$O^{D^{rt}} = R \cdot \mathbf{u}$$
$$O^{D^{rt}} = \mathbf{u} + \mathbf{t} \tag{4}$$

Note that there are multiple combinations of pan and tilt rotations (for different angles θ), and thus multiple R and $\mathbf{t}$. The R for a specific angle θ can be re-written with respect to its axis $\mathbf{r}$ (in this case, only 2 variables), using the Rodrigues formula as:

$$R = (1 - cos(\theta))K^2 + sin(\theta)K + I \tag{5}$$

where $K^2 = \mathbf{r} \cdot \mathbf{r}^T - I$. Now, substituting R from Eq. (5) into the equality resulting by combining the two constraints of Eq. (4), and taking into account that the system has a total of N combination angles, the following minimization problem is formulated:

$$\underset{r,u}{\operatorname{argmin}} \sum_{i \in [1,\ldots,N]} \|((1 - cos(\theta_i))(\mathbf{r} \cdot \mathbf{r}^t - I) + sin(\theta_i)K) \cdot \mathbf{u} - \mathbf{t}_i\| \tag{6}$$

where the rotation axis is a unit vector, i.e. $\|\mathbf{r}\| = 1$.

The minimization with respect to the rotation axis $\mathbf{r}$ and the translation $\mathbf{u}$ is non-convex. However, the problem can be solved searching for the rotation axis and solving for the translation, using the interior-point method. Since the rotation axis has only two degrees of freedom, we use a change of variables to search over spherical coordinates as in (**Bitsakos**, 2010). The minimization cost for our stereo rig is shown in Fig. 4 where the minimum is marked on the sphere with a red star.

The second part computes the *rotation* axis $\mathbf{s}$ of the DAVIS sensor coordinate system. Since the rotation vectors derived for positive and negative angles of pan and tilt were found of nearly same value (but different sign), we did not formulate another minimization, but estimated the axis by taking the average of the values for the first two components. Using the fact that $\mathbf{s}$ is a unit vector provides the third value.

Finally, we obtain the following expression to compute for a given pan or tilt angle α the corresponding rotation R_α and translation T_α in the DAVIS sensor coordinates:

$$T_\alpha = ((1 - cos(\alpha))(\mathbf{r} \cdot \mathbf{r}^t - I) + sin(\alpha)K) \cdot \mathbf{u} \tag{7}$$
$$R_\alpha = (1 - cos(\alpha))L^2 + sin(\alpha)L + I \tag{8}$$

where $L^2 = \mathbf{s} \cdot \mathbf{s}^t - I$. Please note that the rotation and translation of the DAVIS coordinate system is applied independently to pan and tilt rotations, and we have two different rotations and translations for pan and tilt angles, respectively (denoted as θ and ϕ in Fig. 2).

Finally, the motion of the Pioneer 3DX Mobile Platform is always a translation in the horizontal plane in the direction of Z. For our case, we considered the coordinate centers of the Pioneer and the PTU to be aligned. Thus the translation of the mobile platform can be directly applied to the DAVIS sensor.

The code for the extrinsic and intrinsic calibration of the DAVIS and the RGB-D sensors, their stereo calibration, and the calibration between the DAVIS and the Pan-Tilt Unit is provided along with the dataset.

GENERATION OF MOTION FLOW FIELDS

The image motion flow field is the projection of the velocities of 3D scene points onto the image plane. Assuming a rigid motion (with translational velocity $\mathbf{t} = (t_1, t_2, t_3)$ and rotational velocity $\boldsymbol{w} =$

(w_1, w_2, w_3)), the 3D instantaneous motion $\dot{\mathbf{P}}$ of scene points $\mathbf{P} = (X, Y, Z)$ is given as $\dot{\mathbf{P}} = -\mathbf{t} - \boldsymbol{w} \times \mathbf{P}$ (**Longuet-Higgins and Prazdny**, 1980). Then the equations relating the velocity (u, v) at 2D image points (x, y) to the 3D translation and rotation and the depth Z amounts to:

$$u(x,y) = \tfrac{1}{Z}(-t_1 f + x t_3) + w_1 \tfrac{xy}{f} - w_2 \left(\tfrac{x^2}{f} + f\right) + w_3 y \tag{9}$$

$$v(x,y) = \tfrac{1}{Z}(-t_2 f + y t_3) + w_1 \left(\tfrac{y^2}{f} + f\right) - w_2 \tfrac{xy}{f} - w_3 x \tag{10}$$

4 EVALUATION METHODOLOGY

The metrics we use to evaluate event-driven methods are similar to the ones previously used for frame-based techniques. Image motion flow fields will be evaluated using the average endpoint error (**Baker et al.**, 2011; **Otte and Nagel**, 1994), which is defined as the average value of the vector distance between the estimated motion $\mathbf{u}$ and the ground-truth $\hat{\mathbf{u}}$, and is derived for N motion flow values as:

$$AEPE = \frac{1}{N} \sum_{i=1}^{N} \|\mathbf{u}_i - \hat{\mathbf{u}}_i\|. \tag{11}$$

Another representative metric, the average angular error (AAE), measures the average angular distance as:

$$AAE = \frac{1}{N} \sum_{i=1}^{N} \arccos \left(\frac{\hat{\mathbf{u}}_i^t \mathbf{u}_i}{\|\hat{\mathbf{u}}_i\| \|\mathbf{u}_i\|} \right). \tag{12}$$

We provide the code for computing the AEPE and AAE of a motion flow field. Similarly, we evaluate 3D camera motion (given by 3D rotation and translation vectors) as averages using the same measures, but in this case averaging over time.

In order to evaluate the robustness of motion flow field estimation, we provide the R*X* value (**Scharstein and Szeliski**, 2002), which measures the percentage of estimates with an error above *X*. So the larger the value, the worse the motion estimation. In the Middlebury (**Baker et al.**, 2011) evaluation, this measure is used with the endpoint error for R *0.5*, R *1.0*, and R *2.0*. To evaluate the significance of the computed measure, we also provide a statistical significance test. We use the Wilcoxon signed rank test (**Wilcoxon**, 1992), for which a *p-value* less than 0.05 shows statistical significance (see also (**Sun et al.**, 2014; **Roth and Black**, 2005)).

Different from frame-based flow, the flow from event-driven techniques is sparse. We also provide a measure for the sparseness of the estimation. The so-called density value expresses the percentage of motion estimates within a fixed time interval. In Computer Vision, although not common, optical flow density is considered in some works (see e.g. (**Barron et al.**, 1994; **Brandt**, 1997; **Barranco et al.**, 2012)).

5 DATASET EXAMPLES FOR DAVIS SENSOR MOUNTED ON THE ROBOTIC PLATFORM

We recorded more than 40 sequences of diverse scenarios, with the camera mounted on a Pan-Tilt unit on-board the Pioneer 3DX Mobile Platform. All the sequences are due to rigid 3D motions: pure pan or tilt motion, combined pan and tilt motion, translation of the robotic platform only (forward or backward translation), and combinations of pan, tilt and translation. The scenes are from an an office and have

a variety of objects of different sizes and shapes, such as chairs, tables, books, and trash bins. Texture was added to some of the objects to obtain a higher DVS event density. The depth is in the range of approximately 0.8 m - 4.5 m (also constrained by the use of Kinect), and the motion flow between frames (about 50 ms) is up to 5 - 10 pixels. There are a variety of rigid motions, including sequences that are mostly due to rotation, sequences that are mostly due to translation, and sequences with balanced rotation and translation.

Fig. 5 shows a few of the sequences. The first row shows the DAVIS images, the second the depth maps, and the third the motion flow fields (using the color-coding of (**Baker et al.**, 2011)). The first group of five images is from a pan and tilt motion, the last case on the top right and the first in the bottom left are from a pure zoom in and zoom out respectively. The last group at the bottom are from combined pan tilt and zoom in or zoom out motions, and the scenes are cluttered with objects of different shapes and at different depth ranges. The six parameters for the rotation and translation are shown below the figures. The complete dataset is available at the website.

6 CONCLUSIONS

We presented the first datasets for evaluating techniques of visual navigation with neuromorphic sensors. These datasets contain synthetic and real sequences of rigidly moving sensors in static environments. The data, which we provide, includes the images, the event streams, the 3D depth maps, and the 3D rigid motion of the sensor. Using these datasets one can evaluate and compare event-based and classic frame-based techniques of image motion estimation, 3D motion estimation, scene reconstruction and segmentation by depth. We also provide the code for the various calibration procedures used in order to facilitate future data collection and code for evaluation.

We plan to maintain the website, and add new more challenging sequences including a larger variation of scenes and dynamic scenes in the future. We also plan to evaluate and publish the results of different methods. So far we used the same evaluation metrics as in Computer Vision, which only address the accuracy of estimation. Since currently there are very few techniques available, the efficiency of computation on events has not been addressed yet. However, as new neuromorphic methods will be developed, and it becomes useful to evaluate and compare algorithms, we will also need to develop evaluation criteria aimed at the complexity of computation.

DISCLOSURE/CONFLICT-OF-INTEREST STATEMENT

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

ACKNOWLEDGMENTS

Thanks are due to Michael Stevens for helping with the Pioneer 3DX mobile platform. The authors thank the sensors research group at the Institute of Neuroinformatics in Zurich (ETH Zurich and University of Zurich), and IniLabs for their support.

FUNDING

This work was supported by an EU Marie Curie grant (FP7-PEOPLE-2012-IOF-33208), the EU Project Poeticon++ under the Cognitive Systems program, the National Science Foundation under grant SMA

1248056, grant SMA 1540917 and grant CNS 1544797, the Junta de Andalucia VITVIR project (P11-TIC-8120), and by DARPA through U.S. Army grant W911NF-14-1-0384.

REFERENCES

Badino, H., Huber, D., and Kanade, T. (2015), The CMU visual localization data set, http://3dvis.ri.cmu.edu/data-sets/localization/, accessed: 2015-11-01

Baker, S., Scharstein, D., Lewis, J. P., Roth, S., Black, M. J., and Szeliski, R. (2011), A database and evaluation methodology for optical flow, *Int. J. Computer Vision*, 92, 1, 1–31, doi:10.1007/s11263-010-0390-2

Barranco, F., Fermuller, C., and Aloimonos, Y. (2014), Contour motion estimation for asynchronous event-driven cameras, *Proceedings of the IEEE*, 102, 10, 1537–1556, doi:10.1109/JPROC.2014.2347207

Barranco, F., Fermuller, C., and Aloimonos, Y. (2015), Bio-inspired motion estimation with event-driven sensors, in Advances in Computational Intelligence (Springer), 309–321

Barranco, F., Tomasi, M., Diaz, J., Vanegas, M., and Ros, E. (2012), Parallel architecture for hierarchical optical flow estimation based on fpga, *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 20, 6, 1058–1067, doi:10.1109/TVLSI.2011.2145423

Barron, J. L., Fleet, D. J., and Beauchemin, S. S. (1994), Performance of optical flow techniques, *International Journal of Computer Vision*, 12, 43–77

Benosman, R., Clercq, C., Lagorce, X., Ieng, S.-H., and Bartolozzi, C. (2014), Event-based visual flow, *IEEE TNNLS*, 25, 2, 407–417, doi:10.1109/TNNLS.2013.2273537

Bitsakos, K. (2010), Towards segmentation into surfaces, Ph.D. thesis, Computer Science Department, University of Maryland, College Park

Brandli, C., Berner, R., Yang, M., Liu, S.-C., and Delbruck, T. (2014), A 240× 180 130 db 3 μs latency global shutter spatiotemporal vision sensor, *Solid-State Circuits, IEEE Journal of*, 49, 10, 2333–2341

Brandt, J. (1997), Improved accuracy in gradient-based optical flow estimation, *International Journal of Computer Vision*, 25, 1, 5–22, doi:10.1023/A:1007987001439

Butler, D. J., Wulff, J., Stanley, G. B., and Black, M. J. (2012), A naturalistic open source movie for optical flow evaluation, in A. Fitzgibbon et al. (Eds.), ed., European Conf. on Computer Vision (ECCV) (Springer-Verlag), Part IV, LNCS 7577, 611–625

Censi, A. and Scaramuzza, D. (2014), Low-latency event-based visual odometry, in Robotics and Automation (ICRA), 2014 IEEE International Conference on (IEEE), 703–710

Fleet, D. and Jepson, A. (1990), Computation of component image velocity from local phase information, *International Journal of Computer Vision*, 5, 1, 77–104, doi:10.1007/BF00056772

Geiger, A., Lenz, P., and Urtasun, R. (2012), Are we ready for autonomous driving? the kitti vision benchmark suite, in Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, 3354–3361, doi:10.1109/CVPR.2012.6248074

Janoch, A., Karayev, S., Jia, Y., Barron, J. T., Fritz, M., Saenko, K., et al. (2013), A category-level 3d object dataset: Putting the kinect to work, in Consumer Depth Cameras for Computer Vision (Springer), 141–165

Lichtsteiner, P., Posch, C., and Delbruck, T. (2008), A 128× 128 120 db 15 μs latency asynchronous temporal contrast vision sensor, *Solid-State Circuits, IEEE Journal of*, 43, 2, 566–576

Liu, S.-C., Delbruck, T., Indiveri, G., Whatley, A., and Douglas, R. (2015), Event-Based Neuromorphic Systems (John Wiley & Sons)

Longuet-Higgins, H. C. and Prazdny, K. (1980), The interpretation of a moving retinal image, *Proceedings of the Royal Society of London B: Biological Sciences*, 208, 1173, 385–397

Mac Aodha, O., Humayun, A., Pollefeys, M., and Brostow, G. J. (2013), Learning a confidence measure for optical flow, *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35, 5, 1107–1120

Mallik, U., Clapp, M., Choi, E., Cauwenberghs, G., and Etienne-Cummings, R. (2005), Temporal change threshold detection imager, in ISSCC. 2005 IEEE International Digest of Technical Papers. Solid-State Circuits Conference, 2005.

373 Mueggler, E., Huber, B., Longinotti, L., and Delbruck, T. (2015), ROS Driver and Calibration Tool for the
374 Dynamic Vision Sensor (DVS), https://github.com/uzh-rpg/rpg_dvs_ros, accessed:
375 2015-11-01
376 Orchard, G. and Etienne-Cummings, R. (2014), Bioinspired visual motion estimation, *Proceedings of the*
377 *IEEE*, 102, 10, 1520–1536
378 Otte, M. and Nagel, H.-H. (1994), Optical flow estimation: Advances and comparisons, in J.-O. Eklundh,
379 ed., Computer Vision ECCV '94, volume 800 of *Lecture Notes in Computer Science* (Springer Berlin
380 Heidelberg), 49–60, doi:10.1007/3-540-57956-7_5
381 Pérez-Carrasco, J. A., Zhao, B., Serrano, C., Acha, B., Serrano-Gotarredona, T., Chen, S., et al.
382 (2013), Mapping from frame-driven to frame-free event-driven vision systems by low-rate rate coding
383 and coincidence processing–application to feedforward convnets, *Pattern Analysis and Machine*
384 *Intelligence, IEEE Transactions on*, 35, 11, 2706–2719
385 Posch, C., Matolin, D., and Wohlgenannt, R. (2011), A qvga 143 db dynamic range frame-free pwm
386 image sensor with lossless pixel-level video compression and time-domain cds, *Solid-State Circuits,*
387 *IEEE Journal of*, 46, 1, 259–275
388 Raudies, F. and Neumann, H. (2009), An efficient linear method for the estimation of ego-motion from
389 optical flow, in Pattern Recognition (Springer), 11–20
390 Raudies, F. and Neumann, H. (2012), A review and evaluation of methods estimating ego-motion,
391 *Computer Vision and Image Understanding*, 116, 5, 606–633
392 Roth, S. and Black, M. (2005), On the spatial statistics of optical flow, in Computer Vision, 2005. ICCV
393 2005. IEEE International Conference on, volume 1, volume 1, 42–49 Vol. 1, doi:10.1109/ICCV.2005.
394 180
395 Ruckauer, B. and Delbruck, T. (2015), Evaluation of event-based algorithms for optical flow with ground-
396 truth from inertial measurement sensor, *Frontiers in Neuroscience*
397 Scharstein, D. and Szeliski, R. (2002), A taxonomy and evaluation of dense two-frame stereo
398 correspondence algorithms, *International Journal of Computer Vision*, 47, 1-3, 7–42, doi:10.1023/A:
399 1014573219977
400 Sturm, J., Engelhard, N., Endres, F., Burgard, W., and Cremers, D. (2012), A benchmark for the evaluation
401 of RGB-D SLAM systems, in Proc. of the International Conference on Intelligent Robot Systems
402 (IROS)
403 Sun, D., Roth, S., and Black, M. (2014), A quantitative analysis of current practices in optical flow
404 estimation and the principles behind them, *International Journal of Computer Vision*, 106, 2, 115–137,
405 doi:10.1007/s11263-013-0644-x
406 Tschechne, S., Sailer, R., and Neumann, H. (2014), Bio-inspired optic flow from event-based
407 neuromorphic sensor input, in Artificial N.N. in Pat. Rec., 171–182, doi:10.1007/978-3-319-11656-3_
408 16
409 Voigt, R., Nikolic, J., Hürzeler, C., Weiss, S., Kneip, L., and Siegwart, R. (2011), Robust embedded
410 egomotion estimation, in Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International
411 Conference on (IEEE), 2694–2699
412 Wilcoxon, F. (1992), Individual comparisons by ranking methods, in S. Kotz and N. Johnson, eds.,
413 Breakthroughs in Statistics (Springer New York), Springer Series in Statistics, 196–202, doi:10.1007/
414 978-1-4612-4380-9_16

FIGURES

Figure 1. Left: Pan-Tilt Unit FLIR PTU-46-17P70T at http://www.flir.com/mcs/view/?id=53707. Center: Pioneer 3DX Mobile Robot at http://www.mobilerobots.com/ResearchRobots/PioneerP3DX.aspx. Right: DAVIS240b sensor at http://inilabs.com

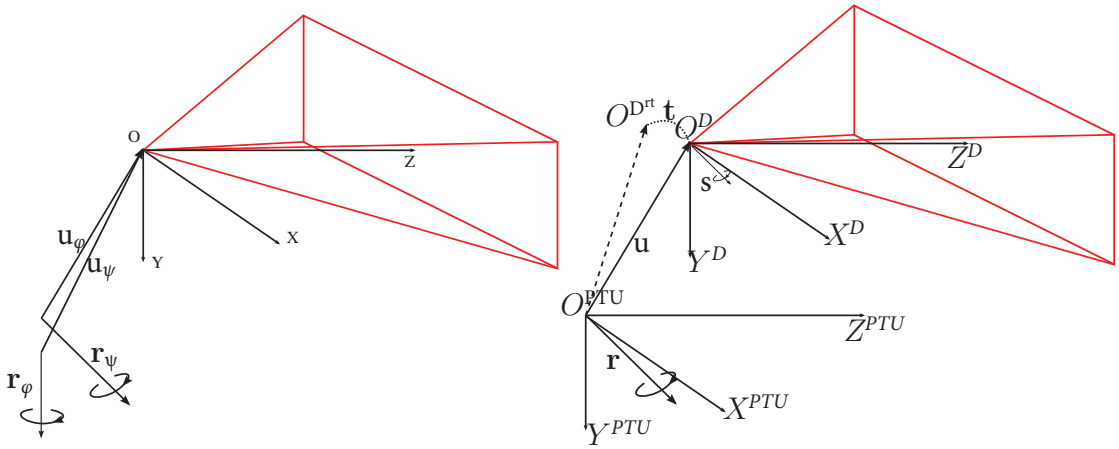

Figure 2. Left: Translation vector **u** of the DAVIS coordinate system with respect to the PTU, and **r**, the PTU rotation axis. The pose of the DAVIS sensor is represented by its axis **s**. Right: DAVIS coordinate system O^D and PTU coordinate system O^{PTU}. $O^{D^{rt}}$ represents the DAVIS coordinate system after a pan-tilt rotation of the PTU, characterized by a translation **t** and the rotation R around its axis **r**. Image adapted from (**Bitsakos**, 2010).

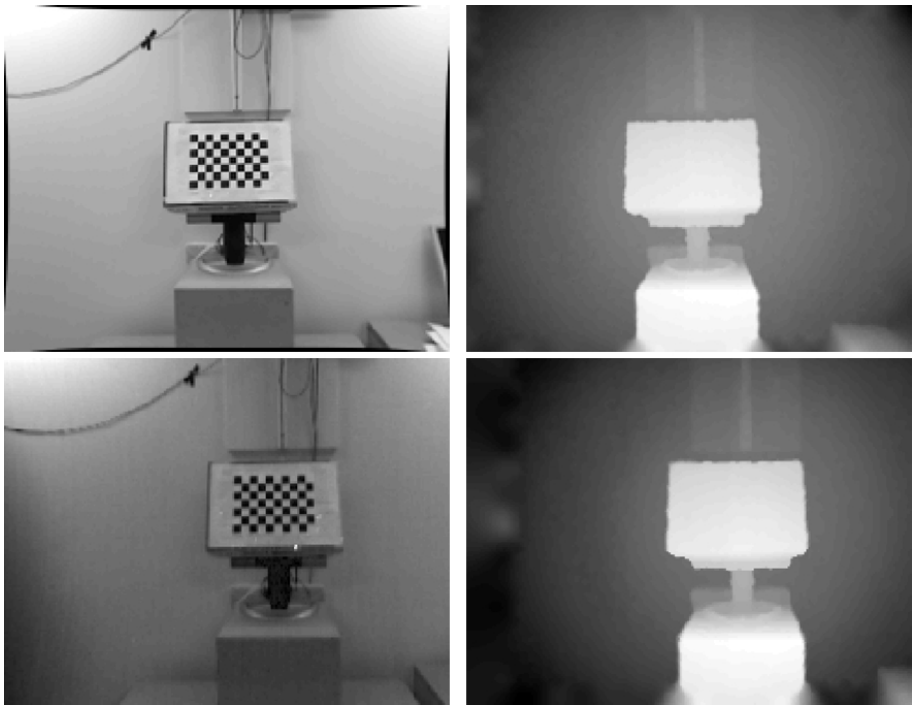

Figure 3. Depth registration from RGB-D sensor (top row) to DAVIS sensor (bottom row).

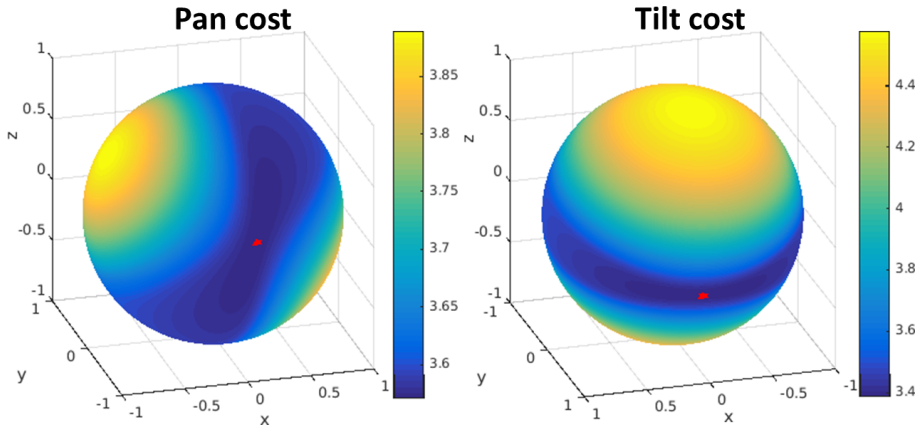

Figure 4. Visualization of the error function from the minimization for pan (left) and tilt (right). The minimum error is marked on the sphere with a red star. The search is done in spherical coordinates over the rotation axis **r**, which has two degrees of freedom. For each rotation we solve for the (best) translation.

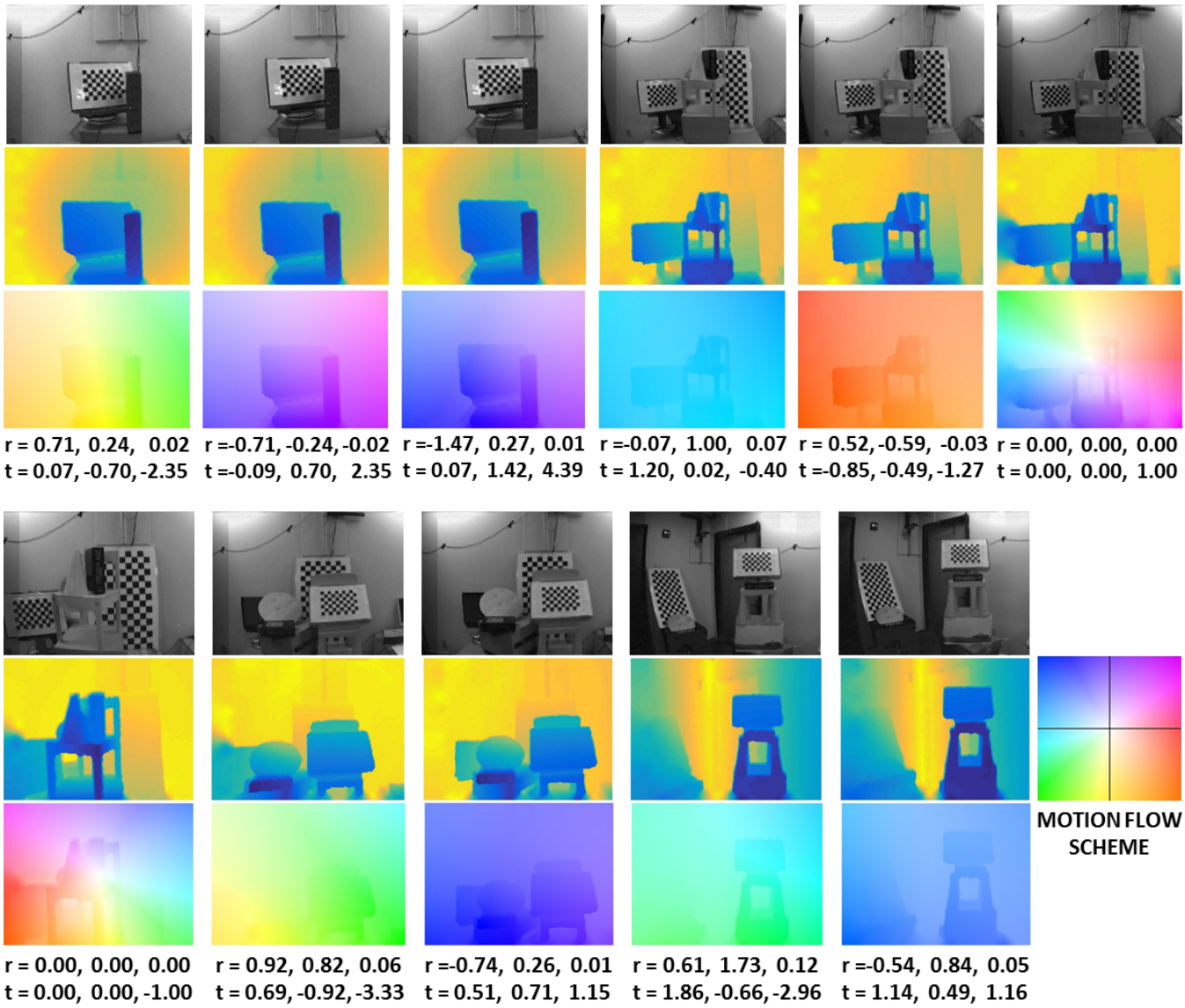

Figure 5. Example sequences from the dataset. For each sequence we show: DAVIS APS frame (first row), depth map (second row), motion flow field (third row), and the rotation and translation values (in 10^{-2} rad/frame and 10^{-2} pix/frame). The color coding for the depth map uses cold colors for near and warm colors for far points. The motion flow fields are color-coded as in (**Baker et al.**, 2011), with the hue representing the direction of motion vectors and the saturation their value.