# Low-latency localization by Active LEDs Markers tracking

Andrea Censi, Jonas Strubel, Christian Braend, Tobi Delbruck, Davide Scaramuzza

***Abstract*—TODO**

## I. Introduction

- We need fast sensors for fast robots
- In an architecture we can distinguish between latency and time-discretization.
- In this paper we consider the lowest-latency sensor available called DVS camera and how it can be incorporated in a robotic system.
- We create a tracking system with high-speed blinking LEDs mounted on the robot
- We could use this for relative localization between robots
- At this point the prototype of a DVS sensor is too big to be mounted on a robot

### A. DVS camera

low latency **[boerlin09getting][lichtsteiner08asynchronous][etienne-cummings99intelligent][oster08quantification]**

- The main disadvantages are:
  - prototype status
  - heavy
  - low-resolution
  - buffer, too many events
  - complicated to tune
- Future improvements:
  - more resolution
  - inclusion of a "normal" RGB camera
  - miniaturization

### B. Paper outline

- Section ???
- Software and datasets are available at ???

## II. Hardware setup and event data

- This section describes the basic hardware setup and how the data looks like.

### A. Active LED Markers (ALMs)

- We have a set of blinking LEDs
- Each LEDs blinks at a different frequency.

Figure II.1.

Figure II.2.

### B. Events

### C. Events data in practice

- Fig. II.1 shows how the data looks like.
  - In this case, the LEDs are fixed in the environment and a *fixed* camera is looking at them.
  - Fig. II.1*a* shows the histogram of events from one pixel
  - Fig. II.1*b* shows the sequence of events from one particular pixel.
- Note also the halo in Fig. II.1*a* cannot be explained by the refractive properties of the optics of the camera and is probably due to properties.
- The idea
- Experimentally the interval is actually very repeatable

### D. Alternate events and motion

- We go from events to "alternate events"
- This needs a buffer
- This series now has the polarity
- Fig. II.3*a* shows the histogram
- The frequency peaks are clearly visible in this histogram
- What about motion?
  - We see that, following motion, in Fig. II.3*b* the peaks are clearly visible.
  - There is also a
  - Of course all of this depends on the statistics of the image.

## III. Tracking algorithm

### A. From raw events to sequence events

### B. Particle filters

### C. Estimation

### D. 3D Reconstruction

*1) :*

Figure II.3.

## IV. Models of nontraditional cameras

Our first goal is to relate

We want to see whether it is still possible to achieve the same tasks.

**Definition 1** (Time and space-continuous vision sensor)**.** Let $y(q, s, t)$, with $s \in \mathbb{S}^2$ be the signal of an ideal light field sensor, at time $t$, pose $q$, in direction $s$.

**Definition 2** (Time and space-discretized vision sensor)**.** Let $y(q, s, t)$, with $s \in \mathbb{S}^2$ be the signal of an ideal light field sensor, at time $t$, pose $q$, in direction $s$.

Snapshot-based sensor

Time-continuous, space-discretized vision sensor

$$\dot{y} = \mu(s) \left( \nabla_i y(s,t) \right) v^i$$

## V. Experiments

The goal of our experimental evaluation is to consider the advantages of a DVS-based tracking solution compared with a tracking solution based on a traditional CMOS camera.

We compare the DVS+LED-based tracking with vision-based tracking using the PTAM algorithm, using the output of an OptiTrack system as the ground truth.

The data show that the DVS+LED-based solution is able to deal with faster motions due to the minimal latency, however, the reconstructed quadrotor pose is not as accurate, as it could be expected from the lower resolution.

### A. Setup

*1) Robot platform:* We used the commercially available ARDrone 2.0, a remarkable platform for its low price of ???. We attached four custom-built ALMs (V.1). Each LED was fixed facing downwards, one under each of the four rotors, so that the four where lying on a plane forming a square of 20cm side length. The USB connector available on the drone provided power to the microcontroller and ALMs.

The drone has also a front-facing ??? CMOS camera that is used in these experiments. The ground-facing camera is not used.

*2) DVS :* The DVS128 camera was used for the tests. This model is currently commercially available from INI labs. It has a resolution of $128 \times 128$ pixels. The lens attached was a ???, with a FOV of ???, giving approximately ??? degrees per pixels resolution.

For tracking the quadcopter, the DVS was installed on the floor facing upwards. Note that the relative motion between DVS and quadcopter would be the same if the positions were switched (ALMs on the floor or another vehicle and DVS on the platform).

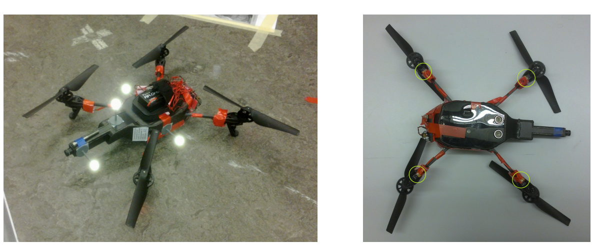
*3) OptiTrack:* To measure the pose estimation accuracy we used a OptiTrack tracking system from NaturalPoint[1], which is a marker-based optical motion tracking system using active infrared light and reflective marker balls.

Our lab setup comprised ??? cameras in an $6\,\mathrm{m} \times 8\,\mathrm{m}$ area; the cost of this system is approximately ???. Four markers have been applied to the drone (Fig. V.1a). We estimate the accuracy of this system in our lab conditions to be approximately ??? mm.

*a) Motion:* The prototypical aggressive maneuver that we use is a "flip" of the quadcopter, i.e. a 360° roll (Fig. V.2a). This happens in approximately ??? seconds. During the flip the frontal camera images are severely blurred (Fig. V.2b).

*b) Interference OptiTrack / DVS:* We encountered an unexpected incompatibility between OptiTrack and DVS camera. The OptiTrack uses high-power infrared spotlights. In the OptiTracks standard configuration, the spotlights are pulsed at a high frequency. This is of course invisible to normal sensors and to the human eye, but it was a spectacular interference for the DVS. Like most cameras, the DVS is very sensitive to the infrared spectrum and is much faster than the OptiTrack strobing frequency. Strong generated a buffer overflow on the DVS as the electronics could not handle the large number of events to be processed contemporaneously. Eventually we understood how to deactivate the strobing for all the cameras prior to recording. Still there was a slight residual interference by the infrared illumination from the OptiTrack, but it should have relatively little impact to the results of our experiments.

[1] http://www.naturalpoint.com/optitrack/



(a) Infrared markers (b) ALMs configuration

Figure V.1. The ARDrone 2.0 equipped with reflecting markers for the OptiTrack (shown in *a*) and the LEDs tracked by the DVS (shown in *b*).

### B. Methods

We compare three ways to track the pose of the quadcopter: 1) The output of our method; 2) The OptiTrack output; 3) The output of a traditional feature-based tracker using the data from the conventional CMOS camera mounted front-facing on the drone. The image data was streamed to a computer via network interface, were the parallel tracking and mapping algorithm (PTAM)**[PTAM]** was employed for pose estimation.

*1) Data recording, synchronization, and alignment:* Using this setup we did several recordings, in which we recorded the OptiTrack tracking data, using its native format, the image data using a ROS interface, as well as the raw event data from the DVS in the native format. All these data, plus other data used for preliminary experiments, are available at the website ???.

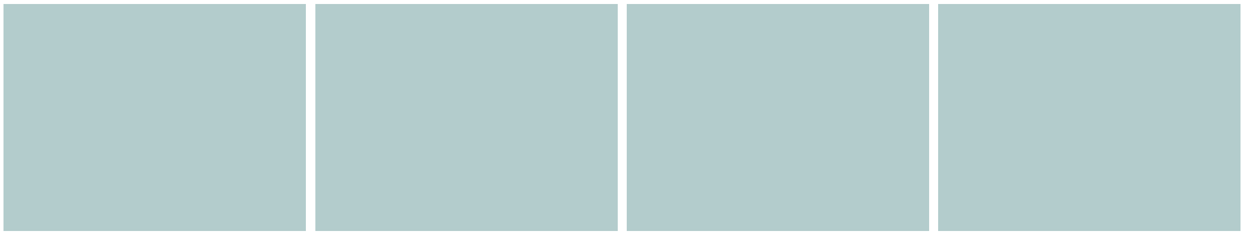
To synchronize the data from different sources we used a motion induced cue. We moved manually the drone up and down, generating an approximated sinusoid curve in the position data. Matching this trend in the data allowed easy synchronization of the various recordings.

After adjusting for the delay, the data sets were brought to the same number of samples with a common time stamps. As our algorithm's output has a lower sampling rate than the OptiTrack (1KhZ vs ???), the OptiTrack data was resampled by linear interpolation.

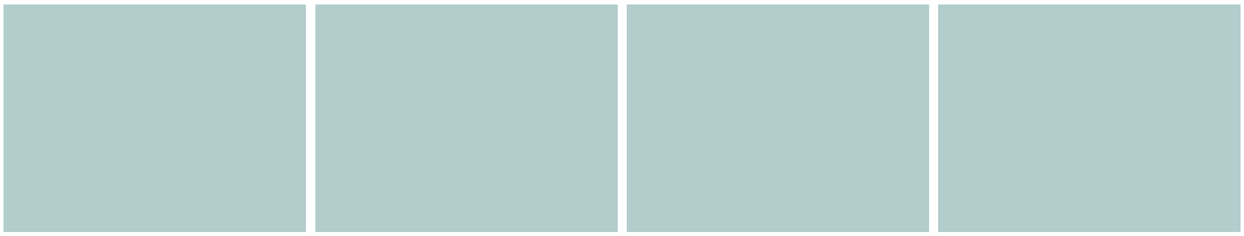
After time synchronization we put all data in the same frame of reference. Given two sequences of points $\boldsymbol{x}_k, \boldsymbol{y}_k \in \mathbb{R}^3$, the rototranslation $\langle \boldsymbol{R}, \boldsymbol{t} \rangle \in \mathsf{SE}(3)$ that matches them can be found by solving the optimization problem

$$\min_{\langle \boldsymbol{R}, \boldsymbol{t} \rangle \in \mathsf{SE}(3)} \sum_k \| \boldsymbol{x}_k - (\boldsymbol{R}\boldsymbol{y}_k + \boldsymbol{t}) \|^2, \tag{V.1}$$

which can be easily solved using ???.



(a) Flip motion seen from outside



(b) Flip motion seen from the frontal camera

Figure V.2. Flip motion

### C. Results

We recorded data from 18 flips, of which only 6 were succesful. During the recordings we met a number of unforeseen difficulties due to our modifications to the drone. Having attached the LEDs and microcontroller to the drone we found that it had become unstable during flight and hard to control due to the additional weight, so while it could hover normally, it did not have enough thrust to stabilize itself after a flip.

*1) Tracking downtimes:* During a flip, both the DVS and PTAM lose tracking: PTAM loses tracking while the image is blurred; the DVS loses track when the ALMs are not visible from the ground. The comparison of these "blackout times" gives a direct measurement of the latency of the two systems.

The length of a flip was measured by considering the roll data from the OptiTrack, taking the interval between the last measurement before the flip and the first measurement after the flip when the helicopter was in a level orientation to the floor.

To measure the onset and offset of the blackout for the DVS, we considered the last sample before losing track (i.e. where the interval position samples were considerably higher than the mean sampling rate) and the first sample of reacquiring track (regaining a steady sample rate). The equivalent operation was performed on the PTAM data.

Figure V.3 shows a statistical comparison of the blackout time intervals for the two approaches compared to the duration of flips. The red bar indicates the median while the blue box marks the first and third quartile. The whiskers extend to a range of 1.5 times the range between the first and third quartiles. All data points outside this range are considered outliers and are marked with a red plus.

Table I shows the mean standard deviation of the different approaches. Our algorithm lost track during the average time of 0.35 seconds. PTAM lost track for a mean of 0.8 seconds, which is more than twice the time of the DVS and takes longer than the average duration of a flip.

Table I
MEAN AND STANDARD DEVIATION OF TRACKING DOWNTIME INTERVALS AND THE FLIP DURATION.

| | mean [s] | std.dev. [s] |
|---|---|---|
| DVS | 0.35 | 0.10 |
| PTAM | 0.80 | 0.33 |
| flip motion | 0.56 | 0.15 |

One can clearly see that the time where tracking is lost is much shorter with our approach in respect to PTAM. As Figure V.4 further illustrates, the downtime for the DVS stays inside the interval of the flip duration. The results emphasize that the DVS is faster in recovering lost tracks than the PTAM approach due to the shorter latency. As verified with our recordings, the downtimes of the DVS correspond to losing sight of the LED markers because of their emission angle.

We reckon that with a suitable configuration of either more markers or dynamic vision sensors, tracking could be maintained during the whole flip. PTAM shows to lose track for a longer duration than the flip takes. In contrary to the DVS the the camera of the drone loses sight of its tracked features due to blurring in the camera image and thus takes a longer time to recover.

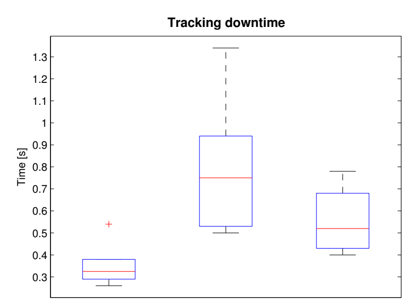



Figure V.3. Statistical plot of measure time interval. The boxplots show the time interval in which tracking is lost for the our algorithm and PTAM compared to the duration of a flip.

Table II
ESTIMATION ERROR OF DVS AND PTAM COMPARED TO OPTITRACK

(A) TRANSLATION

| | mean [cm] | std.dev. [cm] |
|---|---|---|
| DVS | 8.9 | 12.6 |
| PTAM | 19.0 | 12.4 |

(B) ROLL

| | mean [°] | std.dev. [°] |
|---|---|---|
| DVS | 19 | 27 |
| PTAM | 7 | 22 |

(C) PITCH

| | mean [°] | std.dev. [°] |
|---|---|---|
| DVS | 17 | 18 |
| PTAM | 5 | 11 |

(D) YAW

| | mean [°] | std.dev. [°] |
|---|---|---|
| DVS | 6 | 15 |
| PTAM | 3 | 10 |

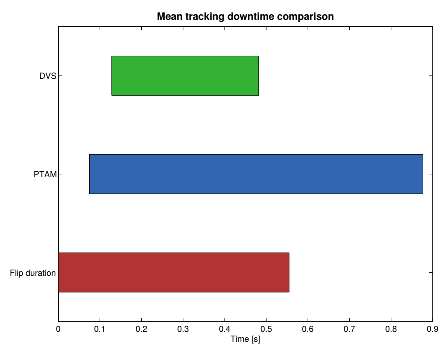
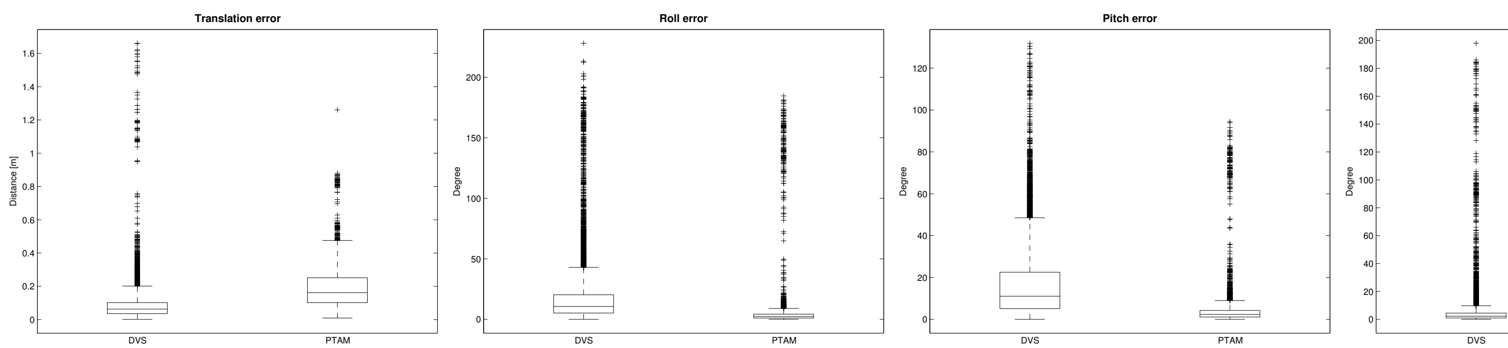



Figure V.4. Comparison of the mean tracking downtime intervals. The mean time intervals of both algorithms are compared against the mean flip time on a time line.

*2) Pose estimation:* Table IIIa shows the mean and standard error for the translation in both approaches. The DVS average error is roughly two times lower than PTAM. V.5a shows the statistical distribution of the pose errors. Although the spread of outliers is higher in our approach compared to PTAM, the translation errors of the latter technique show a broader distribution around their median. Overall this proves that the DVS approach has higher accuracy with less spread, if we neglect the extreme tails of the distribution.

Figures V.5b, V.5c and V.5d show the error distribution for roll, pitch and yaw respectively. The DVS performs worse in roll and pitch compared to yaw. This was to be expected, because of the position of the ALMs. As roll and pitch play a minor roll in qaudrotor pose estimation these can be neglected for finding the drone's orientaion. Table IIId demonstrates that the DVS performs slightly worse than PTAM with a mean error of 6 degrees and a deviation of 15 degrees. This is explained by the much lower resolution of the DVS ($128 \times 128$) compared to the PTAM data (???).




(a) Translation error (b) Roll (c) Pitch

Figure V.5. Distributions of the errors of DVS/PTAM in reference to the OptiTrack measurements. The data is synthetized in Table II.

## VI. Conclusions

Fast robots need fast sensors. A DVS

*3) Future work:*

- other factors that impact performance
- Mounting on another

Using the DVS for tracking has some clear advantages as the high sampling speed enables the tracking of LEDs pulsed with frequencies above a 1000 Hz. Compared to normal high speed cameras, the data output and thus the processing is reduced, as only change is advertised by the camera. This makes the DVS also interesting for embedded processing.

The measurements revealed that the DVS is able to reacquire stable tracking after as the quadrotor flips with negligible delay as soon as the LEDs are visible again. In comparison to the PTAM used by the quadrotor, it is more than twice as fast. The DVS thus has a clear advantage in comparison to conventional cameras as it does not suffer from blurring. Nevertheless, we used active markers in our experiment which are easier to identify than image features. Due to the fact that the DVS not only needs motion in order to produce input but also has a lower resolution than conventional imaging sensors it is not clear yet, how well features from the environment can be used as a tracking reference. The pose estimation has proven to be more accurate than PTAM and should thus perform well in helicopter navigation. Nevertheless, the low resolution of the DVS limits the range in which a helicopter can be tracked robustly. As there are possible improvements to our algorithm in terms of stability and robustness, we reckon that is should be possible to even improve the pose estimation accuracy on the DVS. A possible approach could include considering not only local maxima from single pixels, but averaging over several pixels activated by an LED. Alternatively another particle filter could be used to smoothen the position readings over time. This approach would also help with the occasional appearance of misdetections. On the hardware side we found a couple of improvements as well. For further experiments reliable drone control is important. While reducing or better balancing the payload could have an impact, we were also not sure if the highly used rotor blades or other parts of the quadcopter were responsible for the unstable flight. As recalibration of the internals seems unfeasible one might also consider using a different drone in the future. To improve visibility of the markers, LEDs with higher angle and power output would be beneficial. This requires some additional hardware though which needs to be considered again in terms of payload.

Apart from a number of possible improvements, the approach has shown to be feasible and has demonstrated the advantages of using and event-driven approach for vision based robotics. As this utilization of asynchronous vision is a rather novel approach in robotics it would be interesting to use this camera in other navigational tasks in future projects, for example for visual SLAM on autonomous ground vehicles. While small image features as used in SIFT [**SIFT**] might be difficult to use due to resolution line feature extraction, as describe in [**LineTracking**], could be a feasible approach. Additionally, as edges are the natural source for DVS events it would have an advantage in terms of processing compared to normal cameras.

**Acknowledgements**

- Christian Saner
- Yves