Event-based RGB sensing with structured light

Seyed Ehsan Marjani Bajestani  Giovanni Beltrame
Polytechnique Montreal
{ehsan.marjani,giovanni.beltrame}@polymtl.ca

Abstract

Event-based cameras (ECs) are bio-inspired sensors that asynchronously report pixel brightness changes. Due to their high dynamic range, pixel bandwidth, temporal resolution, low power consumption, and computational simplicity, they are beneficial for vision-based projects in challenging lighting conditions and they can detect fast movements with their microsecond response time. The first generation of ECs are monochrome, but color data is very useful and sometimes essential for certain vision-based applications. The latest technology enables manufacturers to build color ECs, trading off the size of the sensor and substantially reducing the resolution compared to monochrome models, despite having the same bandwidth. In addition, ECs only detect changes in light and do not show static or slowly moving objects. We introduce a method to detect full RGB events using a monochrome EC aided by a structured light projector. The projector emits rapidly changing RGB patterns of light beams on the scene, the reflection of which is captured by the EC. We combine the benefits of ECs and projection-based techniques and allow depth and color detection of static or moving objects with a commercial TI LightCrafter 4500 projector and a monocular monochrome EC, paving the way for frameless RGB-D sensing applications. Our code is available publicly: github.com/MISTLab/event_based_rgbd_ros

1. Introduction

Event-based cameras (ECs) report pixel brightness changes asynchronously, a behavior inspired by the human eye [10]. When the brightness changes over a certain threshold for a pixel, the camera generates an event containing the coordinates of the pixel \((x,y)\), a timestamp, and the polarity of the event (i.e. increasing or decreasing). Although ECs do not capture full images, they can detect movement thousands of times faster than standard frame-based sensors, and since they do not have an external shutter cycle, their output is event-driven and frameless, resulting in very low latency, power, and bandwidth demands.

ECs have been used in various computer vision applications such as fast movement detection and tracking [2, 22, 1], optical flow, pose tracking and visual-inertial odometry [52, 26], Simultaneous Localization And Mapping (SLAM) [37, 40], pattern recognition [44], depth estimation and stereo vision [51, 36, 45], and many more.

In computer vision, color information has an important role [47] and could be essential to many tasks such as segmentation and recognition [19]. The first generation of ECs are monochromatic, with color ECs only recently becoming available [18, 46, 24, 23]. However, due to limitations in terms of sensor size, color ECs have lower resolution than mono ECs because they need to use color filters.

It is worth noting that ECs report pixel brightness changes, meaning that an EC will not report anything when the camera (and/or the object in its field of view) is static or slowly moving (Fig. 1, top left), which can be critical
in some cases (e.g. for a slow-moving robot). To overcome this issue, one could use an external active device such as a laser, a flashing LED, or a light projector to generate events in static and almost static situations. This external active lighting system could also be used to detect depth by projecting detectable patterns called Structured Light (SL) [5, 20, 28, 27].

We present a method to add color and depth to a monocular, monochrome event-based camera while maintaining fast response time and resolution. We use a Digital Light Processing (DLP) projector that emits patterns of lights that we call Active Structured Light (ASL) on a scene, the reflection of which is captured by the EC which in turn generates events tagged with the color and depth of the scene. It is worth noting that our ASL method could also be used with color ECs, allowing the detection of static scenes. By dynamically adjusting the projection, we have color data when needed, managing the overall bandwidth of the system. For example, we can use the full resolution of the camera to detect static color scenes, or use more sparse patterns for fast moving objects. Projecting patterns also allows triangulation-based measurements to create a colorful 3D point cloud of the scene. Overall, our method generates colorful events from a monochrome EC:

1. with no loss of spatial resolution;
2. with the ability to detect static objects and scenes;
3. optimizing the bandwidth of the EC by detecting the color when and where it is needed;
4. using patterns that allow event-based depth measurement, ultimately generating colorful point clouds.

In this work we focused on visual light wavelength (emitted by the LED projector) and materials that are not in the category of fluorescence and they do not change the wavelength of the light. We validated our approach in different dynamic conditions: Fig. 2 shows the experimental setup with a DLP projector\(^1\) and a Prophesee evaluation kit\(^2\). With this setup, we achieved full color detection at an equivalent rate of 1400 frames per second (fps) (note that the camera is frameless, we use fps just for the purpose of comparison). Fig. 2 also shows the color detection of a static printed color wheel.

The rest of the paper is as follows: Section 2 presents related work; Section 3 describes our method for color detection; Section 4 details the results of our method in several conditions; and finally, Section 5 draws some concluding remarks and outlines possible future work.

2. Related Work

Digital color cameras use various Color Filter Array (CFA) or Color Filter Mosaic (CFM) on their sensors to detect different colors for each pixel, and among them, the Bayer array filter [3] is the most common CFA [35]. The size of a CFA is between 4 to 36 pixels (or sometimes larger [15]), which means that we need several monochrome pixels to generate each color pixel, effectively decreasing the resolution (e.g., 4x with a 2 × 2 CFA).

Colorization is the process to generate a color image based on a monochrome sensor or grayscale image without loss of resolution. Colorization requires either external data about the image colors, user interaction, or a trained neural network embedding the knowledge of the colors on the scene, and can be a time-consuming and expensive task [16]. Levin et al. [16] introduced a method that needs a few initial inputs from a user to generate a full color image and keeps tracking the color on upcoming frames in a video. Zhang et al. [50] introduce a fully automatic colorization approach based on a convolutional neural network (CNN) that can change a grayscale image into a near-real colorful image. Their method successfully deceived 32% of human participants in distinguishing the generated and ground-truth images. In contrast to these colorization approaches, our method does not need initial input data to get color out of a monochrome camera and it can provide realistic color information faster than CNN models.

Another approach to generate color data without quality loss on a monochrome image is to use separate cameras: the monochrome sensor takes a more detailed and higher contrast image, while a lower resolution RGB camera adds color information. This combination is common, but the image fusion, colorization, or the color transfer process are still a challenge [14].

Event-based cameras have introduced a new field of imaging systems. Due to their advantages compared to standard cameras, many scientists investigated ways to generate and reconstruct images from events to use in frame-based computer vision algorithms. A monochrome EC has been
used in many image reconstruction works [29, 41, 38, 42, 33, 13]. Also, combining a standard frame-based camera and an EC can produce a deblurred high frame rate (HFR) and high dynamic range (HDR) video [21].

By combining three ECs using dichroic filters Marcireau et al. [19] introduced a prototype to capture a stream of events in RGB separate channels for color segmentation. This method maintains the monochrome resolution but increases the bandwidth 3x.

With the introduction of color event-based cameras [24], some research focused on the reconstruction of images and videos based on color events [39, 32, 25]. Scheerlinck et al. [43] presented a dataset for color ECs. They also compared the output quality of some image reconstruction methods such as [41, 29, 38] in color.

As digital color cameras, current color ECs also use CFA to generate color events, which reduces their output resolution leading to lower bandwidth when compared with Marcireau et al. [19]. Our method reconstructs color data when needed, keeping the bandwidth of the system in check.

3. Monochrome to color

Compared to frame-based cameras, ECs are faster sensors, however, since they report nothing in a static situation or with slowly moving objects, they require an additional sensor to provide visual perception in these situations. We use an external event generator, namely a DLP projector. By emitting a pattern of light on objects in the scene, not only we can detect their color, but we are also able to detect depth, which makes event-based RGB-D sensing possible. Moreover, since ECs have high dynamic range, a high-power light projector is not necessary in dark environments.

There are many standard color formats for digital color descriptions (additive or subtractive), such as CMY (cyan, magenta, yellow), or with black CMYK, RYB (red, yellow, blue), RGB (red, green, blue) or with white RGBW and etc. Selecting the color space could depend on the application and the color range of the desired objects in the environment. Without loss of generality, we select the RGB color space which is more common in vision applications.

We use the EC to measure the amount of reflection of the emitted light on an object. To measure the color, we project three different wavelengths (structured light in red, green, and blue) on the environment and measure the amount of reflected light captured by the EC. During each pattern exposure time, the received events are gathered in an appropriate color channel on the initial frame.

To synchronize the DLP projector with the EC, we connect the trigger pins of the camera to the projector. By changing the pattern color, the DLP sends a pulse to the camera which identifies the incoming events as belonging to the appropriate color channel. Fig. 3 depicts the output of the color detection of a printed RGB color wheel separated in each color channel. The bottom frame of the Fig. 3 shows that the printed color wheel does not have pure green (0, 255, 0) and blue (0, 0, 255) colors in 24 bit RGB format. For example, in the red light channel (bottom left), the green circle also reflected some light (although less than the red circle) and as a result, it appears gray.

3.1. Color detection speed limits

One of the main advantages of the ECs is their response time which is in the range of microseconds. However, with the introduced method, we need to gather events of each color separately, limiting the speed of color detection to the maximum speed of pattern switching of the DLP projector. With the LightCrafter 4500 Evaluation Module, we are able to detect color with an equivalent frame rate up to 1400 fps due to its high frequency (4225 Hz).

However, assuming that the color of the object is not changing, we could still use the other methods to track the object only based on the high speed stream of events [22, 1] and use the color detection method for a short period of time. Fig. 4 shows the output of the color detection of a spinning colorful paper pinwheel reconstructed at different frame rates.

3.2. Advantages over monochrome cameras

Monochrome or grayscale cameras have been used in vision-based applications that do not need color information. As mentioned in Section 2, the combination of a monochrome camera and a color camera could be chal-
environment compared with a low-resolution EC [11], our high-resolution EC could be subject to more noise in a dark which makes the system more reliable. Further, since a mechanical parts to move the camera and receive events, static robot or camera, meaning there is no need to have when and where it is needed. Moreover, our method also efficiently use the bandwidth by detecting the color only is static or moving very slowly. Our method is useful to cannot detect the environment when the camera or object

Higher bandwidth requirements can cause bus saturation (as different channels, which increase the need for bandwidth. Our method allows us to benefit from ECs’ features and detect/update the color information for a given period of time. Moreover, the camera-projector combination enables depth sensing and simplifies feature detection and matching (w.r.t. stereo cameras) [5, 20, 28, 27].

3.3. Advantages over color event-based cameras

As mentioned in Section 2, digital cameras often use CFA to detect color. For instance, the Color-DAVIS346 [46] is one of the most recent color EC that uses an RGBG Bayer pattern with an output resolution of 346 × 260 pixels. This kind of camera is reporting the stream of events in 3 or 4 different channels, which increase the need for bandwidth. Higher bandwidth requirements can cause bus saturation (as described in Section 4). In addition, despite increasing the bandwidth needs and decreasing the resolution, color ECs cannot detect the environment when the camera or object is static or moving very slowly. Our method is useful to efficiently use the bandwidth by detecting the color only when and where it is needed. Moreover, our method also gathers information from the environment from an initially static robot or camera, meaning there is no need to have mechanical parts to move the camera and receive events, which makes the system more reliable. Further, since a high-resolution EC could be subject to more noise in a dark environment compared with a low-resolution EC [11], our method could still get the benefits of the high-resolution monochrome EC in a dark environment.

3.4. White balance and color correction

White balance and color correction can make the captured image close to its natural color. White balance can be adjusted before or after capturing the image. Generating white light with the DLP projector can change the image white balance, because the color temperature of a light source or the warmth/coolness of the white light can change the white balance directly. The DLP projector has three different LED colors: red, green, and blue. Generating LED-based white light could be challenging with wideband wavelength RGB LEDs [31, 30, 7]. Since the DLP projector has narrowband LEDs, the white balance can be adjusted by changing the current of each LED separately.

Lighting model: If we consider the DLP projector as a point-sized light source, we can model the lighting with the Lambertian shading model which is one of the simplest bidirectional reflectance distribution functions (BRDF) and an appropriate approximation to many real-world material surfaces [34]. In the Lambertian shading model, R, G, B values of the resulting pixel are independent of the angle that the viewing ray hits the surface:

\[ W = S_{ref} S_{pow max}(0, n \cdot b), \]

where \( W \) is the combination of \((R, G, B)\) values for a desired pixel, and \( S_{ref} \) is the spectral reflectance of the material, \( S_{pow} \) represents the spectral power distribution of the projector (as the light source), \( n \) is the outward surface normal (of the object) and \( b \) is the light beam vector which is from the surface intersection point to the projector. The dot product of these two unit vectors gives the amount of attenuation based on the angle between the surface to the projector. The \( max \) function is used to prevent a condition where \( n \cdot b < 0 \), because the projector would be behind the object in this case. This model could be divided for each color, for example the model for red light is:

\[ R = S_{ref R} S_{pow R max}(0, n \cdot b) \]

To generate white light in an ideal situation, we consider that each color has the same power distribution and \( S_{pow R} = S_{pow G} = S_{pow B} \). And for a white or gray surface we would have \( S_{ref R} = S_{ref G} = S_{ref F} \). As a result, by controlling the current of each LED \( S_{pow} \) we can have balanced white light.

We can use a gray card, a color wheel, or a Macbeth color chart/color checker to calibrate our system. We used a printed Macbeth color chart to do the calibration, and Fig. 5 shows the output of the color detection with the proposed method with and without white balance calibration.

Absolute error: To check the quality of the reconstructed image, we need to have a base image and specify
an error calculation method. We consider the captured image by a frame-based high-resolution camera as the base image (Ground Truth or GT) in Fig. 5. To calculate the absolute error, we compared the histogram of two images in Hue Saturation Value (HSV) format based on the correlation metric\(^1\):

\[
c = d(H_a, H_b) = \frac{\sum_{i} (H_a(i) - \bar{H}_a)(H_b(i) - \bar{H}_b)}{\sqrt{\sum_{i} (H_a(i) - \bar{H}_a)^2 \sum_{i} (H_b(i) - \bar{H}_b)^2}}
\]

(3)

where

\[
\bar{H}_k = \frac{1}{N} \sum_{j} H_k(j),
\]

(4)

and \(N\) is the total number of histogram bins, which in our case is 256 (8 bit in each color channel). The \(H_a\) and \(H_b\) are respectively histogram of the output image and the baseline image with a Histogram Correlation (HC) between 0 and 1. The \(HC\) between the base image (left) and each reconstructed image is respectively 0.22 and 0.76 for the reconstructed image without white balance (middle) and with white balance (right) in Fig. 5. Moreover, to check the difference between each pixel in the reconstructed image and the GT, and calculate the absolute error, we calculated the root mean square error (RMSE) separately for each channel. As an example, the RMSE for the red channel is:

\[
RMSE_{red} = \sqrt{\frac{\sum_{j} (p_{a,j} - p_{b,j})^2}{N}}
\]

(5)

where \(p_{a,j}\) and \(p_{b,j}\) are respectively the pixel value in the red channel of the output frame and the baseline frame. \(N\) is the total number of pixels, i.e. \(640 \times 480 = 307200\). Table 3.4 shows the quality of the color detection for each image in Fig. 5 compared to the GT image. Table 3.4 also shows that, after manual white balance tuning, all three channels had 12% better RMSE on average. To have a more realistic color detection, an online white balance calibration could be helpful in minimizing the average RMSE if needed. To check the quality of each image we compute the Peak Signal-to-Noise Ratio (PSNR), shown in Table 3.4.

Another way to do the color correction is to capture the image with a white channel by RGBW color spaces and perform the correction on four channels similar to RGBW CFA-equipped sensors [6]. This comes at the cost of adding a 4th color light to the SL, adding at least 33% to the length of the capture time.

4. ASL: Adaptive Structured Light

High-resolution ECs have a higher event rate and need more bandwidth compared to low-resolution ECs, but each

\(^1\)The OpenCV histogram comparison correlation method.

![Figure 5. Color detection of printed Macbeth color chart. Image captured by a frame-based high-resolution camera (left), the colorful image reconstructed by the proposed method captured by monochrome EC, without (middle) and with (right) white balance.](Image 319x491 to 390x600)

| Table 1. Color detection quality w.r.t. ground truth (GT) | \(RMSE_{red}\) | \(RMSE_{green}\) | \(RMSE_{blue}\) | \(RMSE\) | \(PSNR\) |
|---|---|---|---|---|---|
| GT | 0 | 83.89 | 83.65 | 87.79 | 71.16 | 9.88 dB | 8.82 dB | 8.82 dB |
| No WB | 0 | 87.79 | 71.16 | 83.65 | 76.97 | 9.88 dB | 8.82 dB | 8.82 dB |
| WB | 0 | 83.89 | 83.65 | 87.79 | 71.16 | 9.88 dB | 8.82 dB | 8.82 dB |

5462

EC has a limited data rate (finite bandwidth) on the output interface or bus. If the data rate or the number of events exceeds the limit, bus saturation could happen [10, 11]. Filtering [9] or online event-rate control [8] can mitigate this issue. When using an external event generator such as the DLP projector which emits SL on the scene, controlling the event rate is even more important. One method to control the event rate when using a projector is to define a region of interest (ROI) and project the pattern only where it is needed. Muglikar et al. [28] used one EC camera to detect the ROI (generally the area of the image frame that has more events due to the movement) and then projected the SL on that area followed by detecting the depth with a second EC. Instead of adding a second EC to the system, we introduced ASL to control the event-rate. Fig. 6 shows different patterns of the SL which change based on the number of received events. As expected, there is a trade-off between having high-resolution (dense) and high-speed (sparse) color detection. The generated SL patterns are, multiple dots or lines patterns and solid patterns. In static conditions, ASL could also be used with a color EC and white light. However, it should be noted that color ECs need more bandwidth compared to monochrome ECs with the same resolution.

**Bandwidth control:** By frequently projecting SL into the scene, we receive events caused by the SL alongside the events caused by the movement of objects or the camera.
We used a colorful board in our experiments, placed tor, the CP could be different on the camera frame plane. depending on the relative pose of the camera to the projec-
sider the CP on the DLP frame plane despite the fact that, 
assume that the DLP and the EC are close and we can con-
ing a different CP. To additionally simplify the problem, we 
as the coverage percentage (CP), with each pattern type hav-
ing a different CP. To additionally simplify the problem, we 
s the coverage percentage (CP), with each pattern type hav-
 we can control it by changing the pattern and the power 
of the LED projector. Modelling the DLP projector 
variation of power density changes with distance for each 
leds have the same power density. As a result, 
SL, we need to control $S_{\text{pow}}$ from (1).

Considering a one-bit pattern, we can control $S_{\text{pow}}$ by 
changing the pattern (changing the number of white pixels 
in a black and white frame), instead of changing the current 
of the LEDs. We call the number of white pixels per frame 
as the coverage percentage (CP), with each pattern type hav-
ing a different CP. To additionally simplify the problem, we 
assume that the DLP and the EC are close and we can con-
sider the CP on the DLP frame plane despite the fact that, 
depending on the relative pose of the camera to the projec-
tor, the CP could be different on the camera frame plane. We 
used a colorful board in our experiments, placed $160\text{cm}$

\[
\text{Max. Bandwidth} < \text{events}_{\text{SL}} + \text{events}_{M}, \quad (6)
\]

where $\text{events}_{M}$ is the number of events caused by the 
movement of the EC or any object in the scene (i.e., 
any other events that have not emerged due to the SL). 
$\text{events}_{\text{SL}}$ is the number of events caused by the SL, and 
we can control it by changing the pattern and the power 
of the LED projector. $\text{events}_{\text{SL}}$ is not only linked to the 
color of the object (and its reflectivity/fluorescence percent-
age, which we do not investigate in this paper), but also it is 
related to the distance of the camera-projector from the ob-
ject. Increasing the distance, the spectral power distribution 
decreases because of the reduction in power density. Unfor-
tunately, there is no information available concerning the 
variation of power density changes with distance for each 
LED of the DLP projector. Modelling the DLP projector 
power density could be useful, but it is out of the scope of 
this paper. In this work, we make the simplifying assump-
tion that all LEDs have the same power density. As a result, 
to control $\text{events}_{\text{SL}}$, we need to control $S_{\text{pow}}$ from (1).

Multiple-lines pattern Since dot-grid patterns are lead-
ning to a sparse image, to generate a dense image, line pat-
terns are preferred in low-speed 3D scanning and multi-
shot 3D measurement methods. Sequential projection tech-
niques mostly use strip lines [48]. Since the DLP projector 
can quickly switch (4225 Hz) between patterns, it is possi-
ble to generate a dense graph for some region of the object 
by projecting lines and measuring the depth with triangu-
lation. Although for the spaces between lines we do not 
have measurements, increasing the number of lines gener-
ates more features and it covers a larger area. Similarly to 
the dot-grid pattern, increasing the number of lines or dots 
increases the scanning processing time and event rate, so 
speed and detail must be traded off. The third and fourth 
rows from top in Fig. 6, show the proposed ASL with the 
line patterns where $M$ and $N$ ($M < N$) are the number of dots on each 
grid. Fig. 7 shows three different dot patterns with differ-
ent CPs. The top row is generated with a temporal window 
size of 2.5ms (equivalent to 400fps). Similarly, the second 
row has a window of 4.34ms or 230fps, and the bottom 
row for 7.14ms or 140fps. The leftmost column of Fig. 7 
is a ground truth (GT) frame generated with a one-second 
temporal window; the middle column is an example frame 
among the 430 frame samples. We compare each frame 
pixel by pixel with the GT frame to compute the $RMSE$ 
for each channel, shown in the rightmost column.

Moving-line pattern To have a full dense scanning in 3D, 
a line pattern is very common [5, 20, 27]. We propose to 
use a moving line pattern (horizontal or vertical depending 
on the offset between the camera and the projector), when 
the event rate is lower than the bandwidth limits, providing 
dense scanning.

Solid pattern Whenever the 3D scanning is performed, or 
when we need the color information only for a specific area
Figure 7. Colorful board scanned by dot patterns with varying CP. The temporal window sizes are 2.5, 4.3, and 7.14 ms from the top.

To compare different pattern and speed of scanning, we projected patterns with various CPs onto the colorful board. Fig. 9 shows the trade-off between details, speed, and the quality of the reconstructed colorful image. It shows that to have a more detailed image, we need to spend more time switching patterns to cover more area. Also, for high speed scanning, a sparse pattern (lower CP with fewer details) is needed. Note that a sparse pattern does not decrease the quality of the color detection even with high speed sampling. Fig. 9 has been generated by using \( \sim 24000 \) frames.

5. Conclusions

We present a method to add color and depth to a monocular, monochrome event-based camera while maintaining fast response time and resolution. Our method reconstructs colorful events and frames using a monochrome EC aided by adaptive structured light (ASL). By dynamically adjusting the projection, we have color data when needed, managing the overall bandwidth of the system.

We achieved a color detection speed equivalent of 1400 fps with a Texas Instrument’s DLP LightCrafter 4500 projector. Our method could be used in event-based depth measurement and perception projects. Advantages of ECs, makes the colorful depth detection much faster than RGBD cameras.

Although color detection is related to the lighting conditions and material properties at the intersection point (object surface), the scope of this work was the color detection on common materials that are generally matte and not too shiny (with high reflection) or fluorescence. Some materials can interact with light: they can be absorbing, scattering or emitting light [12]. In this work we focused on visual light wavelength (emitted by the LED projector) and materials that are not in the category of fluorescence and they
Figure 8. Colorful board scanned by line patterns with varying CP. The temporal window sizes are 6.67ms for the top and 7.14ms otherwise.

Figure 9. Comparing patterns with different CPs in speed and quality of color detection.

do not change the wavelength of the light. However, the use of the event-based camera with a different type of light source and materials could be investigated in future works. Also, without considering color detection, static reflective materials can be scanned more effectively with ECs when compared to the other depth measurement devices [20]. To detect the color of these kind of materials, a Blinn-Phong shading model [4] could be considered in future works.
References

[1] Ignacio Alzugaray and Margarita Chi. Asynchronous corner detection and tracking for event cameras in real time. *IEEE Robotics and Automation Letters*, 3(4):3177–3184, 2018.

[2] Juan Barrios-Avilés, Taras Iakymchuk, Jorge Samaniego, Leandro D Medus, and Alfredo Rosado-Muñoz. Movement detection with event-based cameras: Comparison with frame-based cameras in robot object tracking using power-link communication. *Electronics*, 7(11):304, 2018.

[3] Bryce E Bayer. Color imaging array. *United States Patent 3,971,065*, 1976.

[4] James F Blinn. Models of light reflection for computer synthesized pictures. In *Proceedings of the 4th annual conference on Computer graphics and interactive techniques*, pages 192–198, 1977.

[5] Christian Brandli, Thomas Mantel, Marco Hutter, Markus Höpflinger, Raphael Berner, Roland Siegwart, and Tobi Delbruck. Adaptive pulsed laser line extraction for terrain reconstruction using a dynamic vision sensor. *Frontiers in neuroscience*, 7:275, 2014.

[6] Wonseok Choi, Hyun Sang Park, and Chong-Min Kyung. Color reproduction pipeline for an rgbw color filter array sensor. *Optics Express*, 28(10):15678–15690, 2020.

[7] Aurélien David and Lorne A Whitehead. Led-based white light. *Comptes Rendus Physique*, 19(3):169–181, 2018.

[8] Tobi Delbruck, Rui Graca, and Marcin Paluch. Feedback control of event cameras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1324–1332, 2021.

[9] Thomas Finatou, Atsumi Niwa, Daniel Matolin, Koya Tsuchimoto, Andrea Mascheroni, Etienne Reynaud, Pooria Mostafalu, Frederick Brady, Ludovic Chotard, Florian LeGoff, et al. 5.10 a 1280×720 back-illuminated stacked temporal contrast event-based vision sensor with 4.86 μm pixels, 1.066 geps readout, programmable event-rate controller and compressive data-formating pipeline. In *2020 IEEE International Solid-State Circuits Conference (ISSCC)*, pages 112–114. IEEE, 2020.

[10] Guillermo Gallego, Tobi Delbruck, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew Davison, Jörg Conradt, Kostas Danilidis, et al. Event-based vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[11] Daniel Gehrig and Davide Scaramuzza. Are high-resolution event cameras really needed? *arXiv preprint arXiv:2203.14672*, 2022.

[12] George G Guibault. *Practical fluorescence*. CRC Press, 2020.

[13] Chen Haoyu, Teng Minggui, Shi Boxin, Wang Yizhou, and Huang Tiejun. Learning to deblur and generate high frame rate video with an event camera. *arXiv preprint arXiv:2003.00847*, 2020.

[14] Hae Woong Jun and Yong Ju Jung. Deep color transfer for color-plus-mono dual cameras. *Sensors*, 20(9):2743, 2020.

[15] Daniel Khoshabi, Sebastian Nowozin, Jeremy Jancsary, and Andrew W Fitzgibbon. Joint demosaicing and denoising via learned nonparametric random fields. *IEEE Transactions on Image Processing*, 23(12):4968–4981, 2014.

[16] Anat Levin, Dani Lischinski, and Yair Weiss. Colorization using optimization. In *ACM SIGGRAPH 2004 Papers*, pages 689–694, 2004.

[17] Hart Levy. Determining local depth from structured light using a regular dot grid. *Technical Disclosure Commons*, 2019.

[18] Chenghan Li, Christian Brandli, Raphael Berner, Hongjie Liu, Minhao Yang, Shih-Chii Liu, and Tobi Delbruck. Design of an rgbw color vga rolling and global shutter dynamic and active-pixel vision sensor. In *2015 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 718–721. IEEE, 2015.

[19] Alexandre Marcireau, Sio-Hoi Ieng, Camille Simon-Chane, and Rヤド B Benosman. Event-based color segmentation with a high dynamic range sensor. *Frontiers in neuroscience*, 12:135, 2018.

[20] Nathan Matsuda, Oliver Cossairt, and Mohit Gupta. MC3D: Motion Contrast 3D Scanning. In *2015 IEEE International Conference on Computational Photography (ICCP)*, pages 1–10. IEEE, 2015.

[21] Nico Messikomer, Stamatis Georgoulis, Daniel Gehrig, Stepan Tulyakov, Julius Erbach, Alfredo Bochicchio, Yuanyou Li, and Davide Scaramuzza. Multi-bracket high dynamic range imaging with event cameras. *arXiv preprint arXiv:2203.06622*, 2022.

[22] Anton Mitrokhin, Cornelia Fermüller, Chethan Parameshwara, and Yiannis Aloimonos. Event-based moving object detection and tracking. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–9. IEEE, 2018.

[23] Diederik Paul Moey, Federico Corradi, Chenghan Li, Simeon A Bamford, Luca Longinotti, Fabian F Voigt, Stewart Berry, Gemma Tavner, Fritjof Helmchen, and Tobi Delbruck. A sensitive dynamic and active pixel vision sensor for color or neural imaging applications. *IEEE transactions on biomedical circuits and systems*, 12(1):123–136, 2017.

[24] Diederik Paul Moey, Chenghan Li, Julien NP Martel, Simeon Bamford, Luca Longinotti, Vasyl Motysni, David San Segundo Bello, and Tobi Delbruck. Color temporal contrast sensitivity in dynamic vision sensors. In *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–4. IEEE, 2017.

[25] Mohammad Mostafavi, Lin Wang, and Kuk-Jin Yoon. Learning to reconstruct hdr images from events, with applications to depth and flow prediction. *International Journal of Computer Vision*, 129(4):900–920, 2021.

[26] Elias Mueggler, Guillermo Gallego, Henri Rebecq, and Davide Scaramuzza. Continuous-time visual-inertial odometry for event cameras. *IEEE Transactions on Robotics*, 34(6):1425–1440, 2018.

[27] Manasi Muglikar, Guillermo Gallego, and Davide Scaramuzza. Esl: Event-based structured light. In *2021 International Conference on 3D Vision (3DV)*, pages 1165–1174. IEEE, 2021.

[28] Manasi Muglikar, Diederik Paul Moey, and Davide Scaramuzza. Event guided depth sensing. In *2021 International Conference on 3D Vision (3DV)*, pages 1165–1174. IEEE, 2021.
Gottfried Munda, Christian Reinbacher, and Thomas Pock. Real-time intensity-image reconstruction for event cameras using manifold regularisation. International Journal of Computer Vision, 126(12):1381–1393, 2018.

Subramanian Muthu and James Gaines. Red, green and blue led-based white light source: implementation challenges and control design. In 38th IAS Annual Meeting on Conference Record of the Industry Applications Conference, 2003., volume 1, pages 515–522. IEEE, 2003.

Subramanian Muthu, Frank J Schuurmans, and Michael D. Pasley. Red, green, and blue led based white light generation: issues and control. In Conference Record of the 2002 IEEE Industry Applications Conference. 37th IAS Annual Meeting (Cat. No. 02CH37344), volume 1, pages 327–333. IEEE, 2002.

Liyuan Pan, Richard Hartley, Cedric Scheerlinck, Miaomiao Liu, Xin Yu, and Yuchao Dai. High frame rate video reconstruction based on an event camera. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020.

Liyuan Pan, Cedric Scheerlinck, Xin Yu, Richard Hartley, Miaomiao Liu, and Yuchao Dai. Bringing a blurry frame alive at high frame-rate with an event camera. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 6820–6829, 2019.

Matt Pharr, Wenzel Jakob, and Greg Humphreys. Physically based rendering: From theory to implementation. Morgan Kaufmann, 2016.

Rajeev Ramanath, Wesley E Snyder, Younghun Yoo, and Mark S Drew. Color image processing pipeline. IEEE Signal Processing Magazine, 22(1):34–43, 2005.

Henri Rebecq, Guillermo Gallego, Elias Mueggler, and Davide Scaramuzza. EMVS: Event-based multi-view stereo—3D reconstruction with an event camera. International Journal of Computer Vision, 126(12):1394–1414, 2018.

Henri Rebecq, Timo Horstschäfer, Guillermo Gallego, and Davide Scaramuzza. EVO: A geometric approach to event-based 6-dof parallel tracking and mapping in real time. IEEE Robotics and Automation Letters, 2(2):593–600, 2017.

Henri Rebecq, René Ranftl, Vladlen Koltun, and Davide Scaramuzza. Events-to-video: Bringing modern computer vision to event cameras. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 3857–3866, 2019.

Henri Rebecq, René Ranftl, Vladlen Koltun, and Davide Scaramuzza. High speed and high dynamic range video with an event camera. IEEE transactions on pattern analysis and machine intelligence, 43(6):1964–1980, 2019.

Christian Reinbacher, Gottfried Munda, and Thomas Pock. Real-time panoramic tracking for event cameras. In 2017 IEEE International Conference on Computational Photography (ICCP), pages 1–9. IEEE, 2017.

Cedric Scheerlinck, Nick Barnes, and Robert Mahony. Continuous-time intensity estimation using event cameras. In Asian Conference on Computer Vision, pages 308–324. Springer, 2018.

Cedric Scheerlinck, Nick Barnes, and Robert Mahony. Asynchronous spatial image convolutions for event cameras. IEEE Robotics and Automation Letters, 4(2):816–822, 2019.

Cedric Scheerlinck, Henri Rebecq, Timo Stoffregen, Nick Barnes, Robert Mahony, and Davide Scaramuzza. Ced: Color event camera dataset. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pages 0–0, 2019.

Amos Sironi, Manuele Brambilla, Nicolas Bourdais, Xavier Lagorce, and Ryad Benosman. Hats: Histograms of averaged time surfaces for robust event-based object classification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1731–1740, 2018.

Lea Steffen, Daniel Reichard, Jakob Weinland, Jacques Kaiser, Arne Roennau, and Rüdiger Dillmann. Neuromorphic stereo vision: A survey of bio-inspired sensors and algorithms. Frontiers in Neurorobotics, 13:28, 2019.

Gemma Taverni, Diederik Paul Moeyts, Chenghan Li, Celso Cavaco, Vasyl Motsnyi, David San Segundo Bello, and Tobi Delbruck. Front and back illuminated dynamic and active pixel vision sensors comparison. IEEE Transactions on Circuits and Systems II: Express Briefs, 65(5):677–681, 2018.

Alain Tremante, Shoji Tomina, and Konstantinos Plataniotis. Color in image and video processing: most recent trends and future research directions. EURASIP Journal on Image and Video Processing, 2008:1–26, 2008.

Sam Van der Jeught and Joris JJ Dirckx. Real-time structured light profilometry: a review. Optics and Lasers in Engineering, 87:18–31, 2016.

Yuwei Wang, Xiangcheng Chen, Jiayuan Tao, Keyi Wang, and Mengchao Ma. Accurate feature detection for out-of-focus camera calibration. Applied optics, 55(28):7964–7971, 2016.

Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In European conference on computer vision, pages 649–666. Springer, 2016.

Yi Zhou, Guillermo Gallego, Henri Rebecq, Laurent Kneip, Hongdong Li, and Davide Scaramuzza. Semi-dense 3d reconstruction with a stereo event camera. In Proceedings of the European Conference on Computer Vision (ECCV), pages 235–251. Springer International Publishing, 2018.

Alex Zihao Zhu, Nikolay Atanasov, and Kostas Daniilidis. Event-based visual inertial odometry. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 5391–5399, 2017.