EventCap

Amogh Tiwari

20.07.2021

01. About

Major Contributions of The Paper

★ To capture fast human motions (eg. Boxing, Dancing etc.), high frame rate is a critical requirement (as otherwise, there is a severe motion blur in the video)

★ Existing methods are constrained by:
  - Lighting requirements
  - High data bandwidth
  - High computational overhead (goes together with high data bandwidth)

★ Paper proposes the first approach for 3D capturing of high speed human motion using a single event camera

★ Paper promises tracking speeds of upto 1000 fps


Our results

Publication Details

- ★ Title: EventCap: Monocular 3D Capture of High-Speed Human Motions using an Event Camera **[1]**
- ★ Conference: CVPR 2020
- ★ Research Groups:
  - ○ Max Planck Institute for Informatics (MPII), Germany
  - ○ Tsinghua-Berkeley Shenzhen Institute, Tsinghua University China
  - ○ Robotics Institute, HKUST

02. Background: Event Cameras

What are Event Cameras?

★ Event Cameras are bio-inspired cameras which respond to local changes in brightness
★ They do not capture images using a shutter like a conventional camera, instead, each pixel operates independently in event cameras
★ They have gained a lot of popularity in the past few years. Specially, in the robotics domain

Suppose you have invited guests to your home

Image Credits: https://www.chicagotribune.com/lifestyles/sc-fam-social-graces-plus-one-0918-story.html

While waiting for them ...

Image Credits: https://www.ignant.com/2018/12/12/the-art-of-waiting/

You may do this

Image Credits: https://www.oxfordlearning.com/tips-for-studying-at-home/

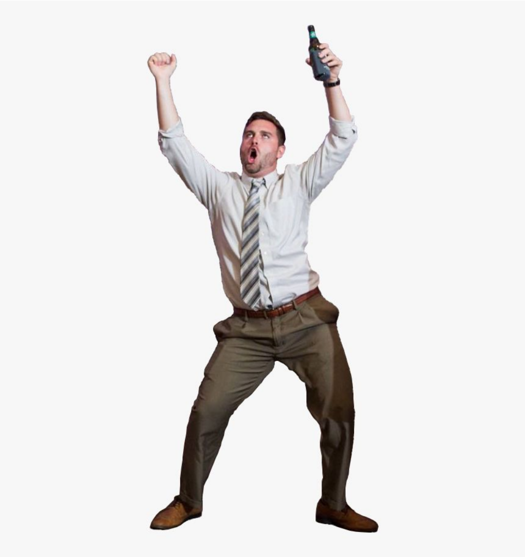


Image Credits: https://www.kindpng.com/imgv/hooRhb_person-dancing-png-transparent-png/

But **NOT** this

Image Credits: https://www.twenty20.com/photos/d500f551-40d0-48c2-86d1-b3c3fec7514a

Image Credits: https://techcrunch.com/2021/01/27/rings-new-doorbell-is-60/,
https://www.chicagotribune.com/lifestyles/sc-fam-social-graces-plus-one-0918-story.html

What does this mean?

- ★ As humans, if nothing is happening, we are “*disengaged*” or doing something else
- ★ Humans wait for things to happen **OR** We “react” to situations
- ★ Event Cameras are based on the same principal. Wait for an “event” to happen, and then, record it

How do we define an “event”?

- ★ Whenever, the brightness change is more than a predefined threshold ‘c’. An event is reported
- ★ +1 if positive change; -1 if negative change
- ★ Output of an event camera looks like: (x_i, y_i, p_i, t_i)
  - x_i, y_i: x, y coordinates of the pixel which was “activated”
  - p_i: Polarity (+1 or -1)
  - t_i: timestamp
- ★ Also called as “neuromorphic cameras” or “Dynamic Vision Sensors (DVS)”
- ★ **Note:** Event Cameras record data in logarithmic scale

Event Cameras: Key Principles

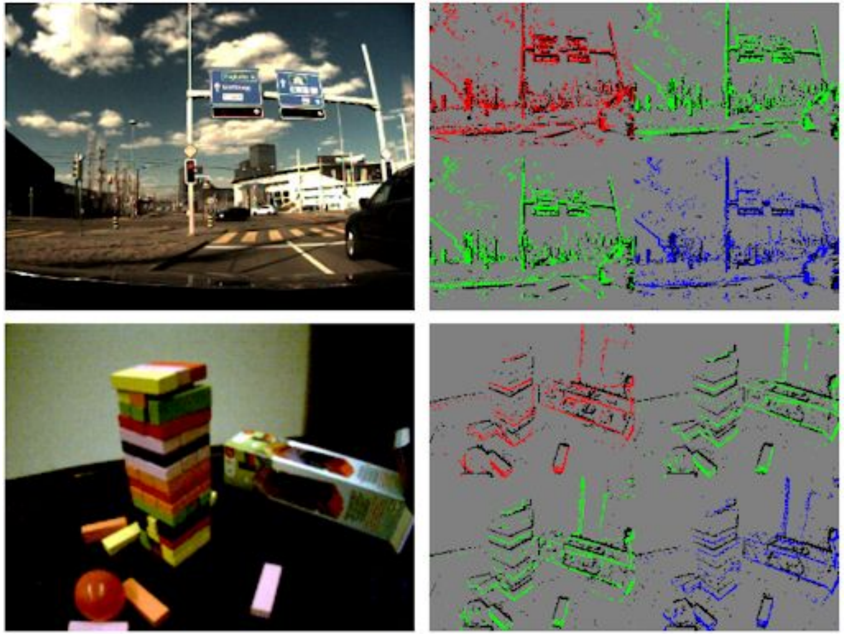
“An event camera, also known as a neuromorphic camera … is an imaging sensor that responds to local changes in brightness” **[1]**

- ★ Each pixel in an event camera operates - (1) Asynchronously; (2) Independently; (There is no “central clock”)
- ★ Each pixel reports changes in brightness as they occur
- ★ If nothing happens, it reports nothing (“stays silent”)

Benefits offered by Event Cameras?

- ★ Event Cameras offer a large number of benefits
- ★ Benefits offered:
  - ○ **Microsecond** temporal resolution
  - ○ High dynamic range (Event cameras operate in logarithmic domain)
  - ○ Less underexposure/overexposure
  - ○ Less motion blur
  - ○ Low power consumption
  - ○ Concise Information
  - ○ Work well in low lighting conditions
- ★ Some papers have also shown that using event camera data leads to better generalizability in a lot of tasks

Event Camera Outputs (Visualized)



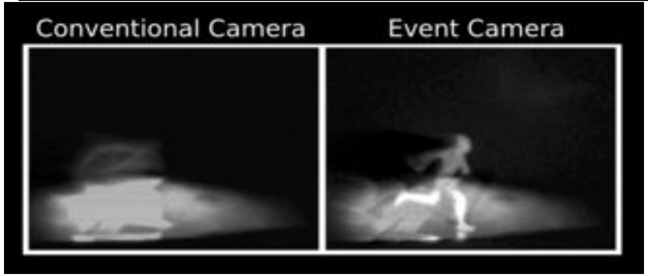


Image Credits: http://rpg.ifi.uzh.ch/research_dvs.html, https://www.youtube.com/watch?v=bVVBTQ7I36I, https://en.wikipedia.org/wiki/Event_camera

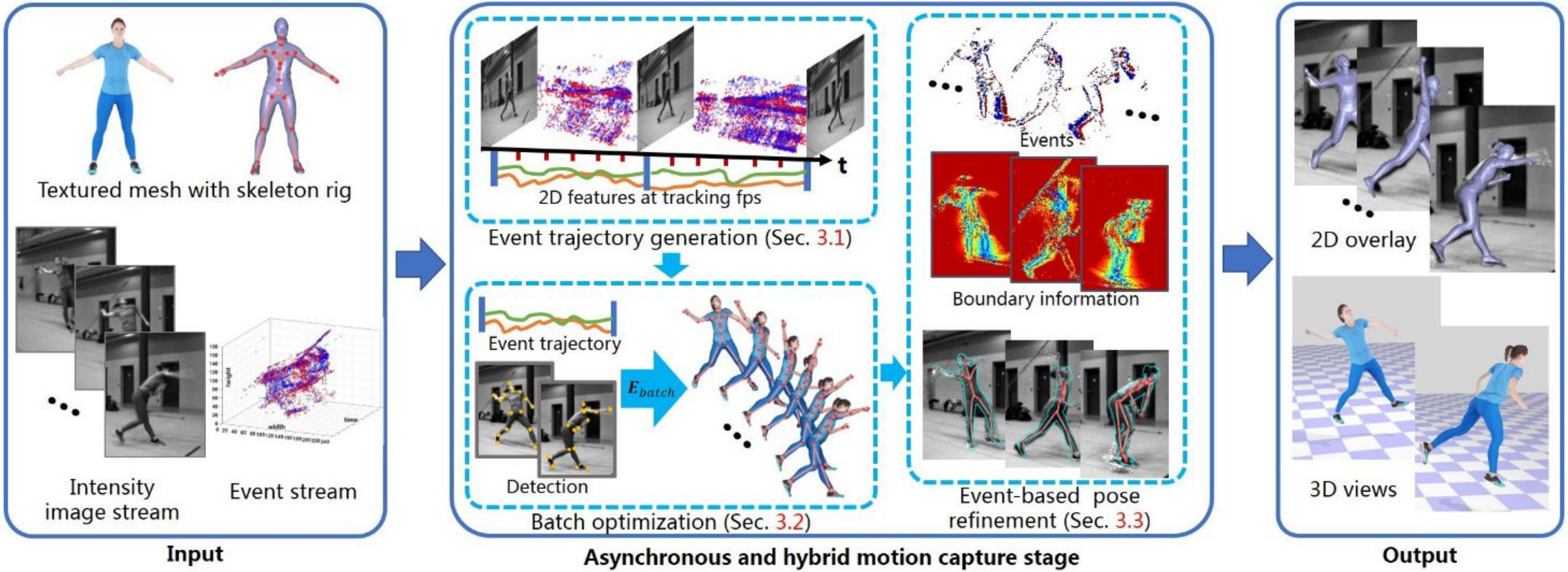
Limitations

- ★ Event cameras offer lots of benefits. However, some information is lost:
  - Spatio temporal information
    - Suppose pixel (2,4) was activated at t=5; and pixel (4,2) at t=10;
  - Exact brightness information is lost (Only polarity is reported)
- ★ How can these issues be solved?
  - DAVIS event cameras
  - DAVIS: **Dynamic and Active VIsion pixel Sensor** (Intensity Frames + Event data)
- ★ Active pixel sensor gives information at a slower frame rate (30-50 fps). Dynamic pixel sensor gives information at a faster frame rate (microsecond resolution)
  - Using both, we can get videos of upto 1000 fps

03. EventCap Method

EventCap

- ★ Key Steps:
  - 0. Template mesh (SMPL)
  - Get feature trajectories
  - Get pose using the trajectories and template mesh
  - Further refine pose using event data


Textured mesh with skeleton rig
Intensity image stream
Event stream
Input
2D features at tracking fps
t
Event trajectory generation (Sec. 3.1)
Event trajectory
Detection
E_{batch}
Batch optimization (Sec. 3.2)
Events
Boundary information
Event-based pose refinement (Sec. 3.3)
Asynchronous and hybrid motion capture stage
2D overlay
3D views
Output

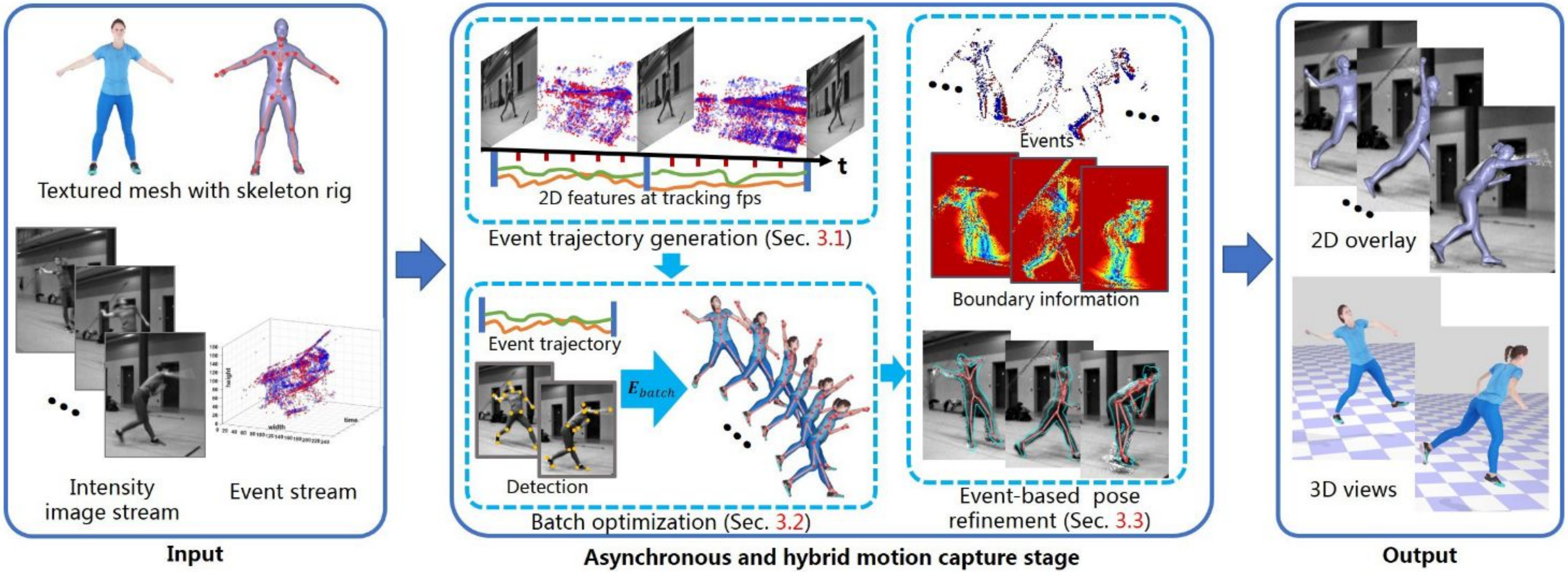
Step-0: Template Mesh Acquisition

★ A 3D body scanner is used to generate the template mesh of the actor

★ To rig the template mesh with a parametric skeleton, the Skinned Multi-Person Linear Model (SMPL) [**3**] is fitted to the template mesh by optimizing the body shape and pose parameters. After that, SMPL skinning weights are transferred to the scanned mesh

★ In case a 3D body scanner is unavailable, image based human shape algorithms can also be used [**4**] to obtain a SMPL mesh as a template mesh

Step-0 (Visually)



Step-1: Asynchronous Event Trajectory Generation

★ A single event does not carry any structural information and therefore tracking based on isolated events is not robust. Therefore, the authors use [**5**] to track 2D features in an asynchronous manner

★ The method in [**5**] requires sharp images but the intensity images captured by the event camera are blurry. So, the authors use [**6**] to sharpen the intensity images.

Step-1: Asynchronous Event Trajectory Generation

★ Feature tracking can drift over time

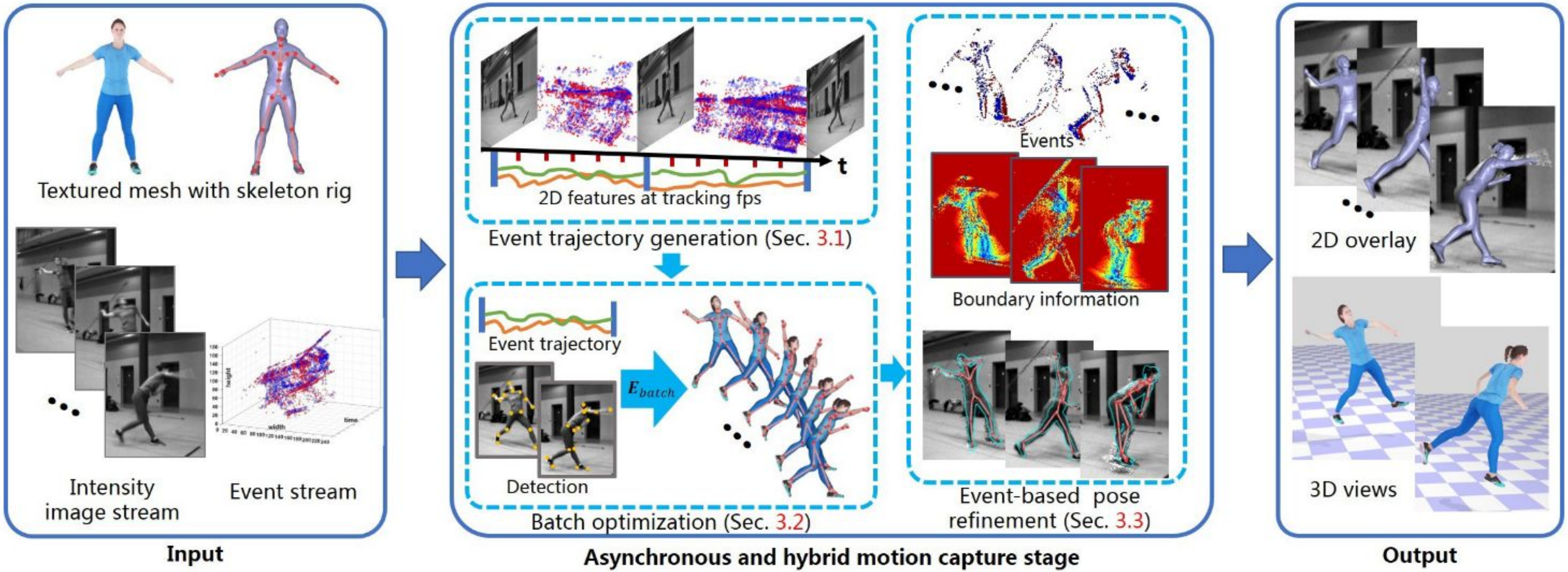
★ To prevent drifting,, once the trajectories of the features have been obtained, they are stitched in both "forward" and "backward" manner

★ If the 2D distance between two pixels is more than a threshold, the "bidirectional" stitching is **not** applied. This is done to ensure that feature tracking does not drift over time

★ For each stitched trajectory, a B-Spline curve is fitted to its discretely tracked 2D pixel locations in a batch to get a continuous event trajectory (**IMPORTANT**)

★ **Result:** At the end of this step, we get the trajectories followed by a set of 2D features across the frames

Step-1 (Visually)



Step-2: Hybrid Pose Batch Optimization

★ In order to tackle the drifting due to accumulation of tracking errors and the inherent depth ambiguities associated with a monocular setting, the problem of pose estimation is phrased as a constrained optimization problem

Note: This is the key step

(This space is intentionally left blank)

Step-2: Hybrid Pose Batch Optimization (contd.)

- All the skeleton poses in a batch are jointly optimized
  - A “batch” refers to the set of tracking frames present between two subsequent intensity images
  - “Tracking frames” refer to the tracking details of the features which we were tracking using [**3**]
  - In the image below, L_0 and L_N represent 2 subsequent intensite image frames (captured at time t_k and t_{k+1}), and L_1, L_2, …., L_{N-1} represent the tracking frames. (The “batch” is given by $\{L_0, L_1, \ldots L_N\}$)

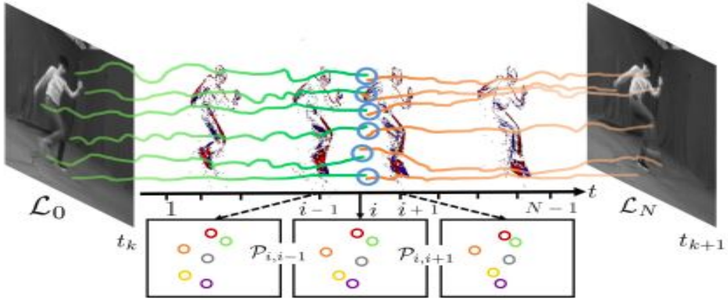


Figure 3: Illustration of asynchronous event trajectories between two adjacent intensity images. The green and orange curves represent the forward and backward event trajectories of exemplary photometric features. The blue circles denote alignment operation. The color-coded circles below indicate the 2D feature pairs between adjacent tracking frames.

Step-2: Hybrid Pose Batch Optimization (contd.)

- It is worth noting that:
  - All the poses in a batch are jointly optimized
  - Different batches are optimized independently of each other
- The paper leverages:
  - Event feature correspondences obtained in step-1
  - CNN based 2D and 3D pose estimates (Obtained using openpose & VNect respectively)
- The problem is framed as follows:

$$\textbf{min.}\ E_{batch}(S) = \lambda_{cor}E_{cor} + \lambda_{2D}E_{2D} + \lambda_{3D}E_{3D} + \lambda_{temp}E_{temp}, \quad \textbf{where:}$$

  - $\mathbf{E_{cor}}$ is the event correspondence term
  - $\mathbf{E_{2D}}$, $\mathbf{E_{3D}}$ are 2D and 3D detection terms
  - $\mathbf{E_{temp}}$ is the temporal stabilization term

Step-2: Hybrid Pose Batch Optimization (contd.)

- **Equation:** min. $E_{batch}(S) = \lambda_{cor}E_{cor} + \lambda_{2D}E_{2D} + \lambda_{3D}E_{3D} + \lambda_{temp}E_{temp}$

- $\mathbf{E_{cor}}$: Event Correspondence Term
  - This term encourages a vertex corresponding to a feature on the ith frame lands on it's trajectory in i+1th and i-1th frame [Event Tracking Information from step-1 is used here to for the Feature Trajectories]

- $\mathbf{E_{2D}}$, $\mathbf{E_{3D}}$: 2D and 3D detection terms
  - These terms encourage the posed skeleton to match the 2D and 3D body joint detection obtained by CNN from the intensity images. In order to get the 2D and 3D joint positions, OpenPose [**8**] and VNect [**9**] are applied respectively on the intensity images

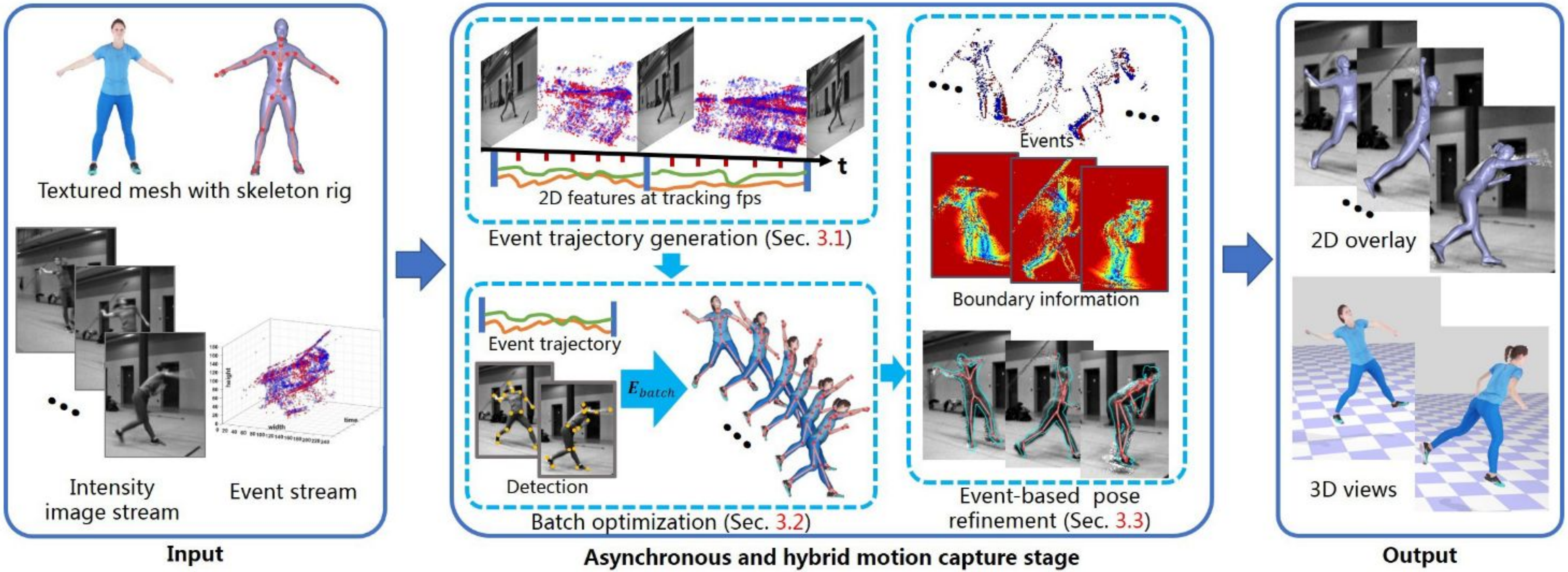
Step-2: Hybrid Pose Batch Optimization (contd.)

★ **Equation:** min. $E_{batch}(S) = \lambda_{cor}E_{cor} + \lambda_{2D}E_{2D} + \lambda_{3D}E_{3D} + \lambda_{temp}E_{temp}$

★ $\mathbf{E_{temp}}$**:** Temporal stabilization term
   - Since only moving body parts can trigger motion, this term penalizes changes in joint positions for non-moving body parts

★ The optimization problem is solved by using the Levenberg-Marquardt (LM) algorithm of ceres solver

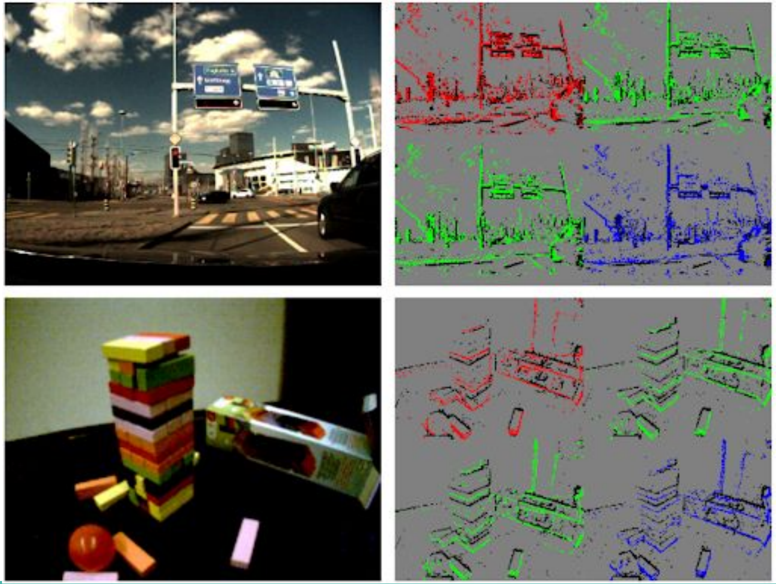
Step-2 (Visually)



Step-3: Event based pose refinement

★ A good estimate of the human motion is already available at the end of step-2. However, to further refine the pose, event based pose refinement is carried out

★ Most of the events are triggered by the moving edges in the image plane (see next slide), which have a strong correlation with the actor's silhouette. Based on this finding, the skeleton pose estimation is refined in an Iterative Closest Point (ICP) manner



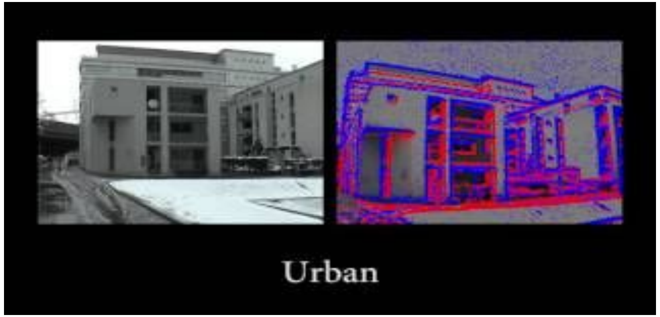
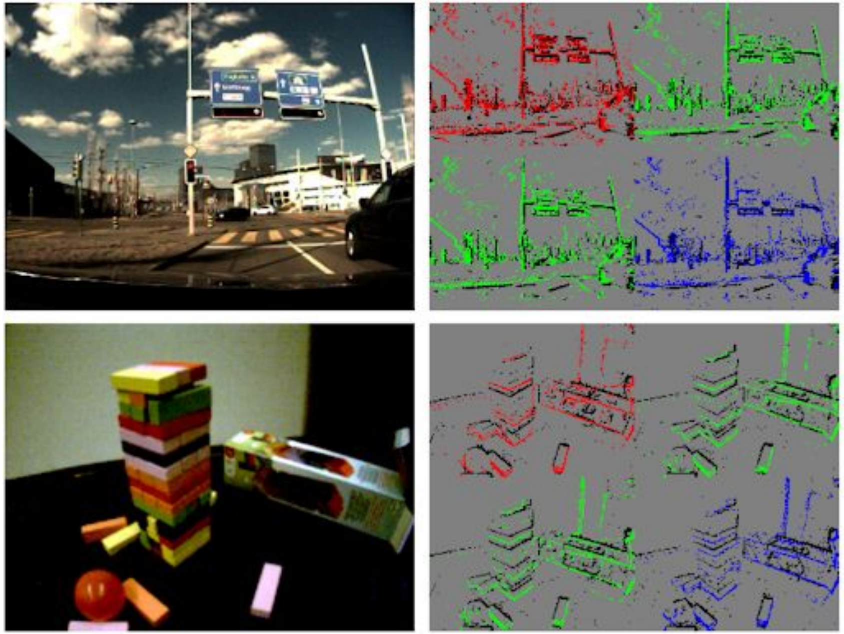
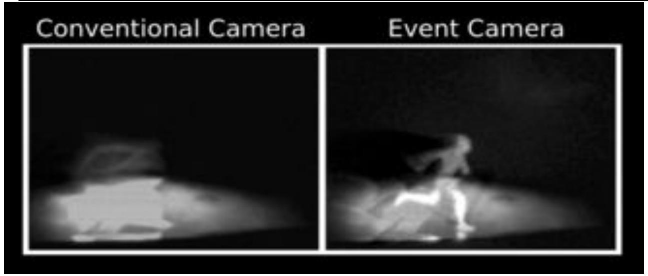


Image Credits:
http://rpg.ifi.uzh.ch/research_dvs.html,
https://www.youtube.com/watch?v=bVVBTQ7l36I

Note that Edges are enhanced in Event Camera Captures





Step-3: Event based pose refinement

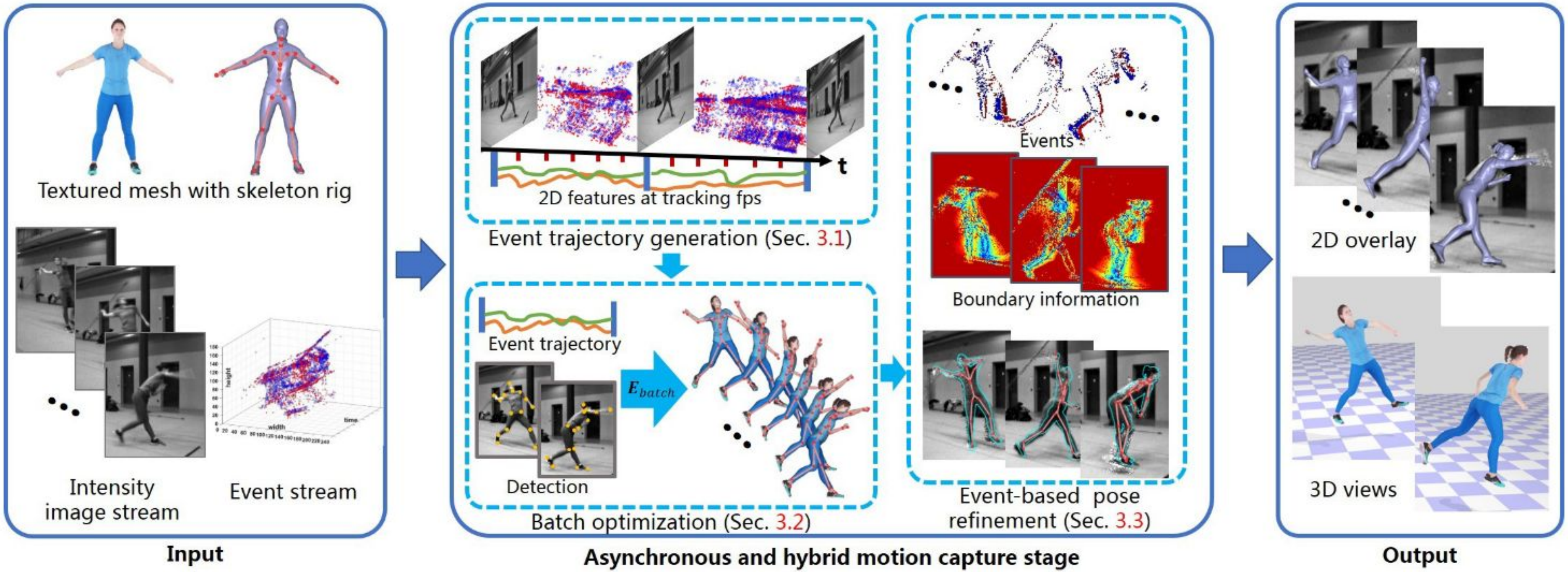
★ In each ICP iteration, we first search for the closest event for each boundary pixel of the projected mesh. Then, the pose is refined by solving the nonlinear least squares optimization problem:

$$E_{refine}(S_f) = \lambda_{sil} E_{sil}(S_f) + \lambda_{stab} E_{stab}(S_f),$$ **where:**

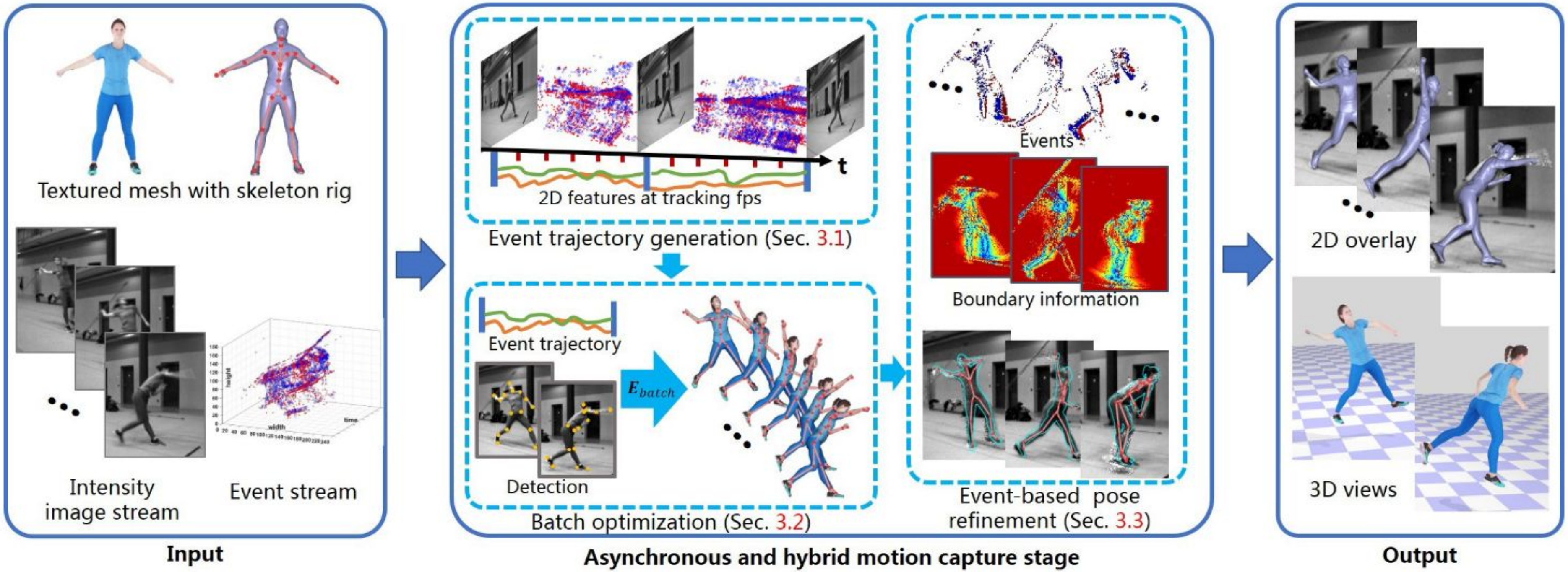
- $\mathbf{E_{refine}}$ is the refined pose
- $\mathbf{E_{stab}}$ enforces the refined pose to stay close to its initial estimate
- $\mathbf{E_{sil}}$**:** relies on closest event search and measures the 2D point-to-plane misalignment of the correspondences

Note: There is a lot more detail to the way E_{refine} and E_{stab} are defined. But this is the main idea

Step-3 (Visually)



Recap



Textured mesh with skeleton rig
Intensity image stream
Event stream
Input
2D features at tracking fps
Event trajectory generation (Sec. 3.1)
Event trajectory
Detection
E_{batch}
Batch optimization (Sec. 3.2)
Events
Boundary information
Event-based pose refinement (Sec. 3.3)
Asynchronous and hybrid motion capture stage
2D overlay
3D views
Output

04. General Notes

Other work using event cameras

- ★ High speed motion capture (Specially in driving scenes)
- ★ Motion deblurring
- ★ Image sharpening
- ★ Using existing algorithms on event data
  - Event based representations
    - Eg. Channel-1 = All positive events; Channel-2 = All negative events; Channel-3 = Time stamp of all positive ents; Channel-4 = Time stamp of all negative events; Channel-5,6 = Mean, Std. Dev of time stamps
  - EV-FlowNet; EV-SegNet
- ★ Refer https://github.com/uzh-rpg/event-based_vision_resources for more

Event Camera Simulators

- ★ Event cameras are costly. Not all groups can afford it
- ★ There have been attempts to make simulators which can help simulate event data
- ★ E-Sim by RPG (ETH Zurich) has been used by a lot of papers. Recently, another simulator named V2E has been proposed lately
  - E-SIM Paper: http://proceedings.mlr.press/v87/rebecq18a/rebecq18a.pdf
  - E-SIM Github: https://github.com/uzh-rpg/rpg_esim
  - V2E Paper: https://arxiv.org/abs/2006.07722
  - V2E Github: https://github.com/SensorsINI/v2e
- ★ Note: E-Sim is ROS based; V2E is python based

Further Readings

1. Wikipedia Page: https://en.wikipedia.org/wiki/Event_camera
2. Survey Paper: https://arxiv.org/pdf/1904.08405.pdf
3. Event Based vision repository: https://github.com/uzh-rpg/event-based_vision_resources
4. Event based vision workshop (CVPR 2021): https://tub-rip.github.io/eventvision2021/
5. RPG, ETH-Zurich website: http://rpg.ifi.uzh.ch/
6. Articles on event cameras:
    a. https://medium.com/@nabil.madali/introduction-to-event-based-vision-d9cfa1d98264
    b. https://medium.com/tangram-visions/event-cameras-where-are-they-now-293343754bfd
7. EventCap: https://gvv.mpi-inf.mpg.de/projects/2020-cvpr-eventcap/
8. Image Sharpening:
    a. Paper: https://openaccess.thecvf.com/content_CVPR_2019/papers/Pan_Bringing_a_Blurry_Frame_Alive_at_High_Frame-Rate_With_an_CVPR_2019_paper.pdf
    b. Github: https://github.com/panpanfei/Bringing-a-Blurry-Frame-Alive-at-High-Frame-Rate-with-an-Event-Camera;
9. EV-FlowNet: https://arxiv.org/abs/1802.06898
10. EV-SegNet: https://arxiv.org/abs/1811.12039
11. SMPL: https://khanhha.github.io/posts/SMPL-model-introduction/

References

References

1. EventCap: Monocular 3D Capture of High-Speed Human Motions using an Event Camera (https://gvv.mpi-inf.mpg.de/projects/2020-cvpr-eventcap/)
2. Event Camera, Wikipedia (https://en.wikipedia.org/wiki/Event_camera)
3. SMPL: A skinned multiperson linear model (https://smpl.is.tue.mpg.de/)
4. End-to-end recovery of human shape and pose (https://arxiv.org/abs/1712.06584)
5. Asynchronous, Photometric Feature Tracking using Events and Frames (https://arxiv.org/abs/1807.09713)
6. Bringing a Blurry Frame Alive at High Frame-Rate with an Event Camera (https://arxiv.org/abs/1811.10180)
7. OpenPose: Realtime multi-person 2d pose estimation using part affinity fields (https://arxiv.org/abs/1812.08008)
8. Vnect: Real-time 3d human pose estimation with a single rgb camera (http://gvv.mpi-inf.mpg.de/projects/VNect/)

Thank You!