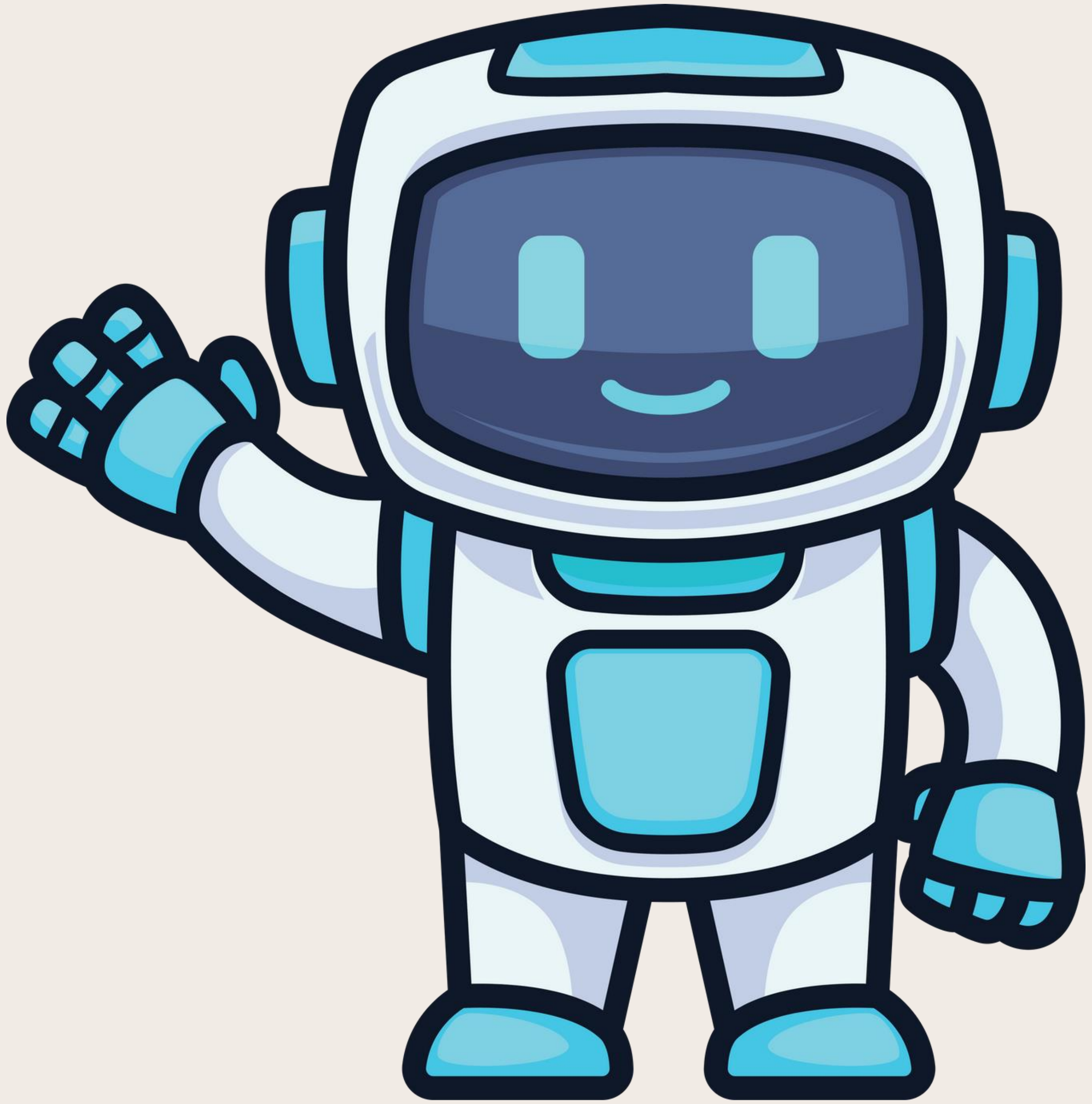
HAND GESTURE 3D POSE ESTIMATION AND RECOGNITION BASED ON EVENT CAMERA RECORDS

FINAL PRESENTATION ROBOTICS PERCEPTION



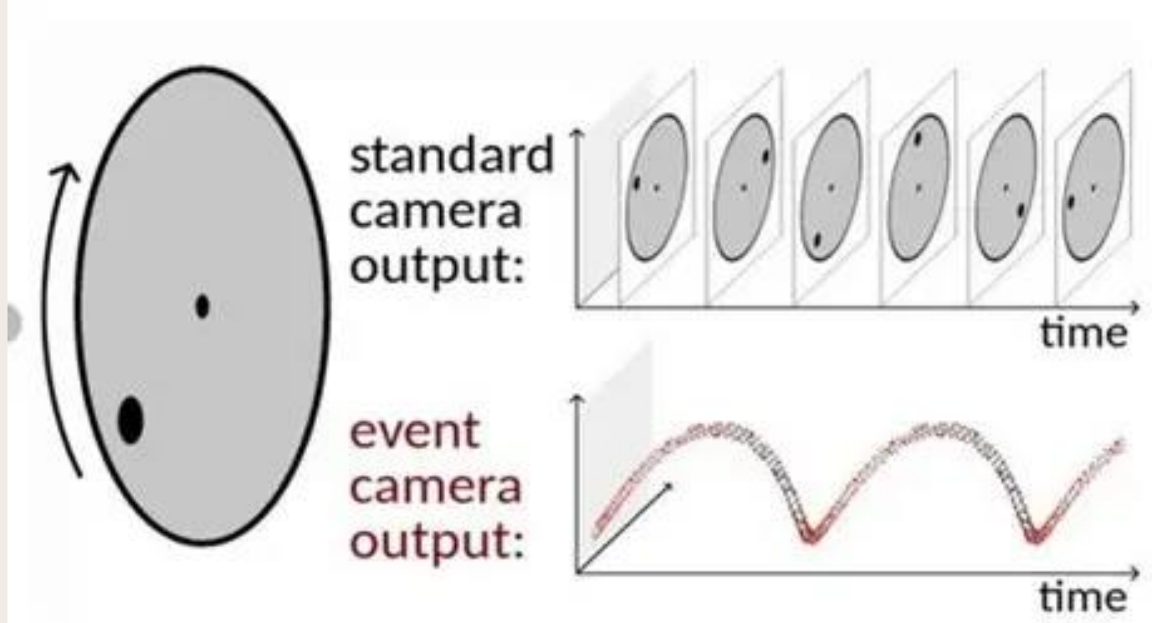
Presented by: **Shamma Alblooshi 100045387**

PROBLEM STATEMENT AND OBJECTIVES

INTRODUCTION

- Event-based vision mimics the human eye by capturing changes in the scene, offering advantages such as:
  - High temporal resolution: Records data at microsecond intervals.
  - Low latency: Enables real-time processing.
  - Power efficiency: Only processes changes, reducing computational overhead.

- Applications of event-based vision include robotics, augmented reality (AR), virtual reality (VR), and human-computer interaction.



3D Hand Gesture Posture and Recognition

- 3D hand gesture posture estimation involves determining the precise 3D positions of hand joints, enabling applications like:
  - Gesture-based interaction in virtual reality (VR) and augmented reality (AR).
  - Real-time control in robotics and gaming.
  - Sign language interpretation for improved accessibility.

Challenges with existing systems:

Motion Blur and Latency:

- Traditional vision systems struggle with motion blur during fast hand movements.
- Latency in processing full video frames impacts real-time applications.

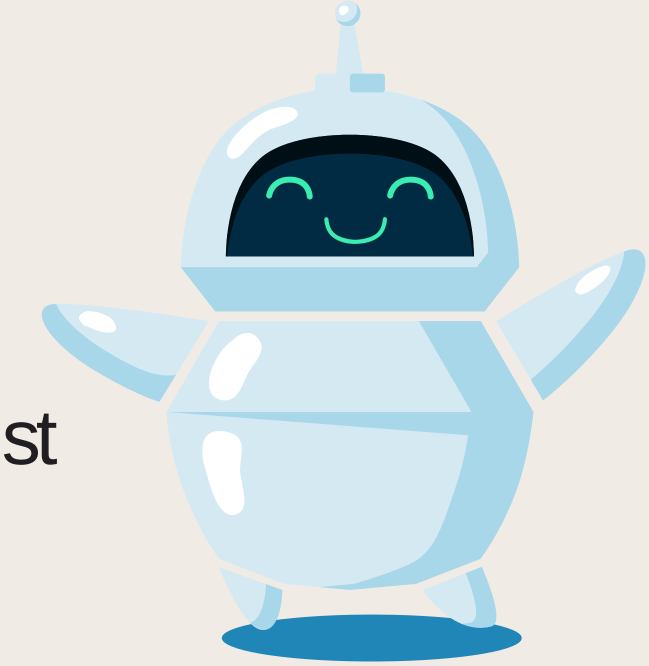
Computational Cost:

- Dense image data requires significant computational resources, which is unsuitable for low-power devices.

Event cameras overcome these issues, providing high-resolution temporal data, which is crucial for accurate and real-time 3D hand posture recognition.

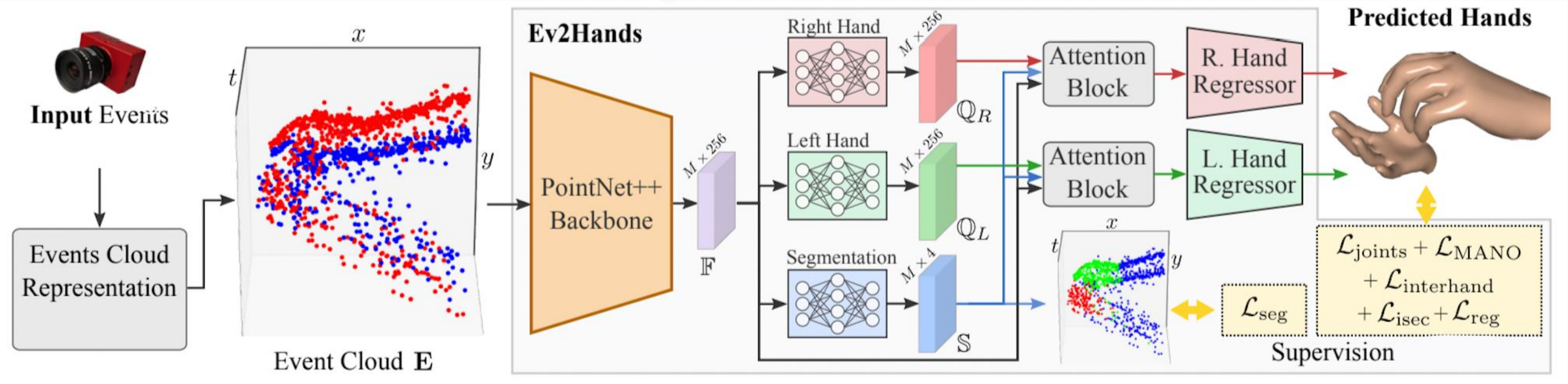
Proposed Solution:

- Utilize the Ev2Hands framework, which combines:
  - Event-based data streams: Captures only changes in the scene.
  - Pre-trained 3D hand pose models: Efficiently estimate joint positions.
  - Hierarchical feature extraction: Learns local and global features for robust predictions.



METHODOLOGY

FULL FRAMEWORK



CONVERTING EVENT DATA TO EVENT CLOUD REPRESENTATION

- **Input**:
  - Raw event stream data from the event camera (contains spatiotemporal information: [x, y, t, p], where p is polarity).

$$\mathbf{e}_i = (x_i, y_i, t_i, p_i)$$

  - High temporal resolution data for capturing dynamic hand movements.

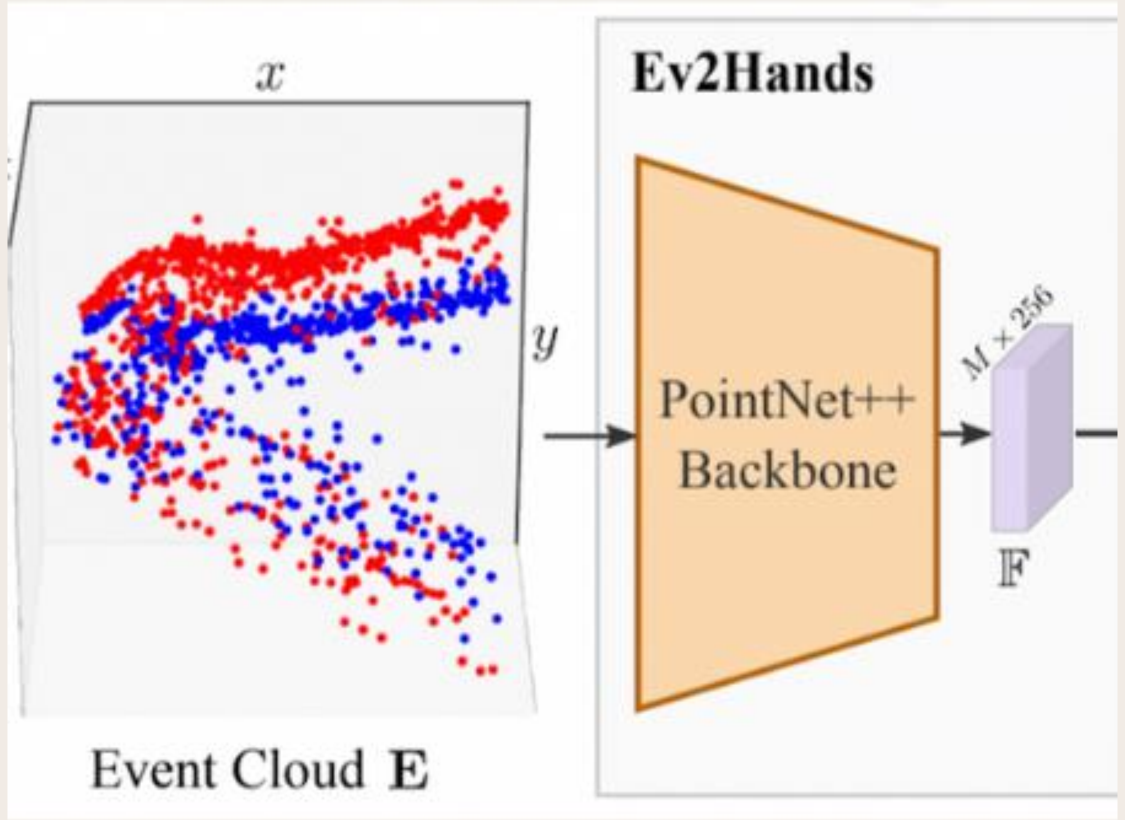
- **Process**:
  - Convert the event data into a structured **event cloud representation**, which is a set of points in a 3D space:

$$\mathbf{E}_k = (x_k, y_k, t_k, P_k, N_k)$$

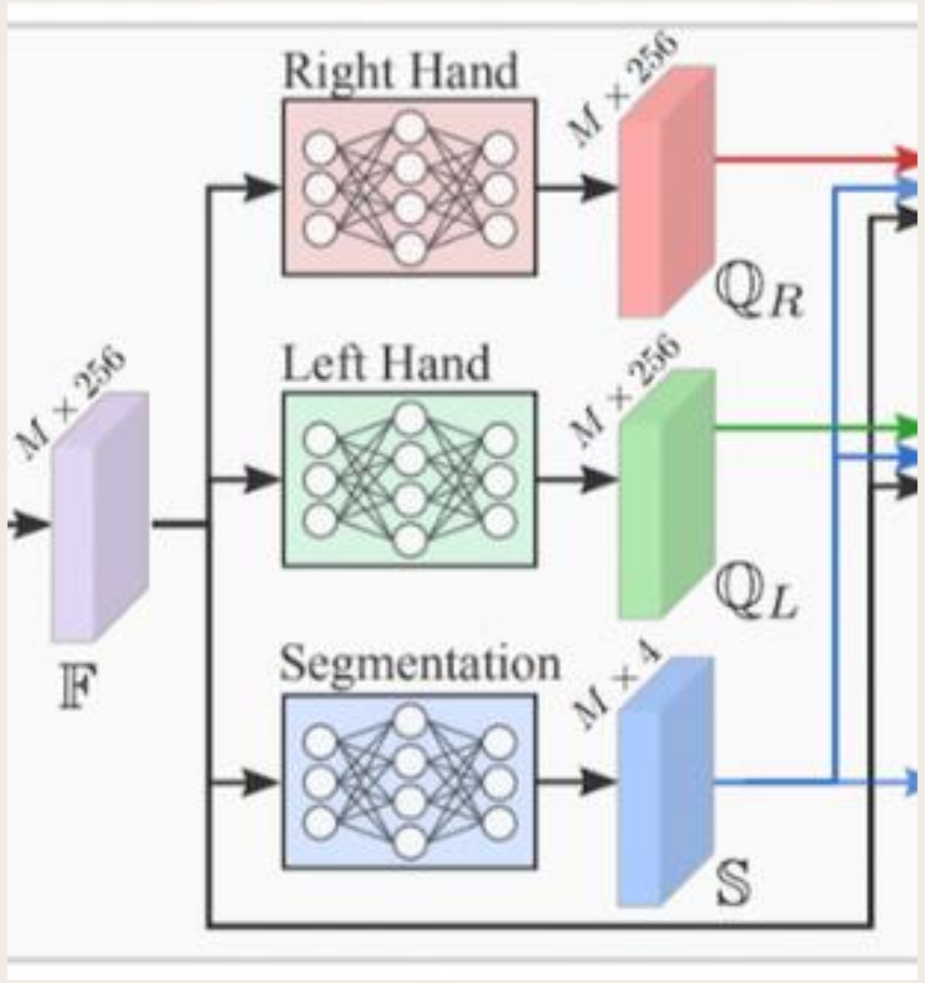
    - x, y: pixel coordinates
    - t: average time of the combined events
    - P, N: number of positive and negative events in the time interval considered
- **Output**:
  - A normalized **event cloud** that represents the spatiotemporal and polarity information of the input events.
  - Enables efficient processing using deep learning frameworks like PointNet++.

PROCESSING EVENT CLOUD WITH POINTNET++

- **Input**:
  - The normalized event cloud from the previous step.
- **Process**:
  - **PointNet++** architecture processes the event cloud hierarchically:
    - **Sampling**: Uses Farthest Point Sampling (FPS) to reduce the number of points while maintaining a uniform distribution.
    - **Grouping**: Applies ball query to group nearby points into local regions.
    - **Feature Learning**: Uses MLPs and max pooling to extract local and global features from the grouped points.
- **Output**:
  - **Event Feature Representation**:
    - A feature vector capturing the spatial and temporal relationships in the event cloud.
    - Encodes hand gesture dynamics for further processing.

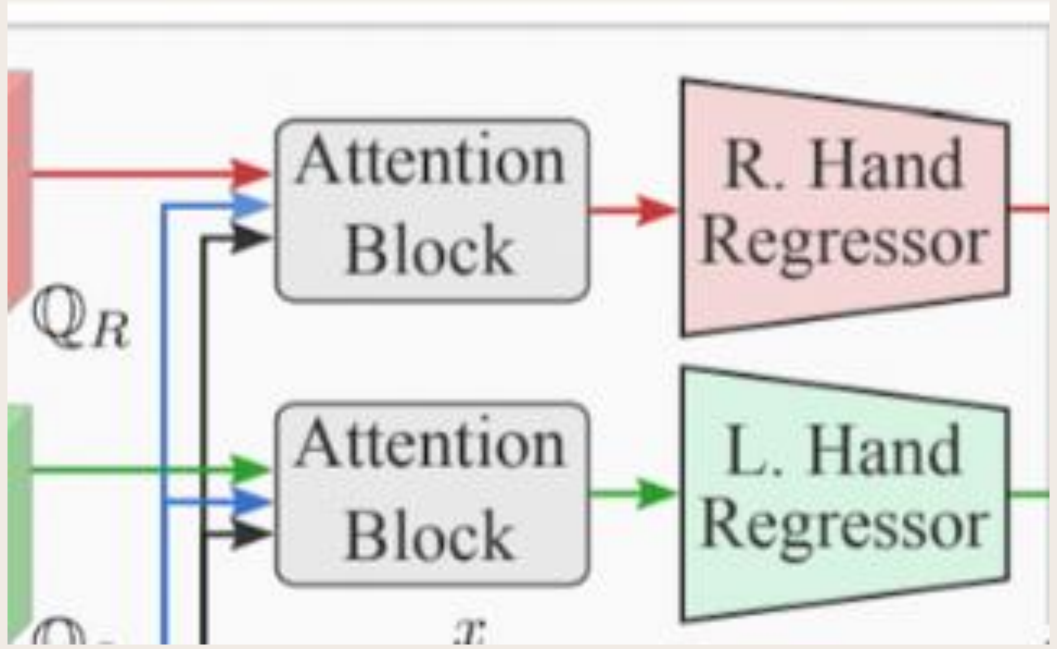


BRANCHING THE EVENT FEATURE REPRESENTATION



- **Input**:
  - Event features extracted from PointNet++.
- **Process**:
  - The event feature vector is passed through three parallel branches:
    1. **Left Hand Branch**:
       - Predicts features specific to the left hand.
       - Outputs: QL (feature vector for the left hand).
    2. **Right Hand Branch**:
       - Predicts features specific to the right hand.
       - Outputs: QR (feature vector for the right hand).
    3. **Segmentation Branch**:
       - Processes the event features to predict class logits for each point in the event cloud.
       - Outputs: S , which is a segmentation map
- **Outputs**:
  - Three feature vectors: QL, QR, and S, representing left hand, right hand, and segmenattion, respectively.

ENHANCING FEATURES WITH ATTENTION



- **Input**:
  - Feature vectors: QL (left hand), QR (right hand), S
- **Process**:
  - **Attention Block**:
    - Enhances the left (QL) and right (QR) hand features using attention mechanisms.
    - Computes the relationship between QL, QR, and S to focus on important spatial-temporal patterns.
    - The attention block adjusts the feature vectors to highlight critical regions and suppress irrelevant information.

$$\text{Attention}(\mathbb{Q}_{(\cdot)}, \mathbb{S}, \mathbb{F}) = \mathbb{F}\left(\text{Softmax}\left(\frac{\mathbb{Q}_{(\cdot)}^T \mathbb{S}}{\sqrt{d_s}}\right)\right)$$

- **Output**:
  - Enhanced feature vectors for the left and right hands: QL' and QR'.

REGRESSING HAND POSES

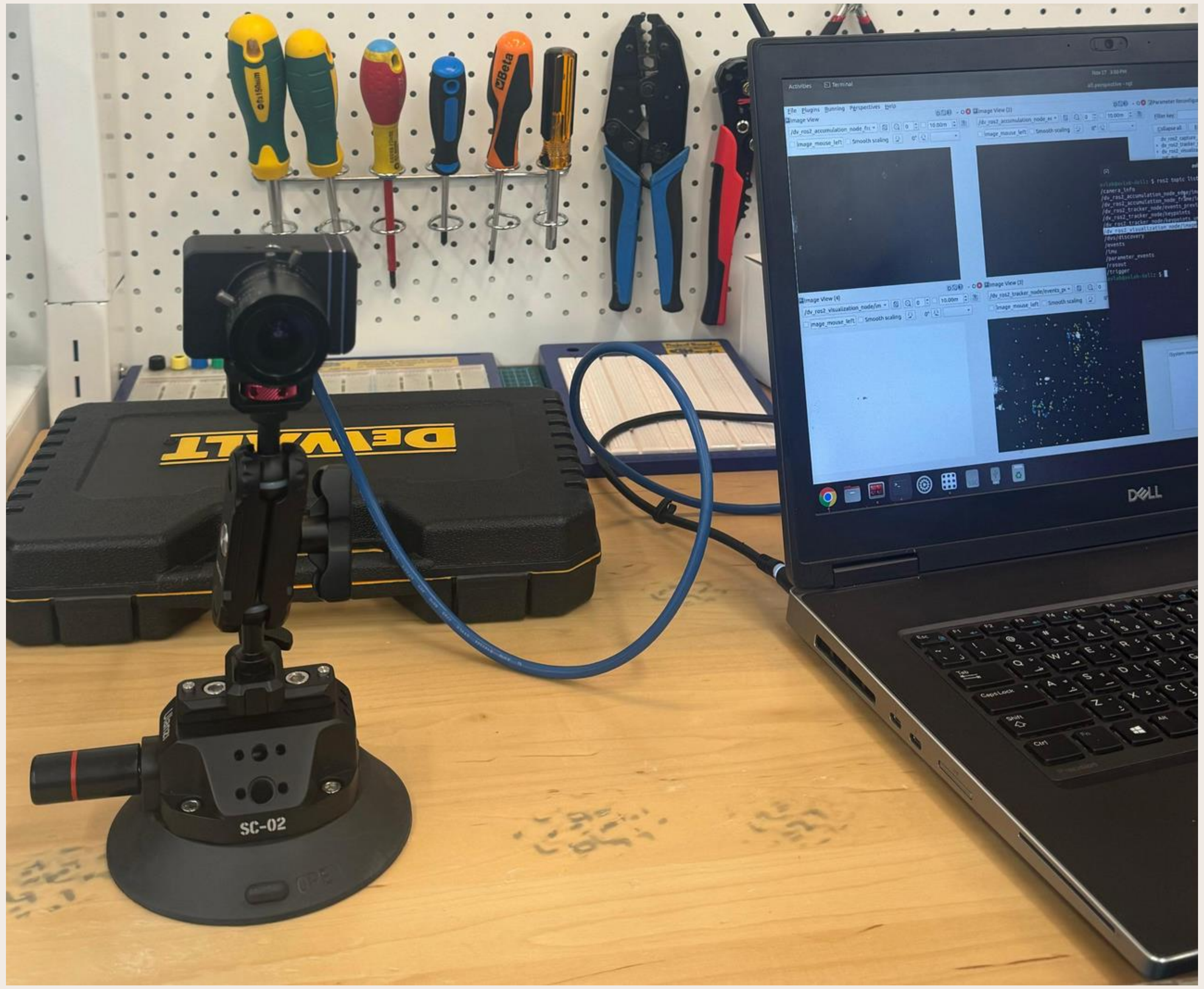
- **Input**:
  - Enhanced feature vectors QL' and QR'.
- **Process**:
  - **Regressor**:
    - A fully connected neural network that maps the enhanced features to 3D hand poses (joint positions).
  - **Loss Function**:
    - Combines multiple loss components to train the model effectively:
      - **Regression Loss**: Minimizes the error between predicted and ground truth joint positions.
      - **Temporal Consistency Loss**: Ensures smooth transitions between consecutive frames.
- **Output**:
  - Predicted 3D joint positions for the left and right hands.

GENERATING 3D HAND POSE PREDICTIONS

- **Input**:
  - Output of the regressor: Predicted 3D joint positions.
- **Process**:
  - The predicted joint positions are scaled and transformed to match the physical hand pose.
- **Output**:
  - Final 3D hand pose predictions:
    - Visualized as a set of 3D joint locations.
    - Enables applications like gesture recognition and virtual interaction.

EXPERIMENTAL SETUP & DATASET COLLECTION

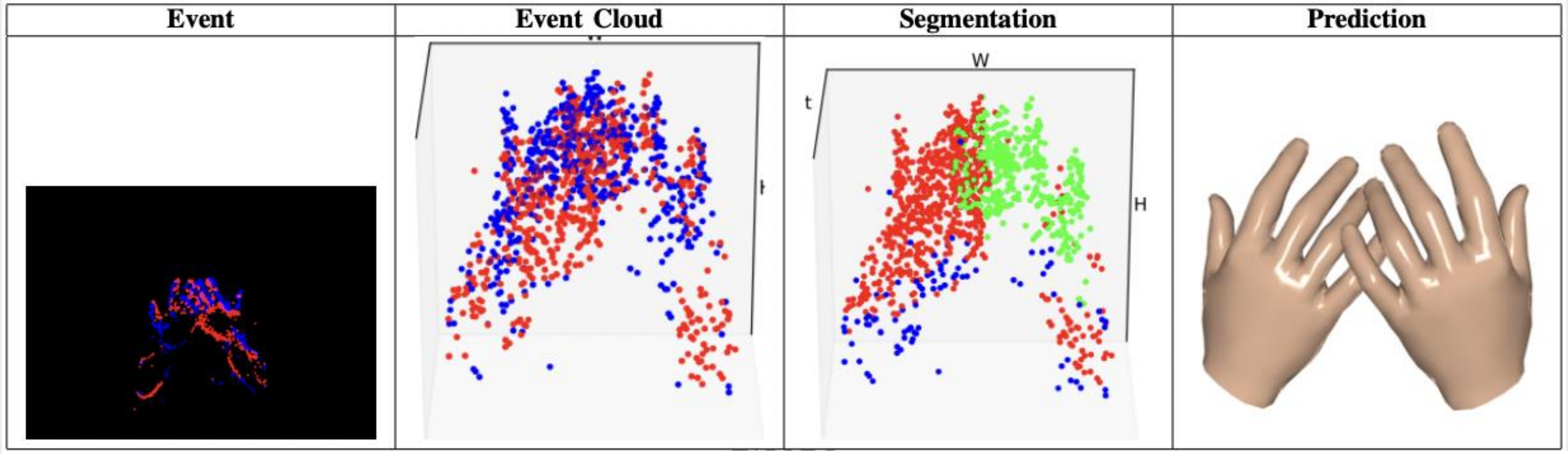
SETUP

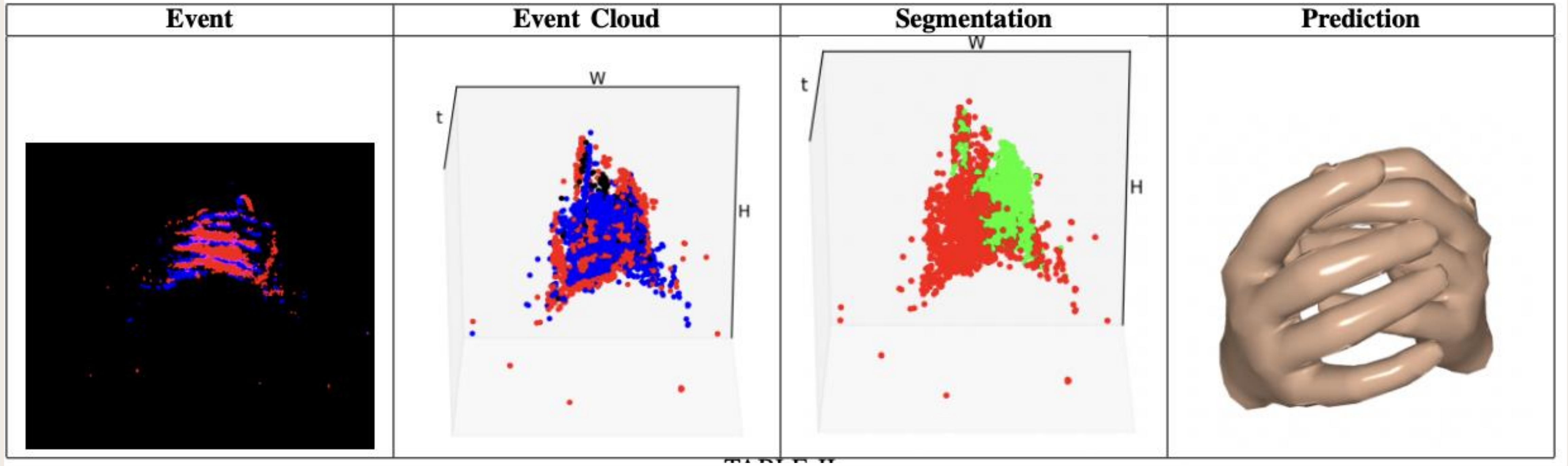


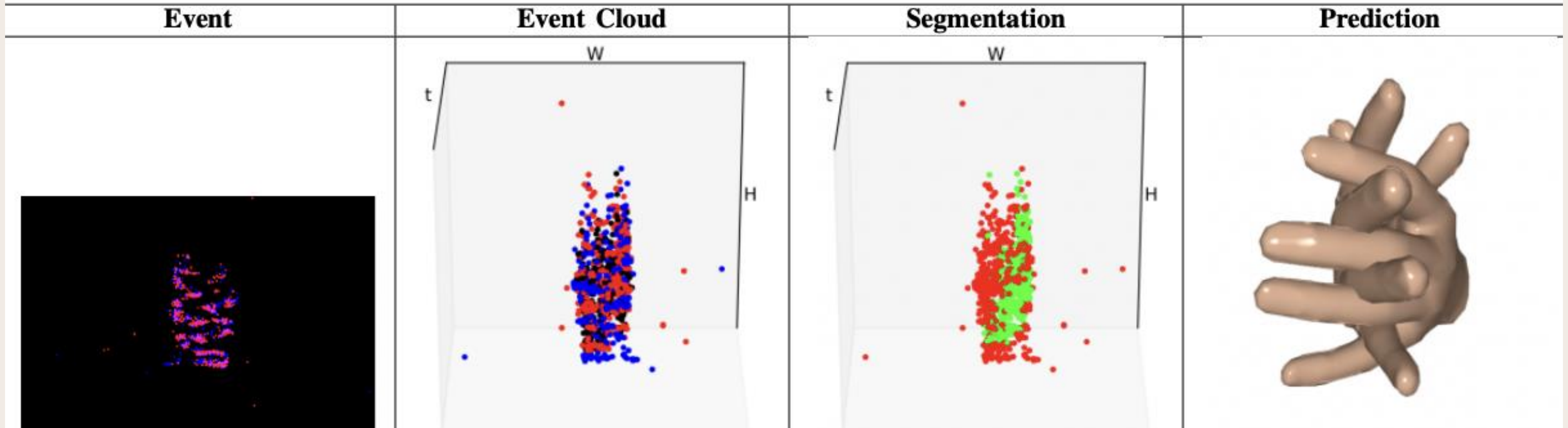
- The experimental setup consists of an event camera mounted securely on a tripod with a suction base for stability.
- The camera is connected to a laptop via a USB interface, facilitating data capture and processing in real-time.
- The output was given by the lab engineer Murad as ros bags
- Ros bags were used to be converted to mp4 videos

RESULTS

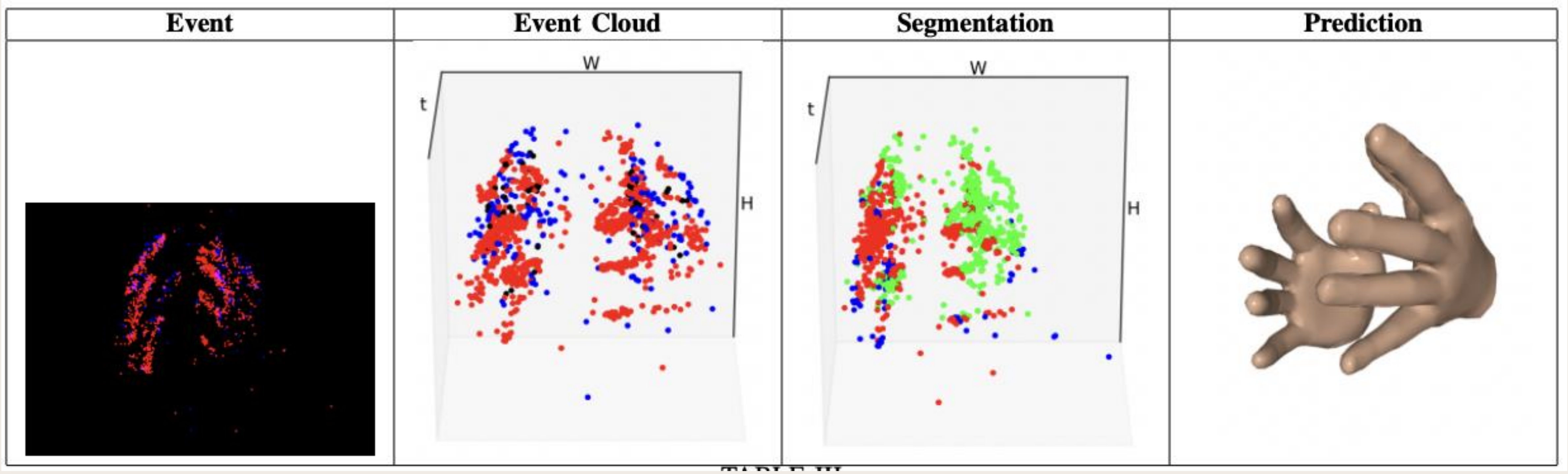
DIFFERENT HAND GESTURE EVENTS





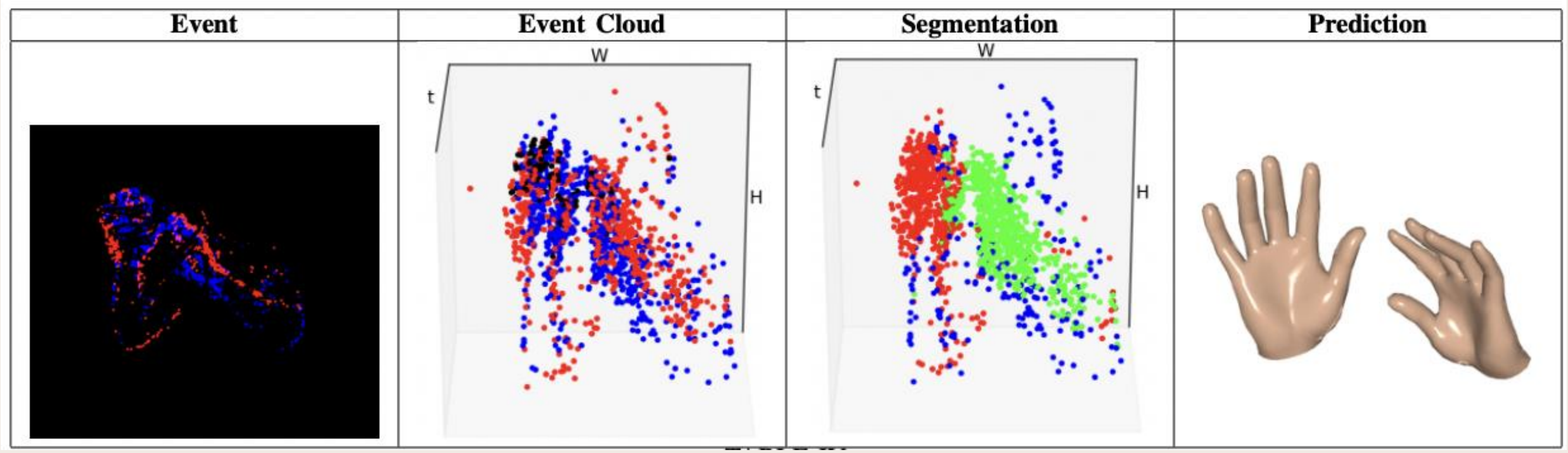


DIFFERENT HAND GESTURE EVENTS



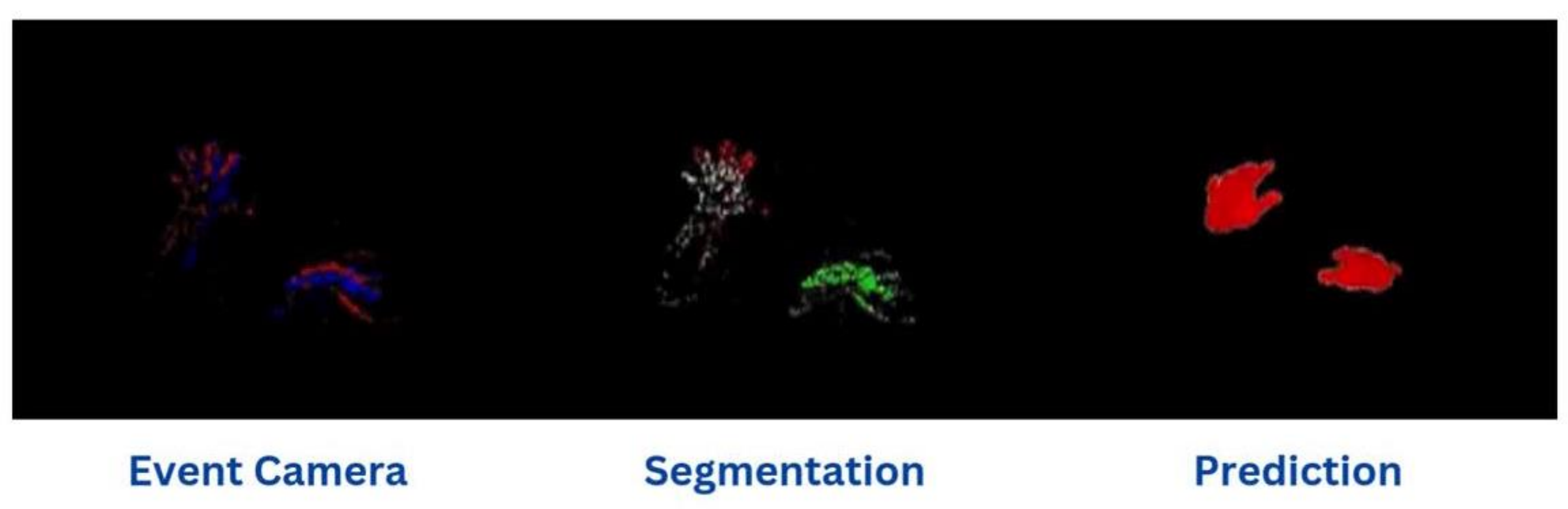
Event	Event Cloud	Segmentation	Prediction

TABLE III

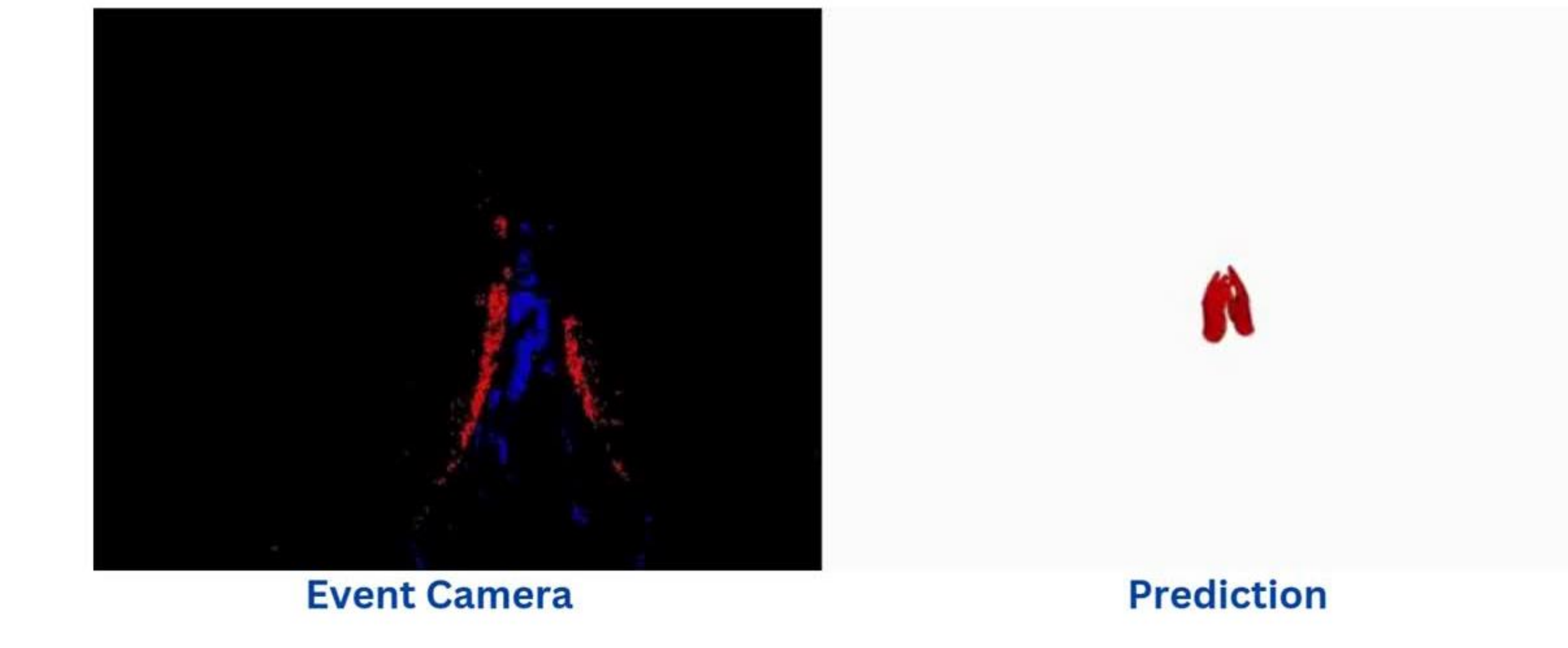


Event	Event Cloud	Segmentation	Prediction

DEMO VIDEO 1



DEMO VIDEO 2



STRENGTHS

STRENGTHS

High Temporal Resolution

- Utilizes event cameras capable of capturing changes at microsecond intervals, making it highly effective for fast and dynamic hand gestures.

Robustness to Motion Blur

- Event cameras inherently avoid motion blur by capturing only changes, ensuring accurate hand pose estimation even during rapid hand movements.

Real-Time Performance

- The framework is optimized for low-latency applications, making it suitable for real-time hand gesture recognition

Handles complex scenarios

- Handles complex hand motions and interactions effectively, showcasing adaptability to real-world scenarios.

LIMITATIONS AND FUTURE WORK

LIMITATIONS AND FUTURE WORK

Fixed Camera Assmption

- The current approach assumes a stationary camera, which simplifies the problem but limits usability in dynamic or mobile setups.

Background Clutter

- Event data generated by moving objects or changes in the background can introduce noise, reducing segmentation accuracy.

Future work

Moving Camera Integration:

- ○ Extend the framework to handle data from moving or portable event cameras, addressing challenges of background clutter and motion compensation.

RGB and Event Data Fusion:

- Combine event data with traditional RGB streams to leverage the strengths of both modalities:
  - ○ Increased visual fidelity from RGB.
  - ○ Low latency from event data.

THANK YOU !

REFERENCES

[1] Seungryul Baek, Kwang In Kim, and Tae-Kyun Kim. Pushing the envelope for rgb-based dense 3d hand pose estimation via neural rendering, 2019.
[2] Adnane Boukhayma, Rodrigo de Bem, and Philip H. S. Torr. 3d hand shape and pose from images in the wild, 2019.
[3] Liuhao Ge, Zhou Ren, Yuncheng Li, Zehao Xue, Yingying Wang, Jianfei Cai, and Junsong Yuan. 3d hand shape and pose estimation from a single rgb image, 2019.
[4] Franziska Mueller, Florian Bernard, Oleksandr Sotnychenko, Dushyant Mehta, Srinath Sridhar, Dan Casas, and Christian Theobalt. Ganerated hands for real-time 3d hand tracking from monocular rgb, 2017.
[5] Zicong Fan, Adrian Spurr, Muhammed Kocabas, Siyu Tang, Michael J. Black, and Otmar Hilliges. Learning to disambiguate strongly interacting hands via probabilistic per-pixel part segmentation, 2021.
[6] Shile Li and Dongheui Lee. Point-to-pose voting based hand pose estimation using residual permutation equivariant layer, 2018.
[7] Franziska Mueller, Micah Davis, Florian Bernard, Oleksandr Sotnychenko, Mickeal Verschoor, Miguel A. Otaduy, Dan Casas, and Christian Theobalt. Real-time pose and shape reconstruction of two interacting hands with a single depth camera. ACM Trans. Graph., 38(4), July 2019.
[8] Andrea Ceccarelli and Francesco Secci. Rgb cameras failures and their effects in autonomous driving applications. IEEE Transactions on Dependable and Secure Computing, 20(4):2731–2745, 2023.
[9] Lan Xu, Weipeng Xu, Vladislav Golyanik, Marc Habermann, Lu Fang, and Christian Theobalt. Eventcap: Monocular 3d capture of high-speed human motions using an event camera, 2019.
[10] Viktor Rudnev, Vladislav Golyanik, Jiayi Wang, Hans-Peter Seidel, Franziska Mueller, Mohamed Elgharib, and Christian Theobalt. Eventhands: Real-time neural 3d hand pose estimation from an event stream, 2021.
[11] Qinyi Wang, Yexin Zhang, Junsong Yuan, and Yilong Lu. Spacetime event clouds for gesture recognition: From rgb cameras to event cameras. 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), pages 1826–1835, 2019.
[12] Jiaan Chen, Hao Shi, Yaozu Ye, Kailun Yang, Lei Sun, and Kaiwei Wang. Efficient human pose estimation via 3d event point cloud. In 2022 International Conference on 3D Vision (3DV), pages 1–10, 2022.
[13] Christen Millerdurai, Diogo Luvizon, Viktor Rudnev, Andre Jonas, Jiayi ´ Wang, Christian Theobalt, and Vladislav Golyanik. 3d pose estimation of two interacting hands from a monocular event camera, 2023.