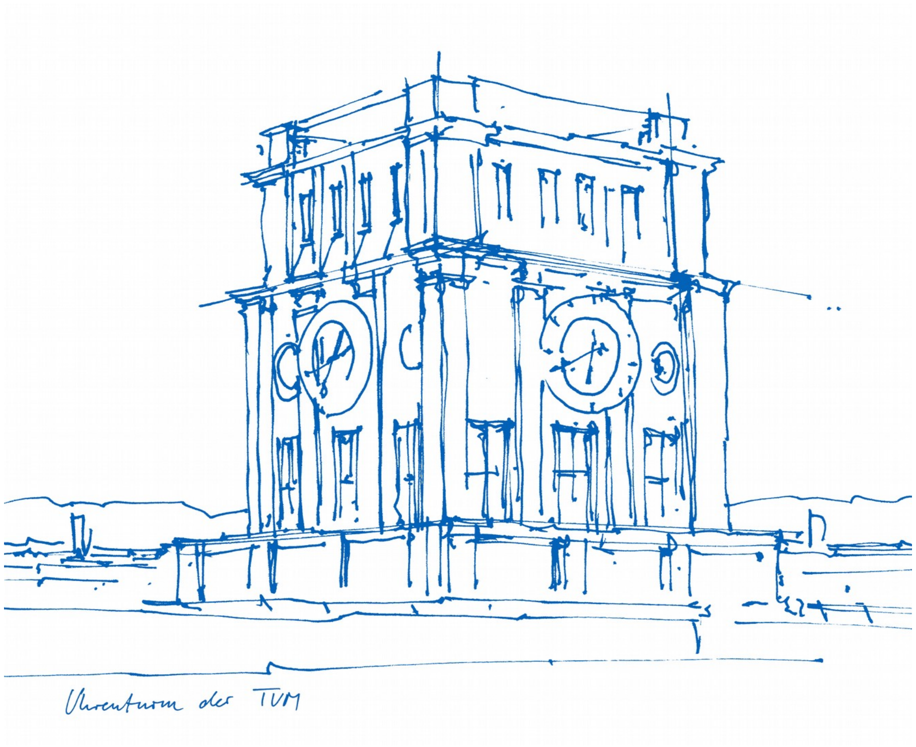
Event-based 3D Reconstruction on a Snake Robot

Shristi Mudgal, Rishabh Raj, Kanstantsin Tkachuk

Technische Universität München

8 February 2019



Event camera

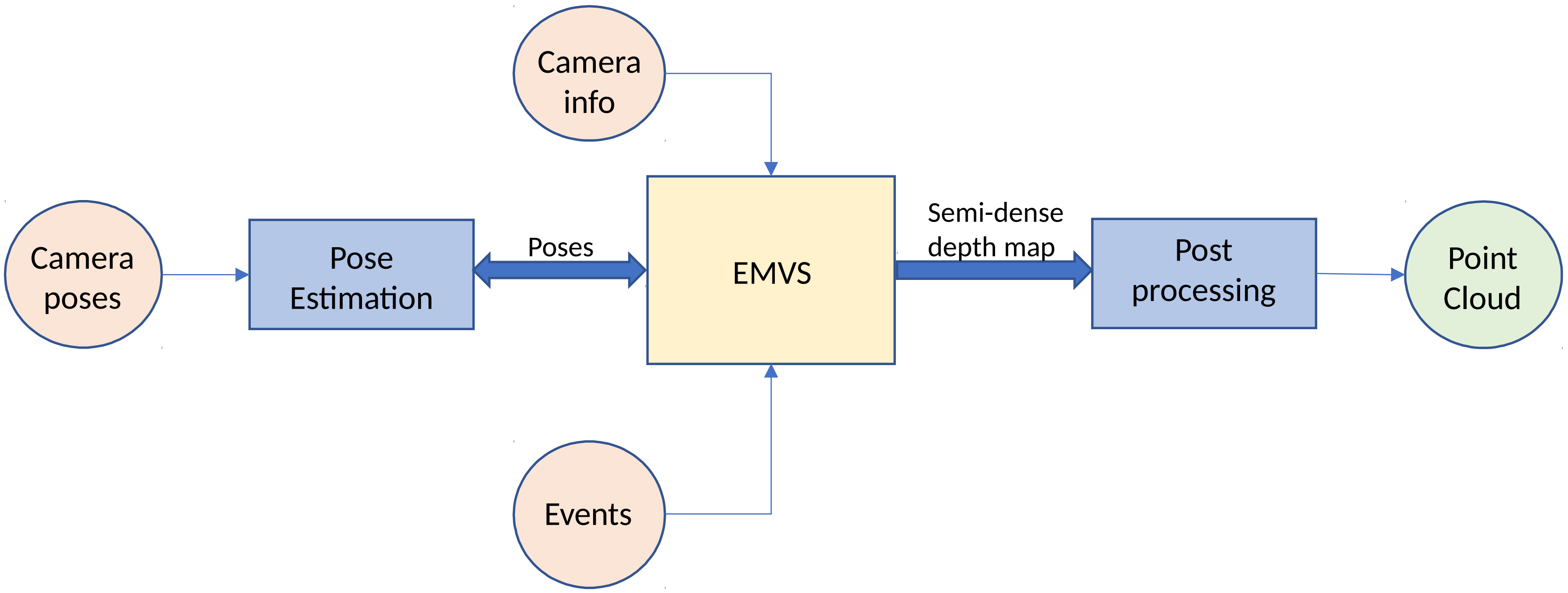
- Naturally highlights edges
- Edges trigger events from multiple viewpoints
- Outputs an asynchronous, sparse event stream at microsecond resolution
- Information contained in an event:
    - Location <x,y>
    - Timestamp in microseconds <t>
    - Polarity <p>

Why don‘t existing methods for camera work?

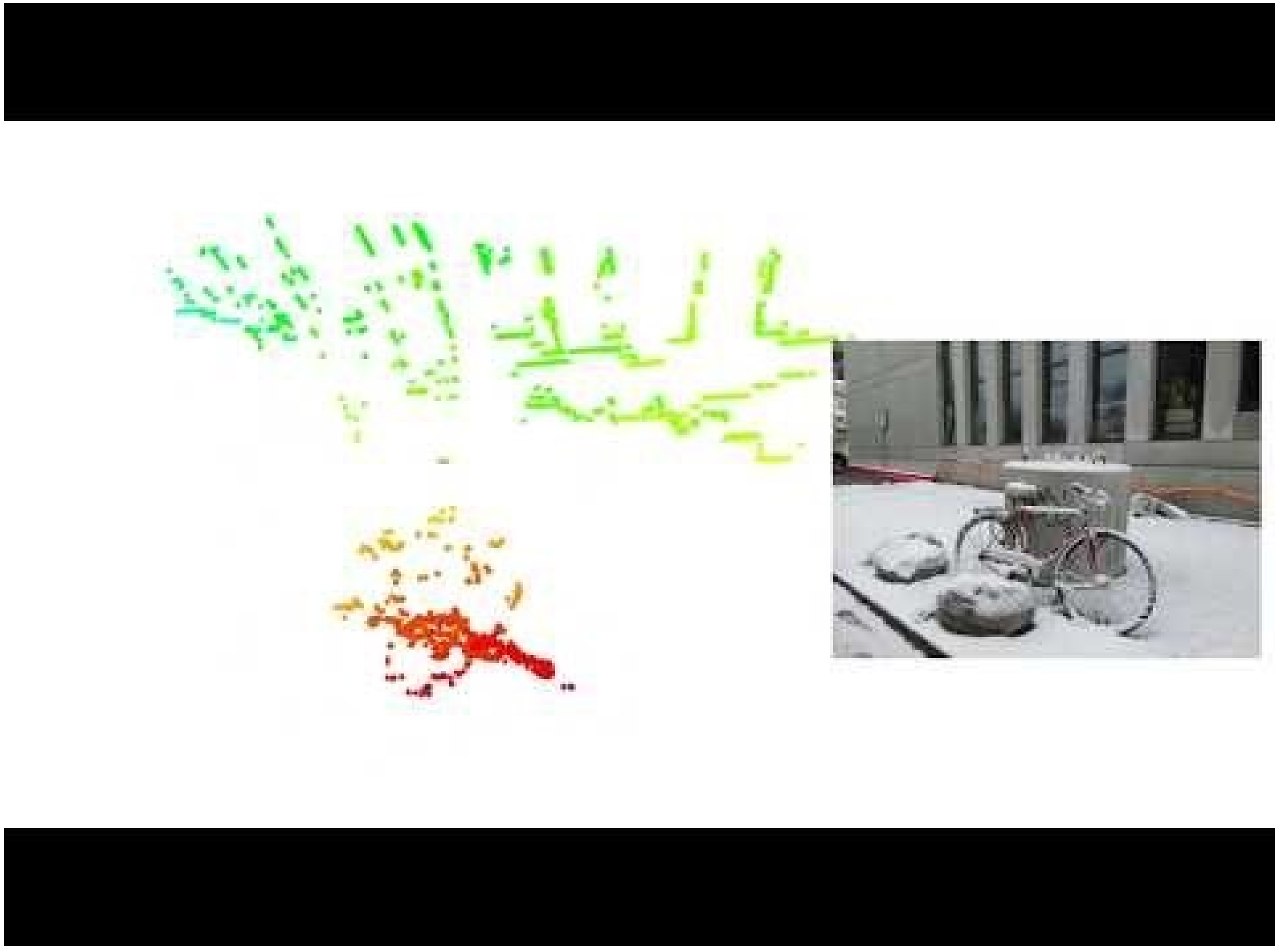
	Multiview Stereo for Camera	Event based Multiview Stereo
Input	• Full images • Dense • Dynamic scene not required • Camera motion not required	• Stream of asynchronous events • Sparse • Requires dynamic scene or camera motion to generate events • Requires camera motion for reconstruction
Approach	Densely populated DSI	DSI with holes
Output	Dense map/3D reconstruction	Semi-dense map/3D reconstruction

Event-based multi-view stereo

- Input: events, camera information, camera poses
- Output: semi dense point cloud

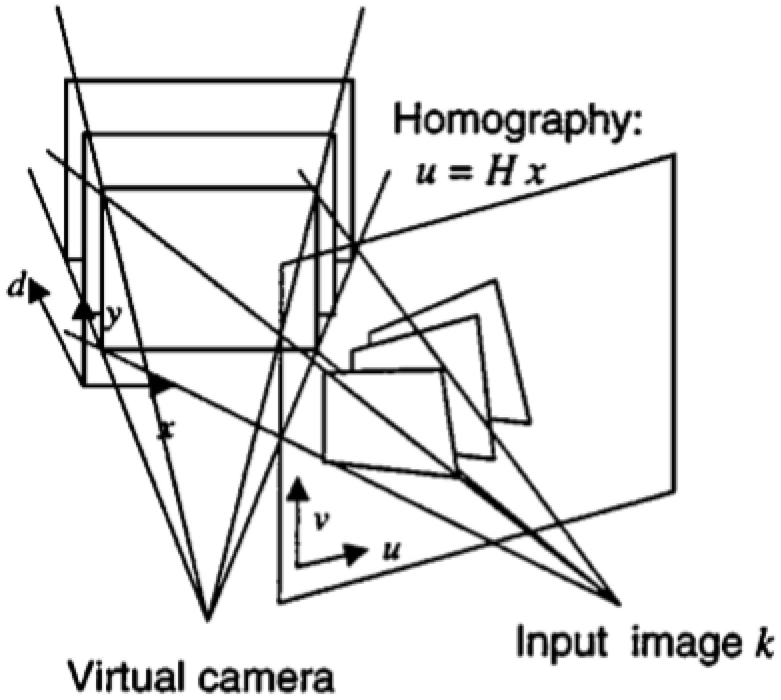


Expected results – demonstration by authors



Disparity space image

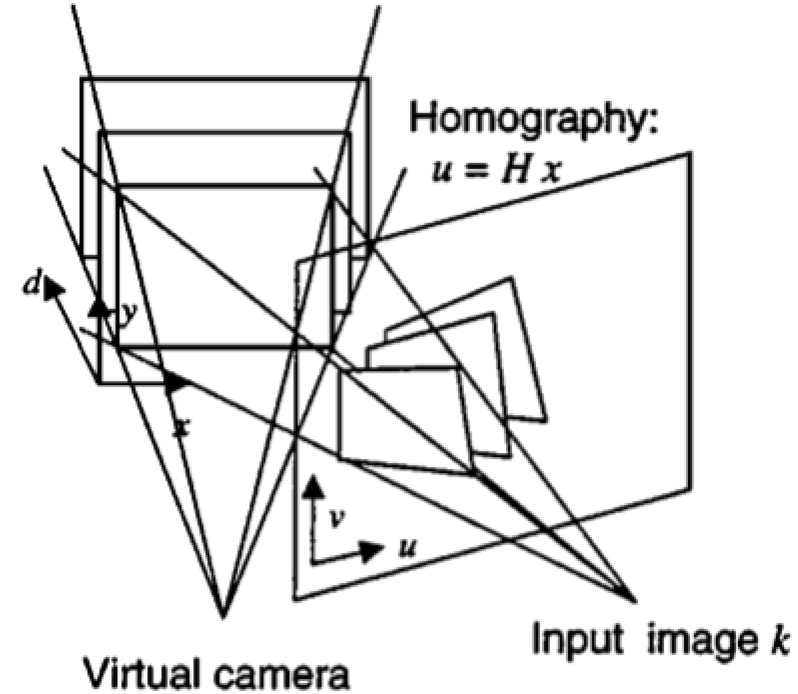
- A generalized disparity volume space
- Projective sampling of discretized volume
- Choose a virtual camera pose
- Choose orientation and spacing of disparity planes
  - Our case: equidistant spacing
- Steps correspond to scaling
  - Our case: Pixels too are scaled up

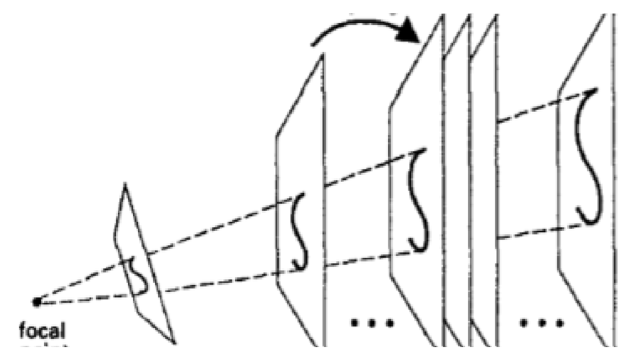


Creating a DSI volume

- Choose virtual camera positions
- DSI's defined at the virtual camera poses
- Create N equidistant depth planes
- Size of DSI: w x h x N
  - *w* x *h* : resolution of the event camera
- Planar Homography from event camera to virtual camera:

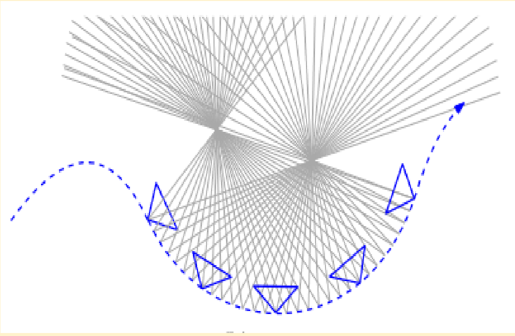
$$\mathrm{H}_{Z_i}^{-1} \sim \mathrm{R} + \frac{1}{Z_i}\mathbf{t}\mathbf{e}_3^{\top}.$$



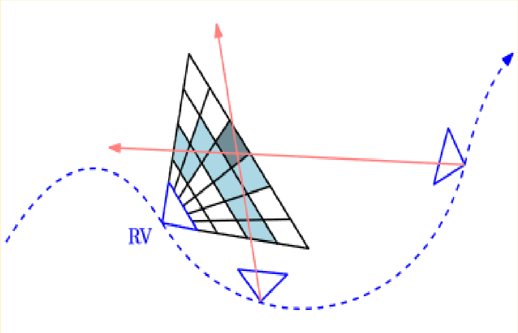


EVMS

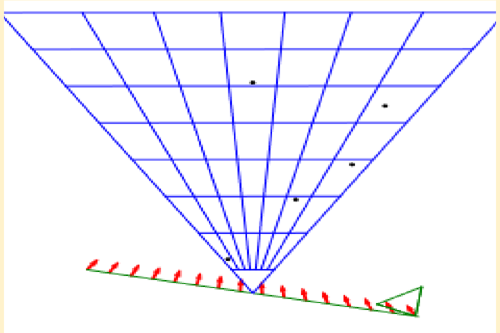
Event Back projection



Volumetric Ray Counting



Detecting Scene Structure



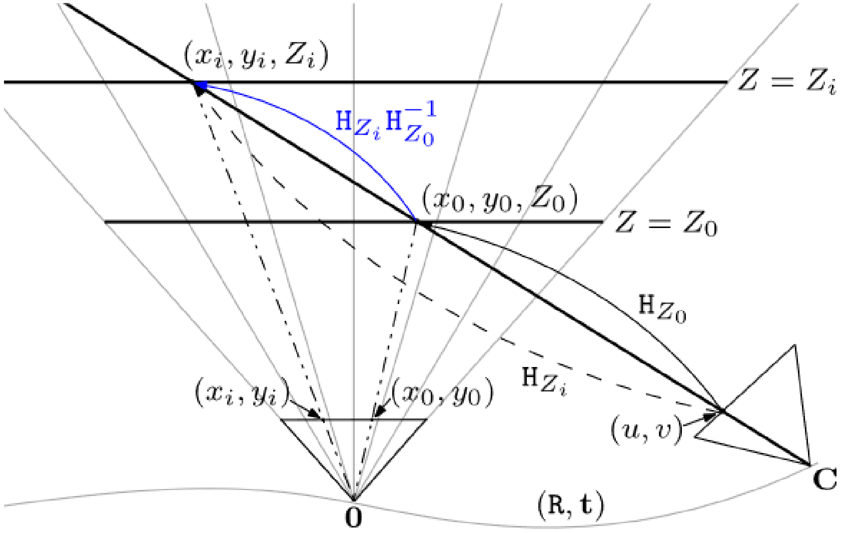
Event projection

Map points from the event camera to the virtual camera in two steps:

- Find homography HZ0 for the Z0 plane and transform the event coordinates as follows:

$$(x(Z_0), y(Z_0), 1)^\top \sim \mathrm{H}_{Z_0}(u, v, 1)^\top$$

- Find homography from the Z0 plane to the Zi plane and transform the coordinates of events at Z0 plane to the Zi plane as follows:



Event back projection

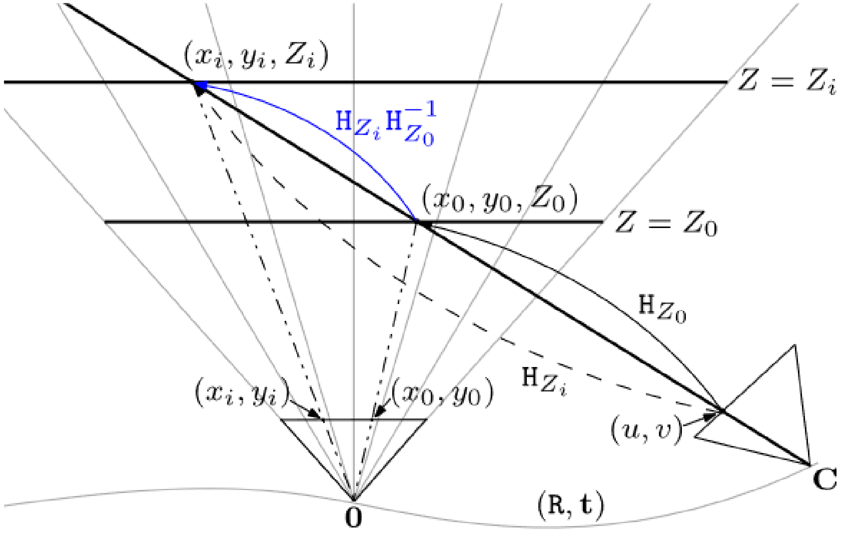
Map points from the event camera to the virtual camera in two steps:

- Find homography HZ0 for the Z0 plane and transform the event coordinates as follows:

- Find homography from the Z0 plane to the Zi plane and transform the coordinates of events at Z0 plane to the Zi plane as follows:

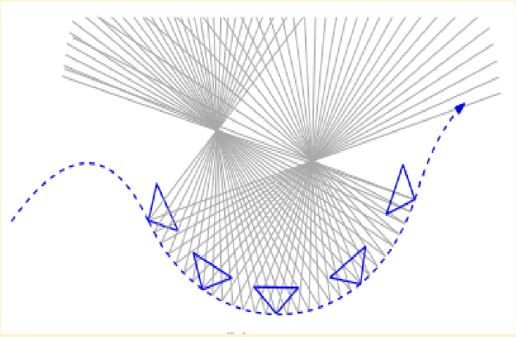
$$\mathrm{H}_{Z_i}^{-1} \sim \mathrm{R} + \frac{1}{Z_i}\mathbf{t}\mathbf{e}_3^\top.$$

$$(x(Z_i), y(Z_i), 1)^\top \sim \mathrm{H}_{Z_i}\mathrm{H}_{Z_0}^{-1}(x(Z_0), y(Z_0), 1)^\top$$

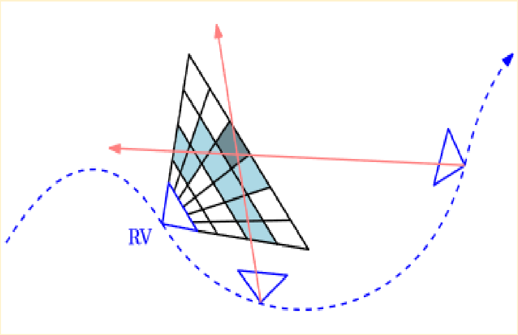


EVMS

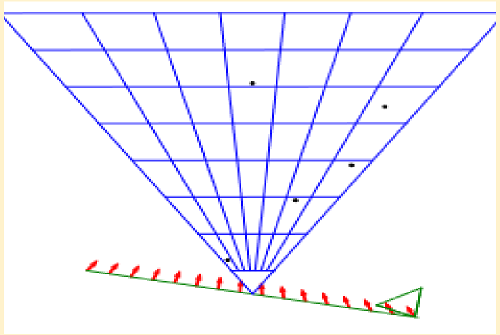
Event Back projection



Volumetric Ray Counting



Detecting Scene Structure

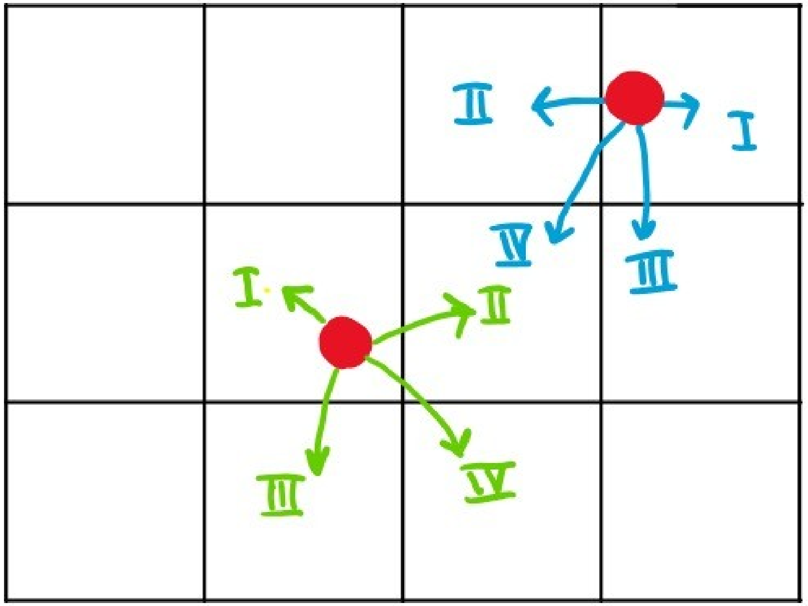


Volumetric ray counting

- Ray density function:

$$f(\mathbf{X}) : V \subset \mathbb{R}^3 \rightarrow \mathbb{R}^+$$

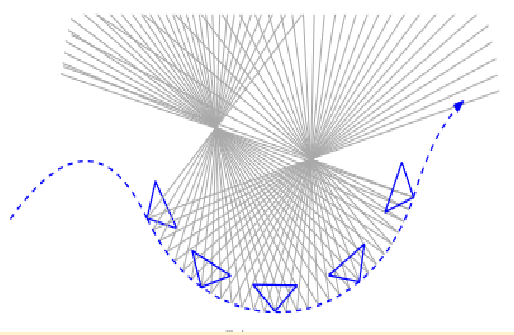
- Forward Mapping:
    - Nearest neighbour:
      $(x_j(Z_i), y_j(Z_i))$ votes for a single cell of a depth plane
    - Bilinear Voting:
      $(x_j(Z_i), y_j(Z_i))$ votes for 4 nearest cells splitting its votes depending on the distances to the cell locations



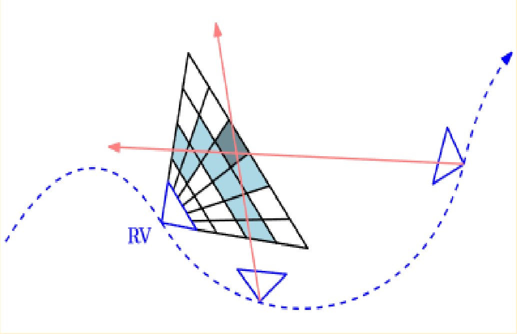
Bilinear Voting

EVMS

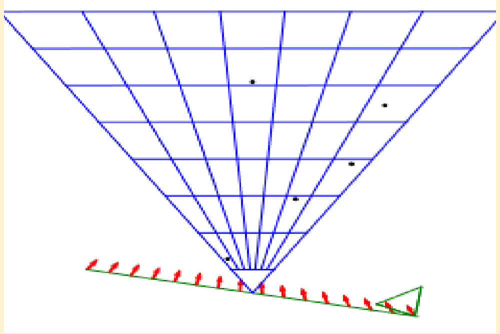
Event Back projection



Volumetric Ray Counting

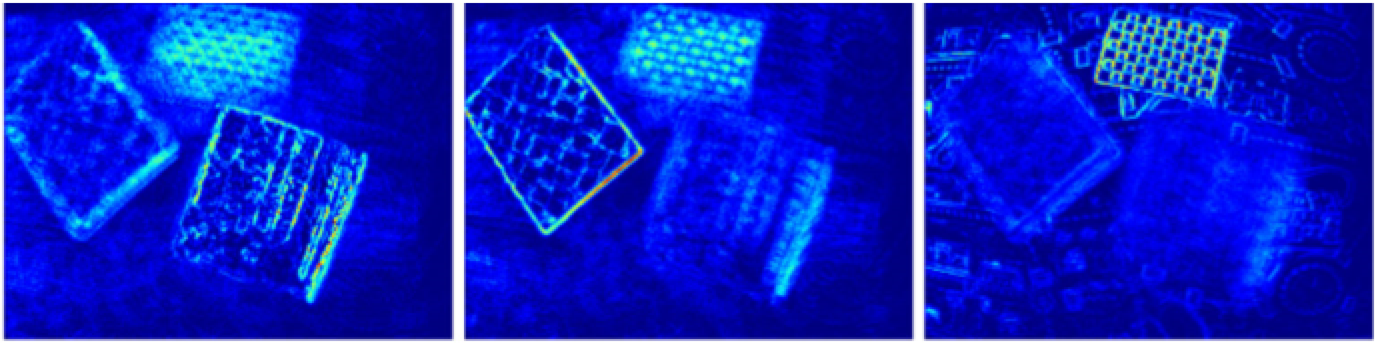


Detecting Scene Structure



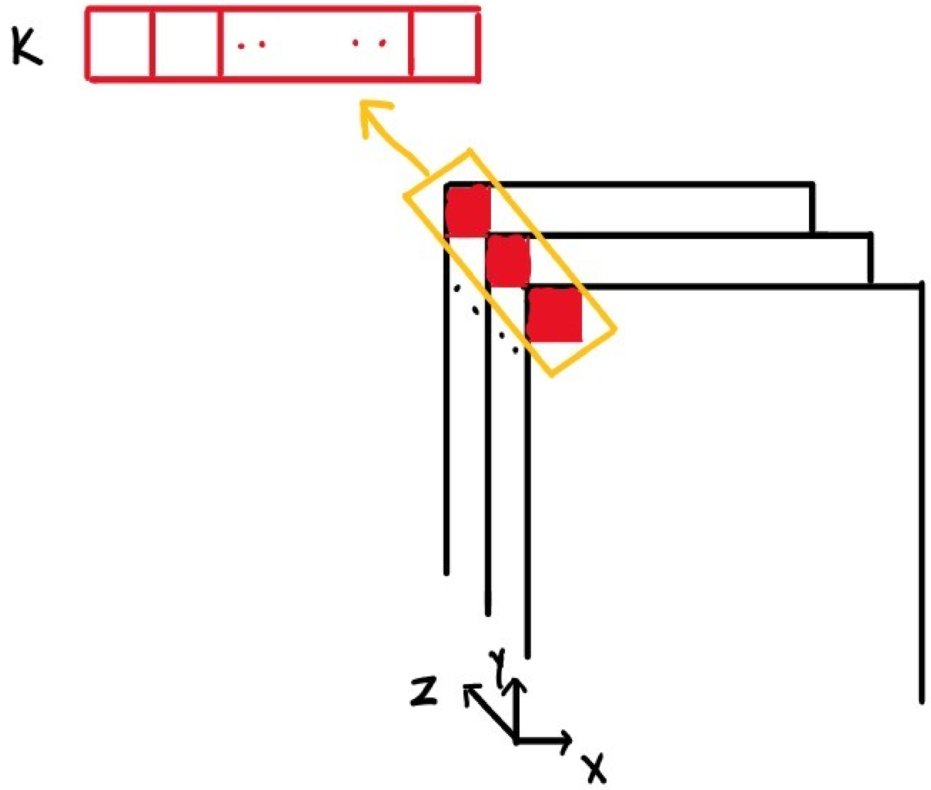
Maximization of ray density

- Scene points occur at local maximas of f(X)
- Detecting local maximas:
  - Generate a depth map and a confidence map
  - Threshold the confidence map
  - Select the confident pixels in the depth map



Various depth slices

Generating depth map and confidence map



Confidence map ← max k

Depth map ← argmax k

Adaptive Gaussian thresholding

$$dst(x,y) = \begin{cases} \texttt{maxValue} & \text{if } src(x,y) > T(x,y) \\ 0 & \text{otherwise} \end{cases}$$

T(x,y) = weighted sum of the kxk neighborhood of (x,y) - C

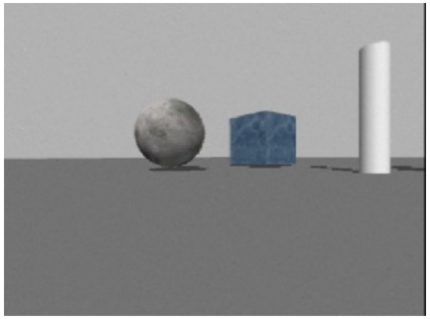
Advantages:

1. Reconstruction is invariant in case of dynamic scenes
2. Reconstruction is invariant to brigntness and illumination changes
3. Reconstruction is free from noise

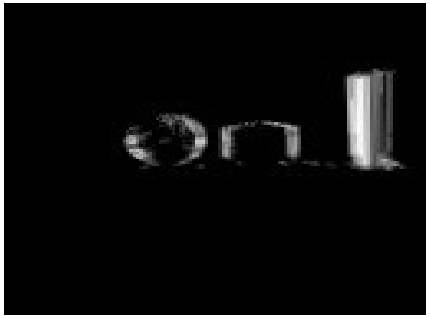
Post-processing

1. Merge the depth maps from all virtual camera positions
2. Apply a median filter on it to remove outliers
3. Convert to point cloud
4. Filter point cloud to remove outliers – radius filter / statistical outlier filter

Results

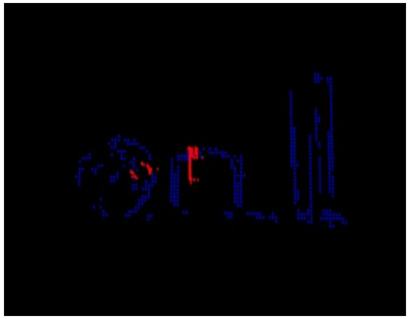


Scene



Confidence Map

Depth Map



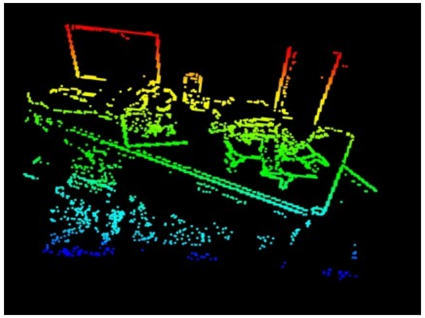
Point Cloud

Results

Scene

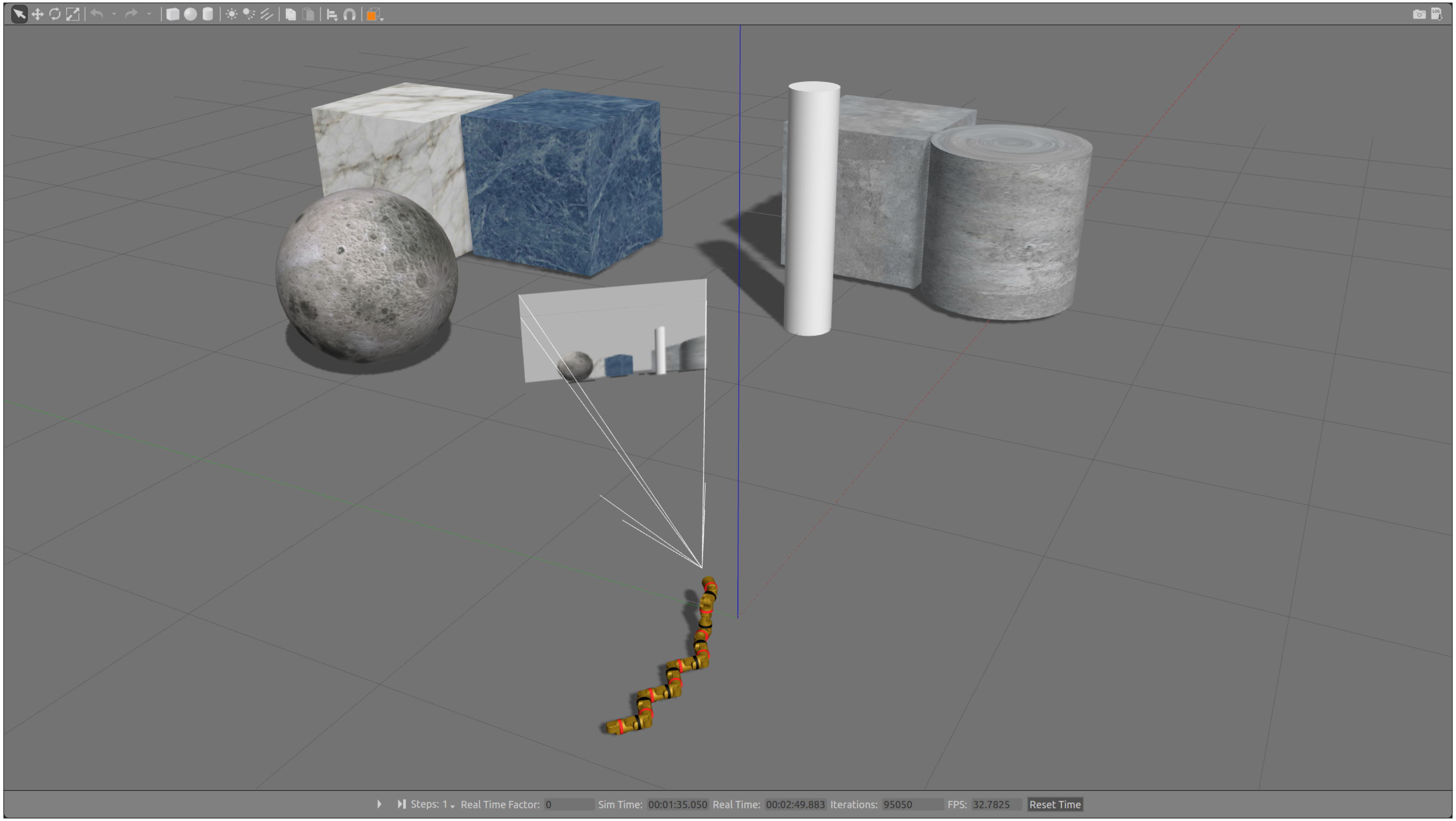
Confidence Map

Depth Map

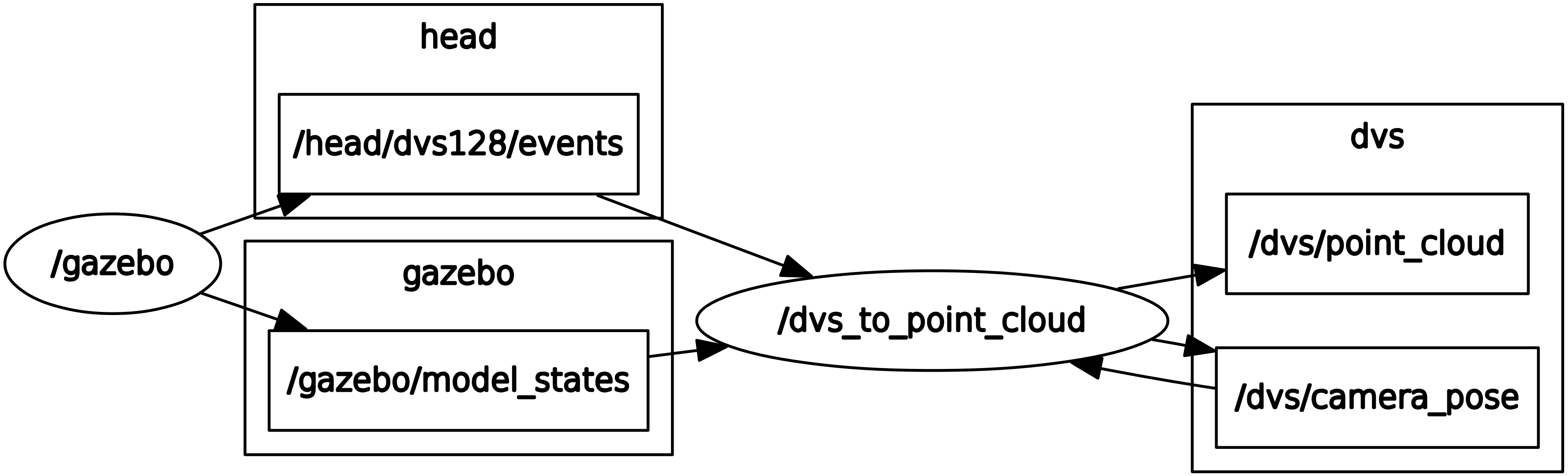


Point Cloud

Real-Time Implementation



ROS infrastructure



Pipeline

Thread 1 – obtain camera poses from Gazebo and publish them with timestamps:

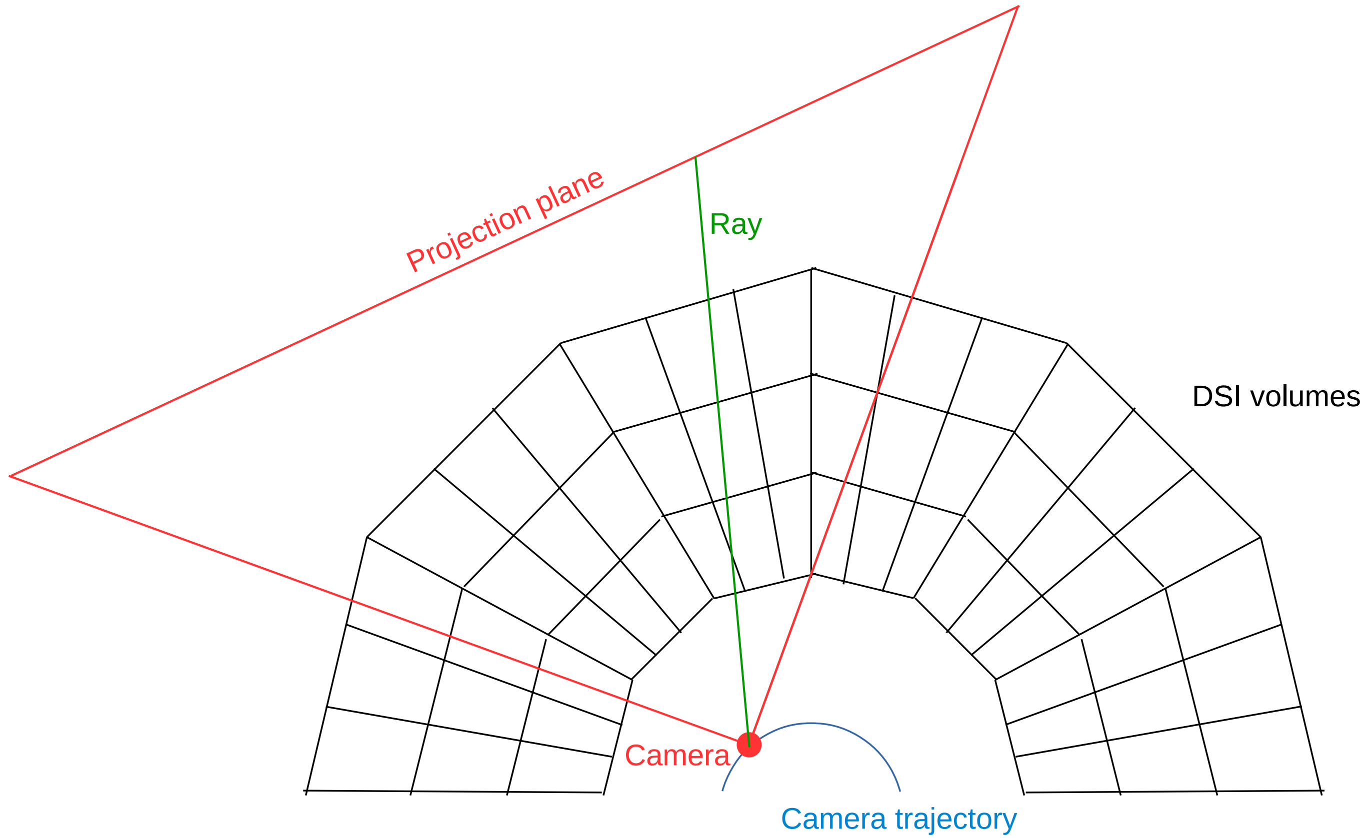
- listen to `/gazebo/model_states` topic for changes in model states
- extract info about the state of the snake model
- get the pose of the snake‘s head
- create a stamped message and publish it on `/dvs/camera_pose` topic

Thread 2 – sync events and poses, then calculate and publish the point cloud:

- messages are synchronized using `ApproximateTime` from `message_filters` library
- extract the events
- create an array of endpoints using back projection
- cast the rays through the DSIs (6 projective voxel grids)
- update scores in the DSIs
- create a depth image for each DSI
- refine depth images
- calculate coordinates of points from depth images and publish the point cloud

Disretizing the 3D scene

View on the scene from above



References

Rebecq, H., Gallego, G., Mueggler, E., Scaramuzza, D. (2017). EMVS: Event-Based Multi-View Stereo—3D Reconstruction with an Event Camera in Real-Time. *Int J Comput Vis*, 126: 1394. https://doi.org/10.1007/s11263-017-1050-6