EvAC3D: From Event-based Apparent Contours to 3D Models via Continuous Visual Hulls

Ziyun Wang*, Kenneth Chaney*, and Kostas Daniilidis

University of Pennsylvania, Philadelphia PA 19104, USA

Abstract. 3D reconstruction from multiple views is a successful computer vision field with multiple deployments in applications. State of the art is based on traditional RGB frames that enable optimization of photo-consistency cross views. In this paper, we study the problem of 3D reconstruction from event-cameras, motivated by the advantages of event-based cameras in terms of low power and latency as well as by the biological evidence that eyes in nature capture the same data and still perceive well 3D shape. The foundation of our hypothesis that 3D-reconstruction is feasible using events lies in the information contained in the occluding contours and in the continuous scene acquisition with events. We propose Apparent Contour Events (ACE), a novel event-based representation that defines the geometry of the apparent contour of an object. We represent ACE by a spatially and temporally continuous implicit function defined in the event x-y-t space. Furthermore, we design a novel continuous Voxel Carving algorithm enabled by the high temporal resolution of the Apparent Contour Events. To evaluate the performance of the method, we collect MOEC-3D, a 3D event dataset of a set of common real-world objects. We demonstrate EvAC3D's ability to reconstruct high-fidelity mesh surfaces from real event sequences while allowing the refinement of the 3D reconstruction for each individual event. The code, data and supplementary material for this work can be accessed through the project page: https://www.cis.upenn.edu/~ziyunw/evac3d/.

1 Introduction

Traditional 3D reconstruction algorithms are frame-based because common camera sensors output images at a fixed frame rate. The fixed frame rate assumption challenges researchers to develop complex techniques to handle undesirable situations due to discontinuity between frames, such as occlusions. Therefore, recovering the association between views of the same object has been an essential problem in 3D reconstruction with a single camera. Such challenges fundamentally arise from the discrete time sampling of visual signals, which forces vision algorithms to recover the missing information between views. However, these problems do not exist naturally in biological systems because visual signals are encoded as a stream of temporally continuous spikes. Continuous encoding tremendously benefits humans and animals in many tasks, including estimating the 3D geometry of an object. The question is: *can a computer vision algorithm do better if it sees the same continuous world as humans do?*

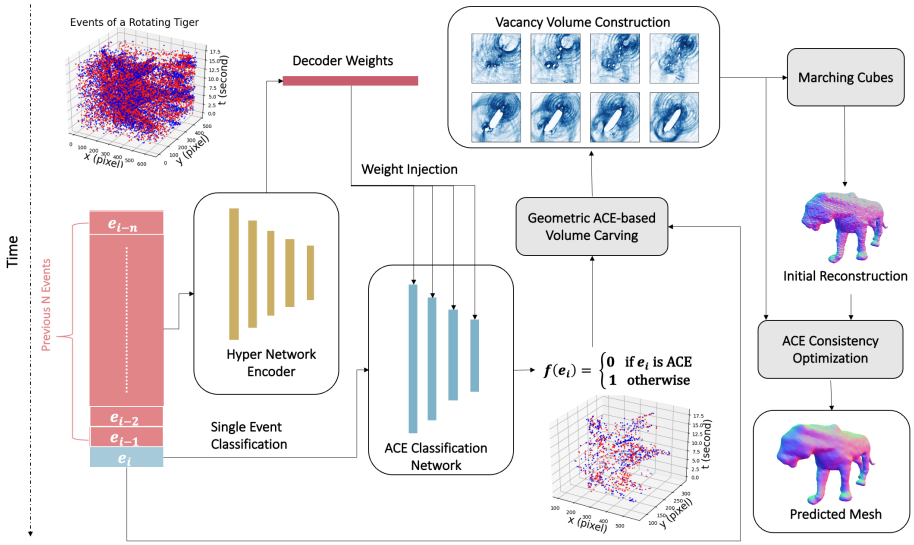

Fig. 1. EvAC3D Pipeline. We use the previous N events as conditional information to predict the label for the current event. A hyper network is used to inject the conditional information into the decoding classifier. The predicted label is then passed into a geometry-based volume event carving algorithm.

In this work, we seek the answer to this question by developing a novel algorithm for bio-inspired event-based cameras. Event-based cameras are novel visual sensors that generate a continuous stream of events triggered by the change of the logarithm of the light intensity. The events are asynchronously generated without temporal binning; therefore, the high-resolution temporal information can be completely recovered from each event with minimal discontinuity. Additionally, the individual pixels of the camera do not have a global shutter speed, which gives the camera extremely high dynamic range. Due to the high dynamic range and high temporal resolution of event cameras, they have become an ideal choice for understanding fast motions. For 3D reconstruction, traditional cameras operate on a fixed frame rate. For image-based visual hull methods, the limited number of views means the smooth surfaces of the object cannot be properly reconstructed, which can be seen from the sphere reconstruction example in Figure 2.

Ideally, one can expect to directly perform incremental updates to the geometry of an object at the same high temporal resolution as events. To this end, we propose a 3D reconstruction pipeline that directly predicts a mesh from a continuous stream of events assuming known camera trajectory from a calibrated camera. We introduce a novel concept of **Apparent Contour Events** to define the boundary of an object in the continuous x-y-t space. Through Apparent Contour Events, we incrementally construct the function of a 3D object

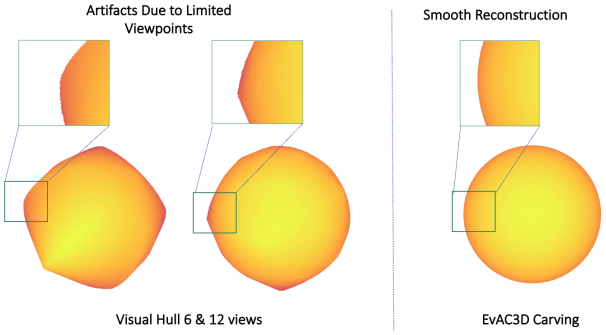

Fig. 2. Reconstruction of a sphere with visual hull (6 and 12 frames) and with EvAC3D reconstruction on simulated events. The 12-view visual hull method uses roughly the same number of operations as EvAC3D.

surface at the same high temporal resolution as events. Here is a list of our main contributions:

- We introduce a novel event concept of **Apparent Contour Events** that relates high-speed events to the tangent rays of the 3D surface to the viewpoint.
- We propose a learning pipeline to predict which events are Apparent Contour Events without manual annotation but using 3D models of known objects. We propose a novel event-based neural network with point-based decoding to classify the Apparent Contour Events.
- We present a continuous algorithm to reconstruct an object directly from a stream of events. The algorithm can accurately reconstruct objects with complex geometry in both synthetic and real environments.
- We collect MOEC-3D, a high-quality real 3D event dataset for evaluating the performance of 3D reconstruction techniques from events that provides events, ground-truth 3D models, and ground-truth camera trajectories.

2 Related Work

3D Reconstruction with Event Cameras Due to the asynchronous and sparse nature of the event sensors, 3D reconstruction algorithms cannot be directly applied. Most current work in event-based 3D reconstruction uses a stereo pair of cameras [13,4,28,29]. The time coincidence of the events observed from a synchronized pair of cameras is used for stereo matching. These methods work in situations where multiple calibrated cameras are used synchronously. Zhu et al. [29] construct a cost volume based on warping using multiple disparity hypotheses. Carneiro et al. [4] use time incidence between two synchronized event streams to perform stereo matching. Chaney et al. [5] use a event single camera in motion to learn the 3D structure of the world. E3D [2] attempts to directly predict meshes from multi-view event images. This method is trained and mainly

evaluated on synthetic data due to the large amount of 3D data needed for training. EMVS [20] adopts an event-based ray counting technique. Similar to our method, EMVS treats individual events as rays to take advantage of the sparse nature of the event data. In Section 3.3, we show how sparse processing can be extended further to work with only a particular type of events that contain rather rich geometric information.

3D Reconstruction from Occluding Contours. Reconstruction from the perspective projection of a geometric surface has been extensively studied in classical computer vision. Among different geometric representations used in such problems, apparent contour representation is most relevant to our work. Apparent contours, or extreme boundaries of the object, contain rich information about a smooth object surface. Barrow et al. [1] argue that surface orientations can be directly computed from a set of apparent contours in line drawings. Cipolla et al. [7] propose the theoretical framework from reconstructing a surface from the deformation of apparent contours. Based on the idea of contour generator [16], the projection of the apparent contours onto the image plane are used as tangent planes to the object. Furthermore, structure and motion can be simultaneously recovered from apparent contours. Wong et al. [26] propose to solve the camera poses and 3D coordinates of "frontier points", the intersection of the apparent contours in two camera views. A circular motion with a minimum of 3 image frames is assumed to solve the optimization problem.

Visual Hull. Visual hull is used to reconstruct 3D objects through Shape-From-Silhouette (SFS) techniques [3,14]. Information from multiple views are aggregated into a single volume through intersection of the projective cones described by the silhouette at each view. Voxel grid and octrees [12,23] are commonly used as discretized volumetric representations. SFS methods are particularly susceptible to false-negative labels (labeling an interior point as an exterior point).

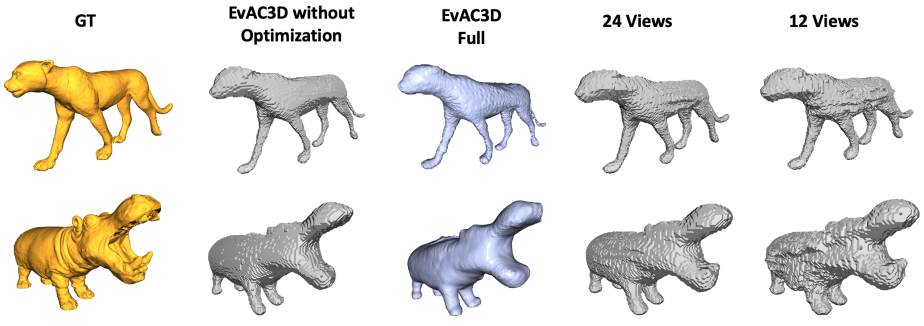

Fig. 3. Qualitative comparisons between EvAC3Dand mask based carving of 12 and 24 views respectively. Cheetah, hippo, and elephant were chosen as a subset of the animal scans.

3 Method

In this section, we explain how a continuous stream of events can be used to reconstruct the object surface. We divide the pipeline into two stages: **Apparent Contour Event Extraction** and **Continuous Volume Carving**.

3.1 Apparent Contour Event (ACE)

The main challenge in reconstructing objects from events is finding the appropriate geometric quantities that can be used for surface reconstruction. In frame-based reconstruction algorithms, silhouettes are used to encode the rays from the camera center to the object. However, computing silhouettes requires integrating frame-based representations, which limits the temporal resolution of the reconstruction updates. Additionally, since events represent the change in log of light intensity, events are only observed where the image gradients are nonzero. Therefore, one would not observe enough events on a smooth object surface. These two facts combined make traditional silhouettes non-ideal for events. To address these two shortcomings, we introduce **Apparent Contour Events (ACE)**, a novel representation that encodes the object geometry while preserving the high temporal resolution of the events.

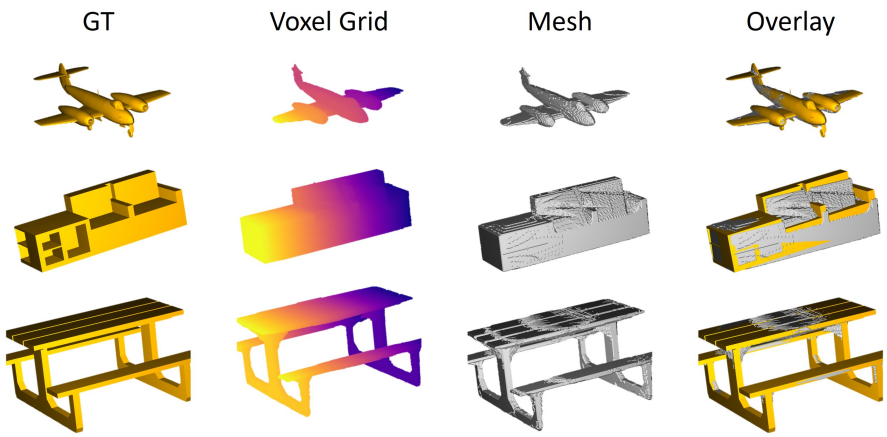

Fig. 4. Qualitative evaluations from ShapeNet using EvAC3D on three categories of objects.

Geometrically, the generator of occluding contours on image planes is constrained by a ray-surface intersection and tangency [9]. A smooth surface $\mathcal{S}$ with well defined surface normals at each point has an occluding contour generator for each camera center ${}^{\mathbf{w}}\mathbf{p_c}$. The contour generator is composed of image rays

that intersect the S at exactly one point ${}^{\mathbf{w}}\mathbf{X}$. A surface point ${}^{\mathbf{o}}\mathbf{x}$ with normal ${}^{\mathbf{w}}\mathbf{n}$ is included in the contour generator for the camera center ${}^{\mathbf{w}}\mathbf{p_c}$ if for an image ray ${}^{\mathbf{w}}\mathbf{v}$ the ray-surface intersection and tangency constraints hold [9]:

$$\lambda {}^{\mathbf{o}}\mathbf{v} + {}^{\mathbf{w}}\mathbf{p_c} = {}^{\mathbf{w}}\mathbf{X} \tag{1}$$

$${}^{\mathbf{w}}\mathbf{n}^{\intercal}({}^{\mathbf{w}}\mathbf{X} - {}^{\mathbf{w}}\mathbf{p_c}) = 0 \tag{2}$$

We define Apparent Contour Events formally. ACEs are events that meet the ray-surface intersection and tangency constraints [9]. Since each event can contain a potentially unique timestamp, the constraints must be thought of in continuous time, as opposed to indexible on a per frame basis. An event e_i generates an image ray ${}^{\mathrm{c}}\mathbf{x}(\mathrm{t})$ at a camera center ${}^{\mathrm{w}}\mathbf{p}_{\mathrm{c}}(\mathrm{t})$. e_i is an ACE if for some point ${}^{\mathrm{w}}\mathbf{X}(\mathrm{t})$ on the surface $\mathcal{S}$:

$${}^{\mathrm{w}}\mathbf{n}(\mathrm{t})^{\intercal}({}^{\mathrm{w}}\mathbf{X}(\mathrm{t}) - {}^{\mathrm{w}}\mathbf{p}_{\mathrm{c}}(\mathrm{t})) = 0 \tag{3}$$

$$\lambda(t)\,{}^{\mathrm{w}}\mathbf{R}_{\mathrm{c}}(\mathrm{t})\,{}^{\mathrm{c}}\mathbf{x}(\mathrm{t}) + {}^{\mathrm{w}}\mathbf{p}_{\mathrm{c}}(\mathrm{t}) = {}^{\mathrm{w}}\mathbf{X}(\mathrm{t}) \tag{4}$$

Intuitively, ACE can be seen as the set of events $e_i = \{x_i, y_i, t_i, p_i\}$ that belong to the active contour of the object at time t_i. Due to the contrast between the active contour of an object with the background, a significant number of events are generated around the contour. Unlike silhouettes, which require filling in holes on the "eventless" areas of an integrated image, an ACE is defined purely on events. Traditional algorithms are limited by the frame rate of the input images. Projecting rays from only through the contours produces far fewer intersections of the rays. With events, we can shoot a ray for each event, which continuously refine the geometry around the active contour, as shown in Figure 1. To fully take advantage of the continuous nature of the events, we design a novel continuous volume carving algorithm based on single events, as described in Section 3.3.

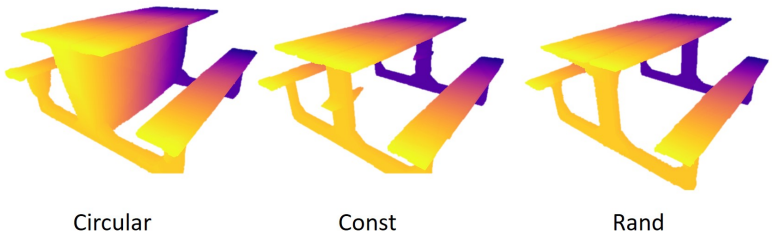

Fig. 5. Comparison of different trajectories in simulation with ShapeNet. The circular and octahedral trajectories only move around major axes missing some contours that would improve the carving results. In comparison, the random trajectory samples evenly across the sphere providing more unique viewpoints.

3.2 Learning Apparent Contour Events

We formulate identification of Apparent Contour Events (ACEs) as a classification problem. In other words, the network learns a function $F_{E_{t_i}}$, which maps an event to whether it is an ACE conditioned on the history of events E_{t_i}. For an event $e_i = \{x_i, y_i, t_i, p_i\}$, we encode the past N events using a function θ as a K dimensional latent vector $C_i \in \mathbf{R}^K$, where K is a hyperparameter.

$$C_i = g_\phi(\{e_j \coloneqq (x_j, y_j, t_j, p_j)\}) : j > \max(i - N, 0) \tag{5}$$

N is a hyperparameter that specifies the history of events as the conditional input to the classification problem. The ACE classification problem is modeled as a function that maps from the latent code and an event to the probability that it is an ACE:

$$q_i = f_\theta(e_i, C_i) \tag{6}$$
$$e_i \coloneqq (x_i, y_i, t_i, p_i) \tag{7}$$
$$q_i \in [0, 1] \tag{8}$$

We use a neural network to parameterize function g_ϕ and f_θ. Note that g_ϕ takes a list of N events. In practice, we use an event volume [30] to encode past events.

$$E(x, y, t) = \sum_i p_i k_b(x - x_i) k_b(y - y_i) k_b(t - t_i^*) \tag{9}$$

We chose this representation because the values in such volumes represent the "firing rate" of visual neurons in biological systems, which preserves valuable temporal information. The temporal information is needed because labeling ACEs requires the network to predict both where the contours are in the past and how they move over time. To supervise the ACE network, we jointly optimize the encoder and the event decoder using a Binary Cross-Entropy loss directly on the predicted event labels.

$$\mathcal{L}_c = \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}_{bce}(f_\theta(e_i, C_i), \hat{q}_i)) \tag{10}$$

Here $\hat{q}_i$ is the ground truth event label for e_i. In practice, the labels are extremely imbalanced especially in either low light conditions (high noise to signal ratio) or scenes where other objects are moving as well. For training, we equally sample half positive and half negative events to help overcome the imbalance of labels.

Architecture To enable classification of individual events, we adopt an encoder-decoder architecture where the decoder maps event coordinates to probabilities. These types of architectures are widely used in learning-based single-view 3D reconstruction methods. Mapping approaches such as AtlasNet [10] and implicit approaches (Occupancy Networks [17], DeepSDF [19]) all use variants of this

Algorithm 1 Event Carving Algorithm

Input V volume initialized to zero
Input E active contour events
Input ${}^{w}\mathbf{R}_{c}(t), {}^{w}\mathbf{p}_{c}(t)$ camera trajectories

procedure CarveEvents(V, E, ${}^{w}\mathbf{R}_{c}(t)$, ${}^{w}\mathbf{p}_{c}(t)$)
  for $i \leftarrow 1, |E|$ **do**
    $(x_i, y_i, t_i, p_i) \leftarrow E_i$
    ${}^{V}T_{C(t_i)} \leftarrow {}^{V}T_{W}\,{}^{W}T_{C(t_i)}$
    $O_i \leftarrow {}^{V}\mathbf{R}_{W}\,{}^{w}\mathbf{p}_{c}(t_i) + {}^{V}\mathbf{t}_{W}$
    $D_i \leftarrow {}^{V}\mathbf{R}_{W}\,{}^{w}\mathbf{p}_{c}(t_i)\,{}^{w}\mathbf{x}_{c}(t_i)$
    $V_i \leftarrow bresenham3D(O_i, D_i, bounds(V))$
    $V[V_i] += 1$
  end for
  return V
end procedure

architecture. In our experiments, we find the decoder part of the network has more weight in the overall mapping performance. Rather than taking fixed-sized latent vector code, we inject the conditional information directly into the weights of the decoder, following [11,17,25,24,18]. We use the Conditional Batch Normalization to inject the encoding of the prior events into the Batch Normalization layers of the decoder network. The architecture of the network is illustrated in Figure 1. The training details and hyperparameters of the network can be found in the Supplementary Material.

3.3 Event Based Visual Hull

In frame-based shape from silhouette and space carving approaches, the goal is to recover the visual hull, defined as the intersection of visual cones that are formed by the apparent contour in each frame. A better definition, though following the original definition by Laurentini [14] would be the largest possible volume consistent with the tangent rays arising from apparent contour events. The visual hull is always a superset of the object and a subset of the convex hull of the object. Due to the continuity of the camera trajectory and the high temporal sampling of events, we expect the obtained visual hull to be tighter to the object than the visual hull obtained from a sparse set of viewpoints that might be closer to the convex hull of the object.

Continuous Volume Carving provides smooth continuous incremental changes to the carving volume. This is accomplished by only carving updates through the use of ACEs. This creates a more computationally efficient as shown in Table 3.

ACEs are defined by the tangent rays to the surface at any given positional location. The time resolution of event based cameras provide ACEs that are from continuous viewpoints through the trajectory of the camera. These continuous viewpoints, $C(t)$, are from around the object in the world frame, W. Projecting

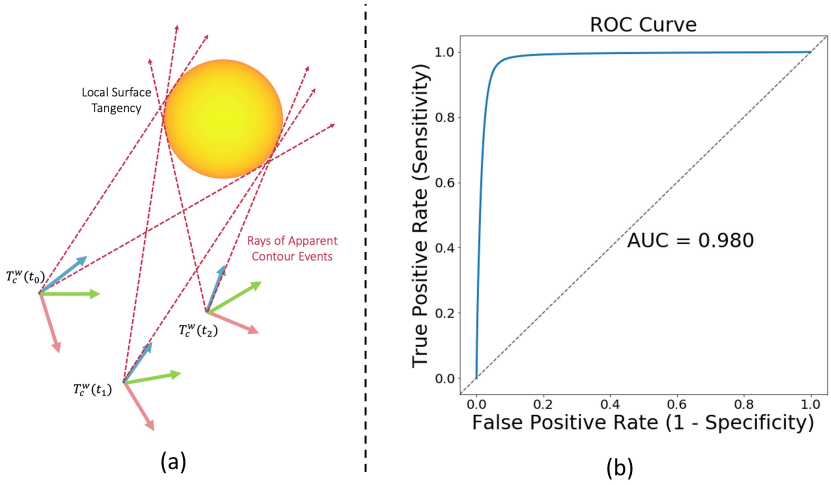

Fig. 6. (a): illustration of carving based on Apparent Contour Events. (b): ROC curve of ACE classification network.

an individual ACE, into the voxel grid coordinate system gives us a ray with origin, ${}^{V}R_{W}\,{}^{\mathrm{w}}\mathbf{p}_{\mathrm{c}}(\mathrm{t}) + {}^{V}\mathbf{t}_{W}$, and direction, ${}^{V}R_{W}\,{}^{\mathrm{w}}\mathbf{R}_{\mathrm{c}}(\mathrm{t})\,{}^{\mathrm{c}}\mathbf{x}(\mathrm{t})$. The ray in the voxel grid coordinate system allows us to project through the volume, which is illustrated in Figure 3.3 (a). To efficiently traverse this volume, a 3D Bresenham algorithm is used to produce a set of voxel coordinates, $\mathcal{V}_i \in \mathbb{Z}^3$, along the ray. All voxels in $\mathcal{V}_i$ are incremented. The interior of the object is left empty, as the rays trace along the continuous surface of the object. This can be seen at the bottom of Figure 1. Algorithm 1 covers the process of generating updates to the voxel grid for every event individually. The object's mesh is then extracted from the volume (algorithm in the supplementary material) and optimized.

3.4 Global Mesh Optimization

The mesh reconstructed from volume carving can be affected by noise either from pose estimation or sensor noise. Specifically, the object will look "smaller" if some rays erroneously carve into the object due to noise. Consequently, we optimize the consistency between the proposed mesh and the high-confidence cells in the vacancy volume, which we call "high-confidence" surface points. We propose a global optimization to further refine the mesh based on these points. Recall that most rays intersect at the surface of the objects. Define point set $\hat{Y}$ as the point set of all high-confidence surface points of $V(x, y, z)$:

$$\hat{Y} = \{(x, y, z) : V(x, y, z) > \epsilon_V\} \tag{11}$$

where ϵ_V is a threshold based on the carving statistics of volume V. For a mesh reconstructed from running Marching Cubes, represented as a graph $G = (P, E)$,

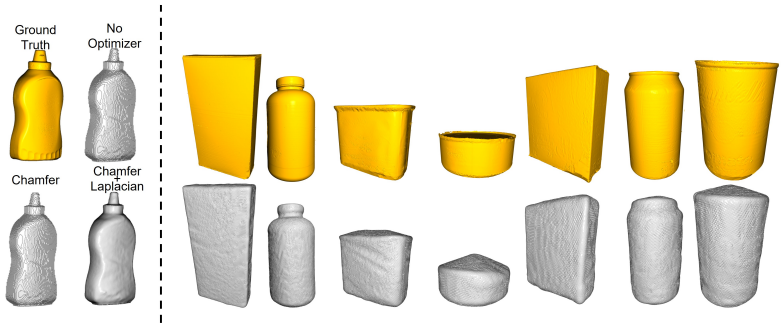

Fig. 7. (Left) Comparison between options available within the optimizer. (Right) Test performance of event-based carving using predicted ACEs from our network.

where P is the set of vertices and E is the set of edges that form the faces. A deformation function f maps the original vertex set P to a deformed set $P' = f(P)$. We first optimize a one-side Chamfer distance from the high-confidence surface points (less than ϵ distance away) to the mesh vertices. In addition, we regularize the mesh by a graph Laplacian loss. The final objective can be written as:

$$L_{rf} = \lambda_1 \frac{1}{|P'|} \sum_{\substack{p'_i \in P' \\ ||p'_i - \hat{y}||_2 < \epsilon_d}} \min_{\hat{y} \in \hat{Y}} ||p'_i - \hat{y}||_2^2 + \lambda_2 \frac{1}{|P'|} \sum_{p'_i \in P'} \sum_{p'_j \in \mathcal{N}(P')} \frac{1}{\mathcal{N}(P'_i)} ||p'_j - p'_i||_2$$

where $\mathcal{N}(P'_i)$ represents all neighbors of a node P'_i, and λ_1 and λ_2 are the weights between the two losses. We find a function f that minimizes the loss. The f function can be treated as the point-wise translation of the vertices. All values used in this optimization come from our predictions without using the ground truth. We use Adam Optimizer to optimize the warping function f.

4 Experiments

In this section, we present the data collection details, evaluation of the carving algorithm, and reconstruction of real objects.To better evaluate the performance of event-based 3D reconstruction algorithms, we collect Multi Object Event Camera Dataset in 3D (MOEC-3D), a 3D event dataset of real objects. Please refer to the Supplementary Material for details about the dataset. For ground truth models, an industrial-level Artec Spider scanner is used to provide the ground truth 3D models with high accuracy. The detailed steps of data collection can be found in the Supplementary Material.

4.1 Evaluation Metrics

We report both Chamfer distance and Cosine similarity of the mesh compared to the ground truth model. Chamfer distance is measuring the average distance

Table 1. Event Carving Evaluation This table contains the results using ground truth Apparent Contour Events (ACEs). Chamfer distance (lower is better) is reported in $10^{-3}m$ (millimeters). Surface normal (higher is better) is reported as cosine similarity between the ground truth and predicted surface normal. We sample 10,000 points uniformly both on the reconstructed mesh and the object mesh.

	Chamfer Distance↓			**Normal Consistency↑**		
Category	EvAC3D	Mask-24	Mask-12	EvAC3D	Mask-24	Mask-12
Mustard	**3.0164**	4.6210	5.5161	**0.9619**	0.9034	0.9035
Coffee	**2.1439**	2.2926	3.3019	0.9826	**0.9893**	0.9877
Soda (b)	1.5231	**1.4601**	2.0635	**0.9834**	0.9717	0.9735
Jello (s)	**0.9657**	2.1973	4.3524	**0.9801**	0.9766	0.9234
Jello (b)	5.9083	**4.4409**	7.7409	0.8843	**0.9541**	0.8952
Tuna	**3.2633**	3.7045	4.3070	0.9598	**0.9665**	0.9644
Soup	**1.6513**	2.0130	2.8556	0.9653	**0.9705**	0.9681
Sugar	**0.8651**	2.4071	4.9491	**0.9935**	0.9862	0.9405
Vitamin	2.6190	**1.4478**	2.4836	0.9683	**0.9947**	0.9896
Spam	**2.0398**	3.1969	4.8615	0.9739	**0.9760**	0.9479
Mean	**2.4267**	3.2652	4.3856	**0.9487**	0.9377	0.9159

between two point clouds, which reflects the positional accuracy of the reconstruction. It is defined as:

$$CD(X, \hat{X}) = \frac{1}{|X|} \sum_{x \in X} \min_{\hat{x} \in \hat{X}} ||x - \hat{x}||_2 + \frac{1}{|\hat{X}|} \sum_{\hat{x} \in \hat{X}} \min_{\hat{x} \in X} ||\hat{x} - x||_2 \tag{12}$$

X and $\hat{X}$ represent the points sampled from the reconstruction and the ground truth model. Surface normal is also a commonly used metric for comparing the geometry of two meshes. We report the average cosine similarity between the corresponding surface samples of two meshes, which is defined as:

$$Cos.Sim(X_{gt}, X_{pred}) = \frac{1}{|X_{gt}|} \sum_{i \in |X_{gt}|} |\vec{n_i} \cdot \vec{m}_{\theta(x, X_{pred})}| \tag{13}$$

$$\theta(x, X_{gt} \coloneqq \{(\vec{y_j}, \vec{m_j})\})) = \arg\min_{j \in |X_{gt}|} ||x - y_j||_2^2 \tag{14}$$

We use the closest point to approximate the correspondence between two sets of oriented samples, similar to the argmin used in Equation 12. We use a k-nearest neighbor search to estimate the normals of sampled points on the mesh, where k is 300.

4.2 Evaluating Carving Algorithm

To test the effectiveness of our continuous carving algorithm, we utilize the meshes collected as part of this dataset within a simulation environment for fair comparisons. Note that the evaluation is done with real objects and we assume

Table 2. Real Object Reconstruction This table contains the results using trained network to predict Active Contour Events from real data. Chamfer distance (lower is better) is reported in $10^{-3}m$ (millimeters). Surface normal (higher is better) is reported as cosine similarity between the ground truth and predicted surface normal. "Mask" means using masks from the event mask network. "Image" means using masks predicted from reconstructed images. The number in each column name represents the number of views used for reconstruction.

	Chamfer Distance↓ / Surface Normal Consistency↑					
	EvAC3D	Mask 24	Mask 12	Image 24	Image 12	E3D [2]
Mus	**1.537**/**0.983**	3.034/0.968	7.061/0.926	4.192/0.868	6.947/0.926	7.986/0.713
Cof	**2.286**/0.957	2.653/**0.971**	7.771/0.915	5.733/0.840	5.930/0.821	8.354/0.756
Sod	2.239/**0.965**	**1.953**/0.957	4.611/0.929	2.380/0.914	3.865/0.884	6.762/0.703
Jel(s)	**2.889**/**0.928**	3.860/0.925	14.248/0.783	3.967/0.862	7.188/0.757	8.255/0.744
Jel(b)	**3.899**/**0.930**	4.818/0.926	13.405/0.750	3.929/0.863	6.657/0.736	14.910/0.767
Tun	**3.624**/**0.938**	3.518/0.937	5.552/0.753	4.254/0.863	8.545/0.734	10.850/0.704
Sou	**2.111**/**0.959**	2.392/0.954	5.200/0.887	2.294/0.922	5.276/0.854	6.133/0.783
Sug	**1.953**/**0.970**	7.904/0.854	9.929/0.833	4.000/0.939	9.724/0.775	5.924/0.691
Vit	**2.191**/0.957	2.338/**0.966**	5.772/0.949	2.226/0.958	5.710/0.915	8.462/0.715
Spa	**2.784**/**0.953**	3.667/0.945	9.635 /0.738	3.295/0.911	6.798/0.849	10.730/0.747
Mean	**2.551**/**0.954**	3.614/0.940	8.312/0.846	3.602/0.900	6.664/0.825	8.837/0.732

the ACEs are known at every point during the camera motion. This is different than the completely synthetic environment employed in [2] because the events generated with an event simulator are not guaranteed to have the same data distribution. We observe a significant amount of noise in the real event data. The quantitative results are provided in Table 1. The mask-based carving is done with ground truth masks as well for fair comparison.

In addition to the real data simulation above, we show ShapeNet examples, Figure 4, of our algorithm on objects with more complicated geometry to provide context for our reconstruction quality. To use these models, we use Open3D [27] to capture high frame rate images and ground truth masks. These images were then processed through ESIM [8] to generate a set of simulated events. To generate a close approximation of the real world dataset, a similar trajectory to the real world dataset was chosen.

4.3 Reconstructing Real Objects

Many network-based methods only work on simulated datasets because they require a large amount of labeled object-level 3D models. In addition, such networks cannot easily be adapted to work on real data. In comparison, the EvAC3D network can be trained on a small set of data because the labels could be obtained geometrically for events. We report the per-class performance evaluation in Table 2. For each object, we evaluate on an unseen sequence withheld from the training set. EvAC3D uses apparent contour events from the network output to perform carving. For baseline comparisons, we train two separate U-Net [22]

Table 3. Mean number of carving operations, mean Chamfer distance, and mean cosine similarity. With ACEs, our continuous carving method outperforms the other frame-based methods while using significantly fewer operations.

Method	Num of Ops↓	Chamfer↓	Normal ↑
GT-Mask-24	6,661,111	3.614	0.940
GT-Mask-12	3,331,536	8.312	0.846
Image-24	6,674,148	3.602	0.900
Image-12	3,345,580	6.664	0.825
E3D [2]	–	8.837	0.732
EvAC3D	**1,921,976**	**2.551**	**0.954**

style networks to output object masks from previous events and from reconstructed images. While they differ in input, they emulate the common situations where a fixed number of frames are used for reconstruction.

We follow the multi-view settings in 3D-R2N2 [6] where views are taken around the object. We choose 12 views as the baseline because we can reconstruct reasonable objects while keeping the computational cost close to EvAC3D. To further show the computational efficiency of EvAC3D, we also compare with 24-view carving, whose computational cost is much higher. We compare with E3D [2], the only event-based method that attempts to achieve multi-view 3D reconstruction. For fair comparison, we directly feed in the ground truth poses to E3D. E3D uses multi-view silhouette optimization over the objects, similar to PMO [15]. E3D directly uses the photometric optimization module in PMO [15] on silhouettes and removes the mesh prior from AtlasNet [10]. In our evaluation, we feed ground truth poses to E3D for fair comparison. In our experiments, we find silhouette-based optimization methods sensitive to the position and size of the mesh. To study the various components of EvAC3D, we report the performance of the ACE classification network and overall object reconstruction. For ACE classification, we provide the AUC curve of the classifier in Fig. 3.3 (b). The overall classification accuracy is 0.9563 (threshold=0.5). In Table 3, we show the mean performance and the mean number of operations. We define number of operations as the number of rays that we shoot out of the camera. EvAC3D uses significantly fewer operations than both 12 and 24 views. We notice that for both mask prediction networks, the quality of reconstruction degrades quickly when the number of views decreases. In practice, the sensor frame rate is not the only limiting factor - the computational power required to carve based on masks is also significantly higher. The average number of carving operations, mean Chamfer distance, and mean normal consistency are summarized in Table 3. This means the reconstruction quality of a frame-based algorithm largely depends on the motion speed, assuming the camera sensor has a fixed frame rate. We overcome this limitation of motion speed by directly operating on a continuous stream of events. We directly compare the qualitative results of the discussed methods in Figure 3.

4.4 Real Objects with Handheld Camera Trajectory

In the previous section, we present the experimental results for circular trajectories. However, camera trajectories can have more degrees of freedoms in real life. In this section, we put EvAC3D under test of more general handheld motions. The additional complexity of tasks comes not only from significant background events, but also the noisy camera pose estimation from handheld camera motion. We show a reconstructed hippo in Figure 8. Our reconstruction on this handheld sequence shows success in the main body of the hippo with an average reconstruction error of 1.5mm. The legs do not appear fully formed likely due to the small errors in the calibration and pose, both of which rely upon the reconstructed image to detect the AprilTags.

5 Conclusions

In this work, we present a novel method for continuous 3D reconstruction using event cameras. At the core of the method is the representation of occluding contours by Apparent Contour Events (ACE), a novel event quantity that can be used to continuously carve out high-fidelity meshes. EvAC3D is able to update the occupancy grid of the object on an event-to-event basis, which achieves better performance than mask-based visual hull approaches while using significantly fewer carving operations. We evaluate the performance of the method on both real and synthetic data. In addition, we contribute MOEC-3D, the first high-quality event-based 3D object dataset. With these contributions, we believe EvAC3D can provide important insights into how we can understand the 3D world through events.

Fig. 8. Results from a handheld trajectory. Left to right: raw events input, raw events overlaid with our reconstruction, image reconstruction using E2Vid [21], image reconstruction overlaid with our reconstruction, and the subsampled 3D camera trajectory with the computed mesh.

Acknowledgement We thank the support from the following grants: NSF TRIPODS 1934960, NSF CPS 2038873, ARL DCIST CRA W911NF-17-2-0181, ARO MURI W911NF-20-1-0080, ONR N00014-17-1-2093, DARPA-SRC C-BRIC, and IARPA ME4AI. We also thank William Sturgeon from the Fisher Fine Arts Materials Library for providing the Artec Spider scanner and assistance.

References

1. Barrow, H.G., Tenenbaum, J.M.: Interpreting line drawings as three-dimensional surfaces. Artificial intelligence **17**(1-3), 75–116 (1981) 4
2. Baudron, A., Wang, Z.W., Cossairt, O., Katsaggelos, A.K.: E3d: Event-based 3d shape reconstruction. arXiv preprint arXiv:2012.05214 (2020) 3, 12, 13
3. Baumgart, B.G.: Geometric modeling for computer vision. Stanford University (1974) 4
4. Carneiro, J., Ieng, S.H., Posch, C., Benosman, R.: Event-based 3d reconstruction from neuromorphic retinas. Neural Networks **45**, 27–38 (2013) 3
5. Chaney, K., Zhu, A.Z., Daniilidis, K.: Learning event-based height from plane and parallax. In: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 3690–3696 (2019). https://doi.org/10.1109/IROS40897.2019.8968223 3
6. Choy, C.B., Xu, D., Gwak, J., Chen, K., Savarese, S.: 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In: Proceedings of the European Conference on Computer Vision (ECCV) (2016) 13
7. Cipolla, R., Blake, A.: Surface shape from the deformation of apparent contours. International journal of computer vision **9**(2), 83–112 (1992) 4
8. Gehrig, D., Gehrig, M., Hidalgo-Carrió, J., Scaramuzza, D.: Video to events: Recycling video datasets for event cameras. In: IEEE Conf. Comput. Vis. Pattern Recog. (CVPR) (June 2020) 12
9. Giblin, P.: Reconstruction of surfaces from profiles. In: Proc. 1st International Conference on Computer Vision, London, 1987 (1987) 5, 6
10. Groueix, T., Fisher, M., Kim, V.G., Russell, B.C., Aubry, M.: A papier-mâché approach to learning 3d surface generation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 216–224 (2018) 7, 13
11. Ha, D., Dai, A., Le, Q.V.: Hypernetworks. arXiv preprint arXiv:1609.09106 (2016) 8
12. Hornung, A., Wurm, K.M., Bennewitz, M., Stachniss, C., Burgard, W.: OctoMap: An efficient probabilistic 3D mapping framework based on octrees. Autonomous Robots (2013). https://doi.org/10.1007/s10514-012-9321-0, `https://octomap.github.io`, software available at `https://octomap.github.io` 4
13. Kim, H., Leutenegger, S., Davison, A.J.: Real-time 3d reconstruction and 6-dof tracking with an event camera. In: European Conference on Computer Vision. pp. 349–364. Springer (2016) 3
14. Laurentini, A.: The visual hull: A new tool for contour-based image understanding. In: Proc. 7th Scandinavian Conf. Image Analysis. vol. 993, p. 1002 (1991) 4, 8
15. Lin, C.H., Wang, O., Russell, B.C., Shechtman, E., Kim, V.G., Fisher, M., Lucey, S.: Photometric mesh optimization for video-aligned 3d object reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 969–978 (2019) 13
16. Marr, D.: Analysis of occluding contour. Proceedings of the Royal Society of London. Series B. Biological Sciences **197**(1129), 441–475 (1977) 4
17. Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., Geiger, A.: Occupancy networks: Learning 3d reconstruction in function space. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4460–4470 (2019) 7, 8
18. Mitchell, E., Engin, S., Isler, V., Lee, D.D.: Higher-order function networks for learning composable 3d object representations. arXiv preprint arXiv:1907.10388 (2019) 8

19. Park, J.J., Florence, P., Straub, J., Newcombe, R., Lovegrove, S.: Deepsdf: Learning continuous signed distance functions for shape representation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 165–174 (2019) 7
20. Rebecq, H., Gallego, G., Mueggler, E., Scaramuzza, D.: Emvs: Event-based multi-view stereo—3d reconstruction with an event camera in real-time. International Journal of Computer Vision **126**(12), 1394–1414 (2018) 4
21. Rebecq, H., Ranftl, R., Koltun, V., Scaramuzza, D.: High speed and high dynamic range video with an event camera. IEEE transactions on pattern analysis and machine intelligence **43**(6), 1964–1980 (2019) 14
22. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. pp. 234–241. Springer (2015) 12
23. Szeliski, R.: Rapid octree construction from image sequences. CVGIP: Image understanding **58**(1), 23–32 (1993) 4
24. Wang, Z., Isler, V., Lee, D.D.: Surface hof: Surface reconstruction from a single image using higher order function networks. In: 2020 IEEE International Conference on Image Processing (ICIP). pp. 2666–2670. IEEE (2020) 8
25. Wang, Z., Mitchell, E.A., Isler, V., Lee, D.D.: Geodesic-hof: 3d reconstruction without cutting corners. arXiv preprint arXiv:2006.07981 (2020) 8
26. Wong, K.Y., Cipolla, R.: Structure and motion from silhouettes. In: Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001. vol. 2, pp. 217–222. IEEE (2001) 4
27. Zhou, Q.Y., Park, J., Koltun, V.: Open3D: A modern library for 3D data processing. arXiv:1801.09847 (2018) 12
28. Zhou, Y., Gallego, G., Rebecq, H., Kneip, L., Li, H., Scaramuzza, D.: Semi-dense 3d reconstruction with a stereo event camera. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 235–251 (2018) 3
29. Zhu, A.Z., Chen, Y., Daniilidis, K.: Realtime time synchronized event-based stereo. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 433–447 (2018) 3
30. Zhu, A.Z., Yuan, L., Chaney, K., Daniilidis, K.: Unsupervised event-based learning of optical flow, depth, and egomotion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 989–997 (2019) 7