Supplementary Material for “Time-varying Extremum Graphs”

Somenath Das[1], Raghavendra Sridharamurthy[1], and Vijay Natarajan[1,2]

[1]Department of Computer Science and Automation, Indian Institute of Science, Bangalore, India
[2]Zuse Institute Berlin, Germany

Abstract
This document presents additional material supporting the paper “Time-varying Extremum Graphs”. It provides pseudo-code for the algorithm and subroutines used for computing the TVEG, and an explanation of the method used to reduce visual clutter in the TVEG tracks computed for the viscous finger data. Next, it presents an additional comparison of TVEG tracks computed using varying weights assigned to the correspondence score components. It also presents detailed runtimes for TVEG computation. Finally, the major aspects of the visual analysis pipeline used in the case studies are explained.

1. Algorithms and psuedocode

We present the pseudocode for all the algorithms related to the computation of TVEG. TEMPORALARCS (Algorithm 1) computes all the temporal arcs. UPDATEMERGESPLITEDGES (Algorithm 2) updates the set of merge and split event sets after removing the highest score edge that participates in both a merge and split event. COMPUTESCORES (Algorithm 3) computes and returns the two best correspondences for each maximum in a time step and the associated scores. FILTERSCORES (Algorithm 4) refines these correspondences based on a threshold τ. DETECTMERGE, DETECTSPLIT, DETECTDEL, DETECTGEN (Algorithms 5,6,7,8) detect topological events merge, split, deletion, and creation respectively. EXGRAPH3D (Algorithm 9) computes the extremum graph of the time-varying scalar field at a given time step and simplifies the graph using a threshold θ. For the case studies that involve 3D data, we use EXGRAPH3D to compute extremum graphs, which in turn calls MS3D [SN12, DFRS14, BGL*18] to compute the MS complex. The extremum graph could also be directly computed from the scalar field [CLB11]. Table 2 contains the list of attributes of a critical point in an extremum graph.

2. Selection of TVEG tracks via cropping

TVEG tracks for the viscous fingers dataset includes a clusters of tracks near the domain boundary. These tracks are shorter in length and clutter the visualization as shown in Figure 1(a). We observed that the major fingers are formed along the central part of the domain and not near the boundary. The TVEG tracks consisting of temporal arcs that lie near the central part of the domain, shown in red, may be highlighted

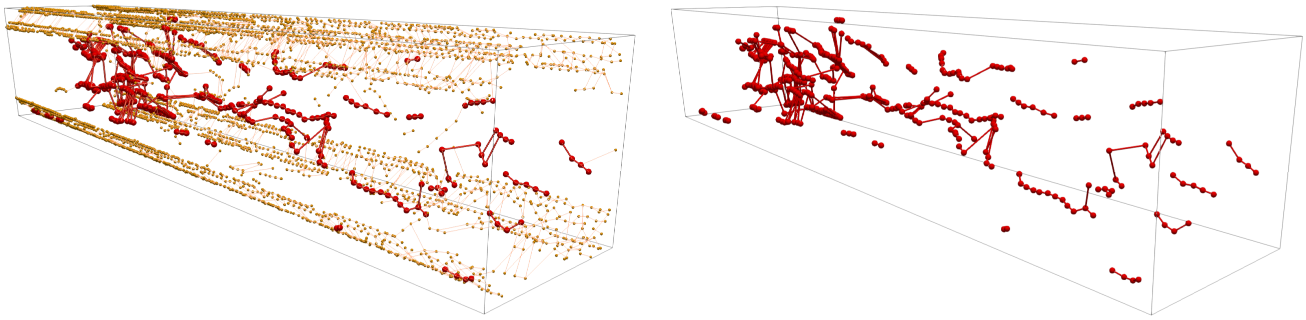
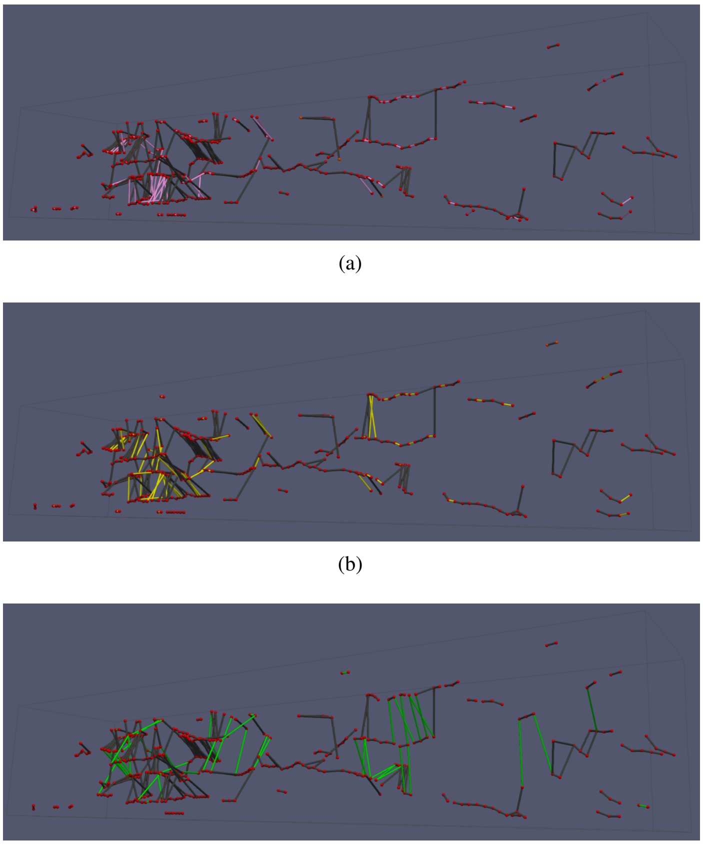

(a) All TVEG tracks
(b) Selected tracks after cropping

Figure 1: *(a) Computation of all TVEG tracks from the viscous finger dataset . Orange tracks are located near the domain boundary. (b) Cropping selects the red tracks, which represent the viscous finger formation.*



(a)

(b)

(c)

Figure 2: *Comparison of* TVEG *tracks computed using different weight combinations assigned to the score components. (G,L)= (a) (0.25,0.75), (b) (0.5,0.5), and (c) (0.75,0.25). The gray arcs are reported for all weight combinations. Additional arcs for a specific weight combination are highlighted in pink, yellow, and green, respectively.*

by removing the arcs near the boundary. Such a collection of tracks is shown in Figure 1(b). In the paper, we employ cropping to present the TVEG tracks that play an important role in explaining the data dynamics. Cropping essentially discards temporal arcs whose endpoint maxima lie within a certain distance threshold from the domain boundary.

3. Qualitative analysis of score components

In this section, we present the results of an additional study that helps understand the contributions of global and local components. This study supplements the parameter study described in the paper. In Figure 2, the TVEG tracks computed for three different weight assignments to the score components are presented. The TVEG tracks presented in Figure 2(a) - 2(c) correspond to a gradual increase in weights assigned to the global component persistence in comparison to the local components. While the gray TVEG tracks are found in common across all three settings, the exclusive contributions are highlighted in pink, yellow, and green, respectively. We observe that increasing the weight of the persistence component results in an increase in the number of abrupt jumps within TVEG tracks. Note that the weight assignment $(G,L) = (0.25,0.75)$ indicates equal assignment of weights to the four score components. This experiment further substantiates our initial choice of assigning equal weights to all score components for computing TVEG tracks.

4. Runtime analysis for TVEG computation

The datasets in the case studies contain different number of critical points, which is an indicator of the complexity of dynamic behavior in the data. The time for computing the TVEG naturally depends on the number of critical points and the size of the extremum graph. Table 1 summarizes the time taken by TemporalArcs for computing the TVEG. It lists the minimum / average / maximum per time step and the cumulative running time for TVEG computation of the three datasets. The number of nodes in the extremum graph (total number of maxima and 2-saddles) are also reported.

Table 1: *Runtimes for* TVEG *computation. The min, max, and average entries correspond to running times for processing individual time steps. Total refers to running time for processing all time steps.*

Dataset		**# Extremum graph nodes**	**Running time (ms)**
Gauss8	*Min*	14	0.21
	Avg	20	0.92
	Max	32	3.31
	Total	1010	53.69
Viscous fingers	*Min*	110	7.06
	Avg	123	24.62
	Max	133	56.17
	Total	14798	2963.96
Vortex street	*Min*	10	0.30
	Avg	82	18.67
	Max	294	265.85
	Total	41883	9595.16

We observe that the *Gauss8* data requires significantly less time since it is a small dataset. While the viscous fingers dataset is smaller than the vortex street dataset, it represents sufficient complexity in terms of temporal dynamics so that the average running time is comparable. The temporal dynamics of viscous fingers may be inferred from the significant difference between the minimum and maximum running time for processing one time step. Finally, the vortex street data is most complex, in terms of temporal dynamics, amongst the three datasets. The difference between the minimum and maximum running time is most significant for this dataset. Though the average running times are comparable, the vortex street data is significantly larger in size and contains more complex temporal dynamics. There is a long temporal phase where the size of the extremum graphs are small, followed by a transition to time steps where the extremum graphs are significantly large. Hence, the cumulative running times is larger for the vortex street dataset.

5. Visual Analysis Pipeline

We illustrate the basic computational, user interaction, and exploration stages involved in a typical visual analysis workflow using Figure 3. Given an input time-varying scalar field, TVEG computation begins with computation of the MS complex. The computed MS complexes at each time step are processed to extract the extremum graph. Blocks **B** and **C** in Figure 3 show the extremum graphs. Maxima are shown in red and 2-saddles green. TVEG tracks are computed as a sequence of temporal arcs, which denote correspondences between maxima from consecutive time steps. Block **D** shows portion of TVEG tracks within three consecutive time steps. The time direction is chosen as the spatial direction of growth of the tracks (z-direction in this dataset). The domain is scaled down along the z-axis to distinguish between spatial arcs of the extremum graph and temporal arcs in the TVEG.

Figure 3 is also used to demonstrate how TVEG can be applied to support simultaneous visualization of data dynamics at both the extremum graph and global data level. This is possible due to the rich geometric context, in the form of the extremum graph, that is present in the TVEG and the tracks within. Once computed, a specific portion of TVEG tracks can be chosen to explore the temporal dynamics within the data. For instance, in Block **E**, we select a section of TVEG tracks highlighted in brown, where the maxima contribute significantly to finger formation dynamics. This selection includes a split followed by a merge within three consecutive time steps. This choice lets us focus on the three corresponding neighborhoods shown in Block **F** within the extremum graphs where a maximum (in blue) at time step 68 is split into two maxima (shown in violet and orange) at time 69. The highlighted maxima at time 69 merge again into a maximum (pink) at time 70. From the extremum graph, we can extract the descending manifolds corresponding to these maxima participating in these topological events to visualize the resulting effect on the global data dynamics. In Block **F**, we have highlighted the descending manifolds with the same color as the corresponding maximum. Thus a TVEG can support visualization of temporal data dynamics influenced by temporal correspondence between maxima from a chosen TVEG track.

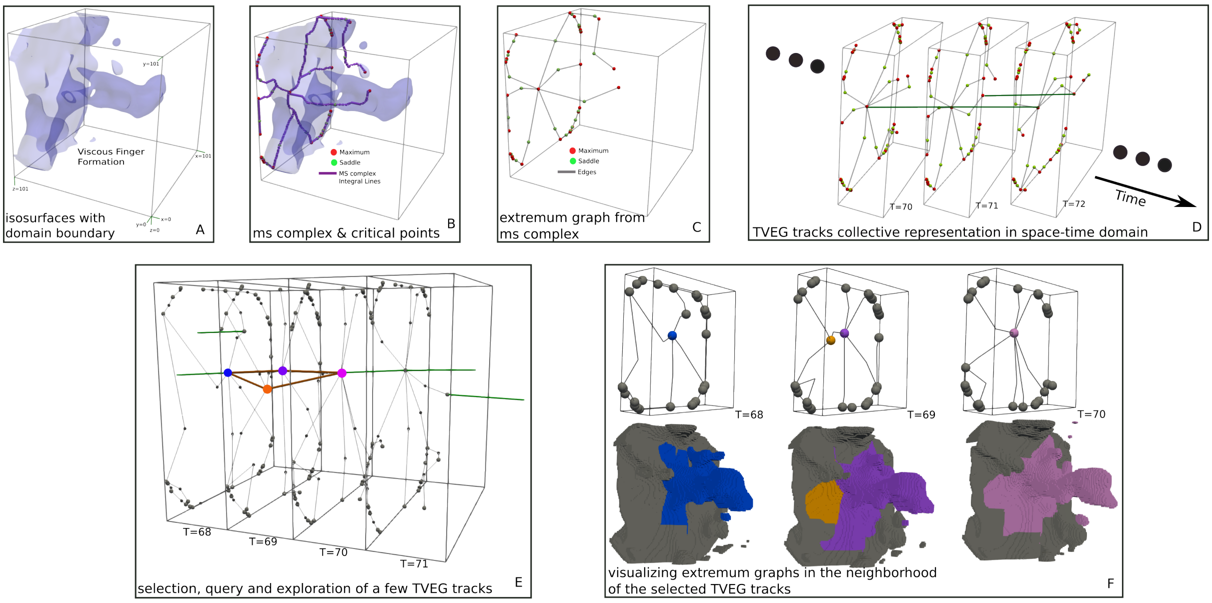

Viscous Finger Formation
y=101
x=101
z=101
x=0
y=0
z=0
isosurfaces with domain boundary
A
Maximum
Saddle
MS complex Integral Lines
ms complex & critical points
B
Maximum
Saddle
Edges
extremum graph from ms complex
C
T=70
T=71
T=72
Time
TVEG tracks collective representation in space-time domain
D
T=68
T=69
T=70
T=71
selection, query and exploration of a few TVEG tracks
E
T=68
T=69
T=70
visualizing extremum graphs in the neighborhood of the selected TVEG tracks
F

Figure 3: *Workflow of a typical visual analysis task based on the* TVEG *for the viscous fingers dataset. Individual steps include computation of the MS complex from the input scalar field (**A**) and subsequent extraction of extremum graphs (**B** and **C**) within each time step. Block **B** shows the integral lines representing the skeletal structure of the MS complex.* TVEG *tracks shown in deep green (**D**) are computed as a collection of arcs between a pair of extremum graphs from consecutive time steps. To avoid ambiguity between a spatial and a temporal arc, the spatial domain is scaled down along the time direction as shown in block **D**. A typical visualization task involves the selection and display of a subset of time steps (**E**) followed by their individual inspection (**F**). Block **E** shows time steps that contain* TVEG *tracks whose constituent maxima contribute most to the viscous finger formation. The* TVEG *track of interest and the maxima in the track are highlighted (**E**). The* TVEG *supports visualization of the neighborhood (arcs of extremum graph) of individual nodes of the chosen track. It also supports visualization of the corresponding dynamic changes (**F**) in the global data contributed by the maxima. In block **F**, a maximum and its descending manifold are highlighted using a common color.*

Table 2: *List of different fields corresponding to the critical points returned from the* EXGRAPH3D *subroutine (Algorithm 9)*

Fields	Description
id	A unique id assigned to the critical point
index	Index
$\bar{x}$	3D coordinates
η	Neighborhood contribution
pers	Topological persistence
ascmfold	Ascending manifold
dscmfold	Descending manifold
geom	Ascending / Descending manifold geometry
t	The time stamp of the scalar field

Algorithm 1: TEMPORALARCS

```
Input  : A set of extremum graphs $[\mathcal{G}^p, \ldots, \mathcal{G}^r]$
Output: Temporal arc set $A^{t*}$
        Topological event sets $\mathcal{E}^{m*}, \mathcal{E}^{s*}, \mathcal{E}^{d*}$, and $\mathcal{E}^{g*}$
Initialization: $A^{t*} \leftarrow \varnothing$; $A^t \leftarrow \varnothing$; $M^0 \leftarrow \varnothing$; $M^1 \leftarrow \varnothing$;
/* Initialize $M^0$ as maxima set of $\mathcal{G}^p$ */
$M^0 \leftarrow M^p$
/* Initialize all topological event sets */
$\{\mathcal{E}^{m*}, \mathcal{E}^{s*}, \mathcal{E}^{d*}, \mathcal{E}^{g*}\} \leftarrow \varnothing$
for $i \leftarrow p+1$ to $r$ do
    /* Initialize $M^1$ as maxima set of $\mathcal{G}^i$ */
    $M^1 \leftarrow M^i$
    $\mathcal{S} \leftarrow$ COMPUTESCORES($M^0, M^1$)
    $\mathcal{S} \leftarrow$ FILTERSCORES($\mathcal{S}$)
    /* Compute the temporal arc set $A^t$ */
    foreach $(m^0, m^1, s) \in \mathcal{S}$ do
        $A^t \leftarrow A^t \cup (m^0, m^1)$
    end
    /* Detect topological events */
    $\mathcal{E}^m \leftarrow$ DETECTMERGE($\mathcal{S}, i$)
    $\mathcal{E}^s \leftarrow$ DETECTSPLIT($\mathcal{S}, i$)
    $\mathcal{E}^d \leftarrow$ DETECTDEL($\mathcal{S}, M^0, i$)
    $\mathcal{E}^g \leftarrow$ DETECTGEN($\mathcal{S}, M^1, i$)
    /* Remove z-shape configurations. */
    $\mathcal{W} \leftarrow \mathcal{E}^m \bigcap \mathcal{E}^s$
    repeat
        $w \leftarrow$ MAXSCOREEDGE($\mathcal{W}$)
        $A^t \leftarrow A^t \setminus w$
        $\mathcal{E}^m, \mathcal{E}^s \leftarrow$ UPDATEMERGESPLITEDGES($\mathcal{E}^m, \mathcal{E}^s, w$)
        $\mathcal{W} \leftarrow \mathcal{E}^m \bigcap \mathcal{E}^s$
    until $\mathcal{W} = \emptyset$
    /* Populate temporal arc set */
    $A^{t*} \leftarrow A^{t*} \cup A^t$
    /* Update topological event sets */
    $\mathcal{E}^{m*} \leftarrow \mathcal{E}^{m*} \cup \mathcal{E}^m$
    $\mathcal{E}^{s*} \leftarrow \mathcal{E}^{s*} \cup \mathcal{E}^s$
    $\mathcal{E}^{d*} \leftarrow \mathcal{E}^{d*} \cup \mathcal{E}^d$
    $\mathcal{E}^{g*} \leftarrow \mathcal{E}^{g*} \cup \mathcal{E}^g$
    /* Re-initialize for next iteration */
    $A^t \leftarrow \varnothing, M^0 \leftarrow M^1$
end
```

Algorithm 2: UPDATEMERGESPLITEDGES Update the set of edges participating in merge/split events

```
Input  : Edge sets $\mathcal{E}^m$, $\mathcal{E}^s$, maximum score edge $w$
Output: Modified Edge sets
/* Remove edge adjacent to $w$ in $\mathcal{E}^s$ and $\mathcal{E}^m$ */
$u \leftarrow$ edge that participates in split with $w$
$\mathcal{E}^s \leftarrow \mathcal{E}^s \setminus \{w, u\}$
$U \leftarrow$ set of edges that participate in merge with $w$
if $U == \{u\}$ then
    $\mathcal{E}^m \leftarrow \mathcal{E}^m \setminus u$
$\mathcal{E}^m \leftarrow \mathcal{E}^m \setminus w$
return $\mathcal{E}^m$, $\mathcal{E}^s$
```

Algorithm 3: COMPUTESCORES Compute all correspondences and scores for a given set of maxima

```
Input  : Two maxima sets $M^0, M^1$
Output: A set of optimal scores $\mathcal{S} = \{(u,v,s)\}$ s.t.$(u,v) \in M^0 \times M^1$ and $s \in \mathbb{R}$
Initialization: $\mathcal{S} \leftarrow \varnothing$
foreach $m^0 \in M^0$ do
    $Q \leftarrow \varnothing$
    foreach $m^1 \in M^1$ do
        /* Compute score for $(m^0, m^1)$ given weights $G, L_1, L_2, L_3$ */
        $s \leftarrow G|m^0.\text{pers} - m^1.\text{pers}| + L_1|\mathcal{F}(m^0.\bar{x}) - \mathcal{F}(m^1.\bar{x})| + L_2|m^0.\bar{x} - m^1.\bar{x}|_2 + L_3|m^0.\eta - m^1.\eta|$
        $Q \leftarrow Q \cup s$
    end
    /* Insert two lowest scores to $\mathcal{S}$ */
    $s_1 \leftarrow min(Q)$, $\mathcal{S} \leftarrow \mathcal{S} \cup (m^0, m^1, s_1)$
    $Q \leftarrow Q \setminus s_1$
    $s_2 \leftarrow min(Q)$, $\mathcal{S} \leftarrow \mathcal{S} \cup (m^0, m^1, s_2)$
end
return $\mathcal{S}$
```

Algorithm 4: FILTERSCORES Filter the input set of scores based on a threshold

```
Input  : A list of scores $\mathcal{S}$ from Algorithm 3
Output: A filtered version of $\mathcal{S}$
Initialization: $\mathcal{Y} \leftarrow \varnothing$
foreach $(m^0, m^1, s) \in \mathcal{S}$ do
    $\mathcal{Y} \leftarrow \mathcal{Y} \cup s$
end
/* Compute the mean and standard deviation of all scores */
$\mu \leftarrow \text{MEAN}(\mathcal{Y})$, $\sigma \leftarrow \text{STD}(\mathcal{Y})$
/* Refine $\mathcal{S}$ using the threshold $\tau$ */
$\tau \leftarrow \mu + \sigma$
foreach $(m^0, m^1, s) \in \mathcal{S}$ do
    if $s \geq \tau$ then
        $\mathcal{S} \leftarrow \mathcal{S} \setminus (m^0, m^1, s)$
end
return $\mathcal{S}$
```

References

[BGL*18] BHATIA H., GYULASSY A. G., LORDI V., PASK J. E., PASCUCCI V., BREMER P.-T.: TopoMS: Comprehensive topological exploration for molecular and condensed-matter systems. *Journal of Computational Chemistry 39*, 16 (2018), 936–952. doi:10.1002/jcc.25181. 1

[CLB11] CORREA C., LINDSTROM P., BREMER P.-T.: Topological spines: A structure-preserving visual representation of scalar fields. *IEEE Transactions on Visualization and Computer Graphics 17*, 12 (2011), 1842–1851. doi:10.1109/TVCG.2011.244. 1

[DFRS14] DELGADO-FRIEDRICHS O., ROBINS V., SHEPPARD A.: Skeletonization and partitioning of digital images using discrete Morse theory. *IEEE Transactions on Pattern Analysis and Machine Intelligence 37*, 3 (2014), 654–666. doi:10.1109/TPAMI.2014.2346172. 1

[SN12] SHIVASHANKAR N., NATARAJAN V.: Parallel computation of 3D Morse-Smale complexes. *Computer Graphics Forum 31*, 3 (2012), 965–974. doi:10.1111/j.1467-8659.2012.03089.x. 1

Algorithm 5: DETECTMERGE Detect merge events between two consecutive time steps

```
Input  : A list of scores $\mathcal{S}$ from Algorithm 3;
         Time step $t$
Output: Set of merges $\mathcal{E}^m$ in $\mathcal{S}$ between time $t$ and $t+1$
Initialization: $\mathcal{E}^m \leftarrow \varnothing$
foreach $(m^0, m^1, s) \in \mathcal{S}$ do
    $e \leftarrow m^1$
    /* Count correspondences mapped to $e$                    */
    $c \leftarrow 0$; $\mathcal{K} \leftarrow \varnothing$
    foreach $(m^0, m^1, s) \in \mathcal{S}$ do
        if $m^1 == e$ then
            $\mathcal{K} \leftarrow \mathcal{K} \cup (m^0, t)$; $c \leftarrow c+1$
    end
    /* Merge detected. Update $\mathcal{E}^m$                    */
    if $c > 1$ then
        $\mathcal{E}^m \leftarrow \mathcal{E}^m \cup (e, t+1) \cup \mathcal{K}$
    $c \leftarrow 0$; $\mathcal{K} \leftarrow \varnothing$
end
return $\mathcal{E}^m$
```

Algorithm 6: DETECTSPLIT Detect split events between two consecutive time steps

```
Input  : A list of scores $\mathcal{S}$ from Algorithm 3;
         Time step $t$
Output: Set of splits $\mathcal{E}^s$ in $\mathcal{S}$ between time $t$ and $t+1$
Initialization: $\mathcal{E}^s \leftarrow \varnothing$
foreach $(m^0, m^1, s) \in \mathcal{S}$ do
    $e \leftarrow m^0$
    /* Count correspondences mapped from $e$                    */
    $c \leftarrow 0$; $\mathcal{K} \leftarrow \varnothing$
    foreach $(m^0, m^1, s) \in \mathcal{S}$ do
        if $m^0 == e$ then
            $\mathcal{K} \leftarrow \mathcal{K} \cup (m^1, t+1)$; $c \leftarrow c+1$
    end
    /* Split detected. Update $\mathcal{E}^s$                    */
    if $c > 1$ then
        $\mathcal{E}^s \leftarrow \mathcal{E}^s \cup (e, t) \cup \mathcal{K}$
    $c \leftarrow 0$; $\mathcal{K} \leftarrow \varnothing$
end
return $\mathcal{E}^s$
```

Algorithm 7: DETECTDEL Detect deletion events between two consecutive time steps

```
Input  : A list of scores $\mathcal{S}$ from Algorithm 3;
         A list of maxima $M^t$ for time step $t$;
         Time step $t$
Output: Set $\mathcal{E}^d$ with all the deletion events detected from $\mathcal{S}$ between time $t$ and $t+1$
Initialization: $\mathcal{E}^d \leftarrow \varnothing$; $\mathcal{K} \leftarrow M^t$
/* Record maxima with no edges to $t+1$ */
foreach $(m^0, m^1, s) \in \mathcal{S}$ do
    $\mathcal{K} \leftarrow \mathcal{K} \setminus m^0$
end
/* Add time step information */
foreach $e \in \mathcal{K}$ do
    $\mathcal{E}^d \leftarrow \mathcal{E}^d \cup (e, t)$
end
return $\mathcal{E}^d$
```

Algorithm 8: DETECTGEN Detect generation events between two consecutive time steps

```
Input  : A list of scores $\mathcal{S}$ from Algorithm 3;
         A list of maxima $M^{t+1}$ for time step $t+1$;
         Time step $t$
Output: Set $\mathcal{E}^g$ with all the generation events detected from $\mathcal{S}$ between time $t$ and $t+1$
Initialization: $\mathcal{E}^g \leftarrow \varnothing$; $\mathcal{K} \leftarrow M^{t+1}$
/* Record maxima with no edges from $t$ */
foreach $(m^0, m^1, s) \in \mathcal{S}$ do
    $\mathcal{K} \leftarrow \mathcal{K} \setminus m^1$
end
/* Add time step information */
foreach $e \in \mathcal{K}$ do
    $\mathcal{E}^g \leftarrow \mathcal{E}^g \cup (e, t+1)$
end
return $\mathcal{E}^g$
```

Algorithm 9: EXGRAPH3D Compute extremum graph

```
Input  : A scalar field $\mathcal{F}(t)$ at time step $t$
         A persistence threshold $\theta$
Output: Extremum Graph $\mathcal{G}^t = (V^t, E^t)$
Initialization: $V^t \leftarrow \varnothing$;$E^t \leftarrow \varnothing$
Compute MS complex: $\mathcal{C} \leftarrow \mathrm{MS3D}(\mathcal{F}(t), \theta)$
/* Store neighborhood maxima and saddles */
foreach $c \in \mathcal{C}$ do
    if $c.index == 2$ then
        $M \leftarrow c.\mathrm{ascmfold}$
        foreach $m \in M$ do
            $m^t \leftarrow m$, $V^t \leftarrow V^t \cup m^t$
            $E^t \leftarrow E^t \cup (c.\mathrm{id}, m.\mathrm{id})$
        end
    $c^t \leftarrow c$, $V^t \leftarrow V^t \cup c^t$
end
$V^t \leftarrow \mathrm{set}(V^t)$
return $\mathcal{G}^t = (V^t, E^t)$
```