This article has been accepted for publication in IEEE Transactions on Visualization and Computer Graphics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2025.3630550

LoGCC: Local-to-Global Correlation Clustering for Scalar Field Ensembles

Nicolas F. Chaves-de-Plaza, Renata G. Raidou, Prerak Mody, Marius Staring, René van Egmond, Anna Vilanova and Klaus Hildebrandt

Abstract— Correlation clustering (CC) offers an effective approach to analyze scalar field ensembles by detecting correlated regions and consistent structures, enabling the extraction of meaningful patterns. However, existing CC methods are computationally expensive, making them impractical for both interactive analysis and large-scale scalar fields. We introduce the Local-to-Global Correlation Clustering (LoGCC) framework, which accelerates pivot-based CC by leveraging the spatial structure of scalar fields and the weak transitivity of correlation. LoGCC operates in two stages: a local step that uses the neighborhood graph of the scalar field's spatial domain to build highly correlated local clusters, and a global step that merges them into global clusters. We implement the LoGCC framework for two well-known pivot-based CC methods, Pivot and CN-Pivot, demonstrating its generality. Our evaluation using synthetic and real-world meteorological and medical image segmentation datasets shows that LoGCC achieves speedups—up to 15× for Pivot and 200× for CN-Pivot—and improved scalability to larger scalar fields, while maintaining cluster quality. These contributions broaden the applicability of correlation clustering in large-scale and interactive analysis settings.

Index Terms—Uncertainty visualization, scalar field ensemble analysis, correlation clustering, structural variability.

1 Introduction

Scalar field ensembles are collections of scalar fields representing the same variable or property across different instances, conditions, or scenarios, such as multiple simulations, measurements, or image segmentations, capturing variability and uncertainty in the data. Correlation clustering provides means for analyzing these ensembles by revealing coherent regions of correlated behavior across ensemble members, facilitating the identification of consistent structures, quantification of uncertainty, and extraction of meaningful patterns [1]–[4].

Pivot-based schemes are used for computing correlation clusters because of their conceptual simplicity and low memory requirements, as they avoid the need to store all pairwise correlations explicitly. Instead, they iteratively select pivot cells and form clusters by aggregating cells with sufficiently strong correlations [1], [5]. Despite these advantages, CC methods—including pivot-based approaches—suffer from poor scalability: their worst-case time complexity grows quadratically with the number of cells. As a result, existing techniques have been applied mainly to relatively small 2D scalar field ensembles, where a complete analysis involving only a few thousand cells can still take hours [1].

In this paper, we present the Local-to-Global Correlation Clustering (LoGCC) framework, which accelerates pivot-based CC methods by leveraging the spatial structure of the scalar field domain (i.e., its neighborhood graph) and the weak transitivity of correlation. LoGCC divides CC into two stages: a local step and a global step. In the local step, the domain's neighborhood graph is used to construct highly correlated local clusters from randomly sampled spatial cells. In the global step, these local clusters are merged into global clusters by exploiting weak transitivity relationships between their representative cells. This process is illustrated in Fig. 1, where (a) shows a scalar field ensemble and (b) and (c) depict the local and global steps of LoGCC, respectively.

A key contribution of the LoGCC framework is its ability to exploit weak transitivity relationships between correlations to reduce the number of cell pairs that must be compared when merging local clusters. Assessing all possible relationships between cells in a pair of local clusters would negate runtime gains. Instead, we focus on evaluating relationships between the pivots of the local clusters. By leveraging the weak transitivity properties of correlations, we derive lower bounds on the possible correlation values between clusters. Our results show that, for a range of local and global correlation thresholds, LoGCC provides stricter theoretical bounds than the Pivot method [5].

We demonstrate LoGCC's generality and flexibility by adapting two practically relevant pivot-based CC methods—Pivot [5] and correlation neighborhood CN-Pivot [1]—to its two-step structure. The resulting LoGCC-accelerated versions execute significantly faster while producing clustering results of comparable quality. The local step scales linearly with the scalar field resolution, and the global step inherits the time complexity of the original CC algorithm but operates on a substantially reduced number of primitives produced by the local step. These improvements enable CC analysis on larger datasets in less time and support the exploration of different correlation thresholds.

We evaluate the runtime improvements of the local-to-

- *N.F. Chaves-de-Plaza, R. van Egmond, and K. Hildebrandt are with TU Delft, Netherlands. E-mail: n.f.chavesdeplaza@tudelft.nl*
- *R.G. Raidou is with TU Wien, Austria.*
- *P. Mody and M. Staring are with the Department of Radiology, Leiden University Medical Center, Netherlands.*
- *M. Staring is with the Department of Radiation Oncology, Leiden University Medical Center, Netherlands.*
- *A. Vilanova is with TU Eindhoven, Netherlands.*
- *N.F. Chaves-de-Plaza and P. Mody are with HollandPTC, Netherlands.*

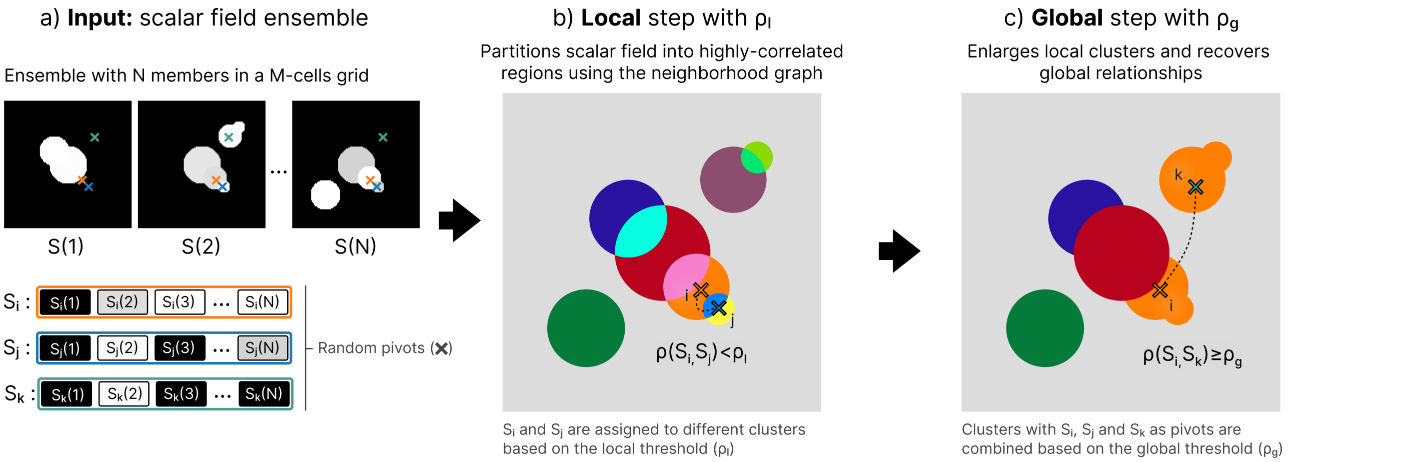

a) Input: scalar field ensemble
Ensemble with N members in a M-cells grid
S(1)
S(2)
...
S(N)
S_i :
$S_i(1)$ $S_i(2)$ $S_i(3)$ ... $S_i(N)$
S_j :
$S_j(1)$ $S_j(2)$ $S_j(3)$ ... $S_j(N)$
S_k :
$S_k(1)$ $S_k(2)$ $S_k(3)$ ... $S_k(N)$
Random pivots (×)
b) Local step with ρ_l
Partitions scalar field into highly-correlated regions using the neighborhood graph
i
j
$\rho(S_i, S_j) < \rho_l$
S_i and S_j are assigned to different clusters based on the local threshold (ρ_l)
c) Global step with ρ_g
Enlarges local clusters and recovers global relationships
k
i
$\rho(S_i, S_k) \geq \rho_g$
Clusters with S_i, S_j and S_k as pivots are combined based on the global threshold (ρ_g)

Fig. 1. Overview of the Local-to-Global Correlation Clustering (LoGCC) framework. The input is an ensemble of scalar fields over a discretized domain (a), where blobs present across the ensemble members define regions of spatially correlated cells. LoGCC first computes local clusters based on correlations between adjacent cells (b), and then merges them into global clusters (c). The decision to merge two local clusters is guided by the correlation of their pivots, a strategy justified by the weak transitivity of correlations. In the shown example, the global correlation threshold ρ_g is smaller than the local one ρ_l, which explains why pivots $i, j,$ and k are merged to form one cluster.

global variants of Pivot and CN-Pivot using a synthetic dataset that allows controlled variation of key parameters, such as the field size and the number of clusters. These experiments demonstrate speedups of up to 15× for Pivot and 200× for CN-Pivot. To validate LoGCC's practical utility, we further assess its performance on three real-world datasets from diverse application domains. The experiments confirm LoGCC's performance gains, enabling efficient processing of larger datasets in significantly less time.

In summary, the main contributions of the paper are:

- *The Local-to-Global Correlation Clustering (LoGCC) framework.* LoGCC substantially accelerates existing CC algorithms by using the spatial structure of the domain underlying the scalar fields to form local clusters and then merging them into global clusters.
- *Weak correlation transitivity for cluster merging.* LoGCC leverages the weak transitivity properties of correlation to reduce the number of comparisons between cells during cluster merging. This is crucial for the runtime efficiency and is based on lemmas providing lower bounds for threshold selection.
- *Adaptation of CC algorithms.* LoGCC accelerates the Pivot and CN-Pivot methods without compromising clustering quality, facilitating the efficient analysis of larger datasets and the exploration of correlation thresholds.

2 Related Work

Ensemble uncertainty visualization. Our work falls within the uncertainty visualization domain, focusing specifically on the visualization of ensemble data. Previous studies in this area can be categorized according to data type, dimensionality, and visualization strategies [6]. The predominant approach for analyzing scalar field ensemble uncertainty consists of examining the spread of features or variables at individual cells either directly on the scalar field [7] or, more commonly, on iso-contours derived from scalar fields [8] using parametric [9] and non-parametric models [10]–[14]. Our approach extends beyond cell-based analysis by considering the relationships between cells and operating directly on the scalar fields, as extracting contours can lead to structural information loss.

Correlation analysis and visualization. Several techniques address correlations between variables. Correlation maps permit analyzing pairwise relationships between large numbers of categorical and numerical variables [15]. An alternative strategy is to group similarly distributed variables and represent these groups with a centroid [16]. For scalar field ensembles, Multifield-Graphs [17] allow visual exploration of correlation fields derived from ensemble members. This approach remains location-based, comparing values at the same cell across ensemble members rather than comparing values at different cell locations, which is essential for detecting structural changes in scalar field ensembles due to dependencies between cells [18].

Correlation clustering. Correlation clustering (CC) provides a framework for analyzing structural changes in scalar field ensembles due to dependencies between cells. CC methods are related to superpixel approaches, which partition an input image's domain into perceptually similar regions [19]. Nevertheless, they differ from superpixels in their optimization objectives as well as in the type of data and representations they operate on. Existing CC methods vary in the problem formulation and their support of extensions such as extracting inverse correlations and performing hierarchical clustering. Some approaches represent cells in alternative spaces to enhance clustering quality. For example, correlation-based metrics can be used in nearest neighbor searches on a hyperspectral projection, producing clusters allowing exploration via a dendrogram [2]. Another method maps cells to a 3D space aligned with a color space, where segmentation reveals regions of local and global cell connections [4]. Although some work utilizes the locality of correlations in scalar fields [18], [20], they do not leverage this to accelerate the CC computation as we do.

A major challenge of existing CC methods is their high computational cost, as they require computing all pairwise correlations between scalar field cells, limiting their scalability to larger datasets (e.g., 3D ensembles). To improve efficiency, some methods use distributed and parallel processing [21], while others employ sampling techniques based on domain knowledge [22] or Bayesian optimal sampling [23]. However, these sampling-based methods are designed to accelerate specific correlation queries (e.g., finding maximum correlations) rather than to support correlation clustering and are typically confined to hierarchical grids, overlooking the anisotropic shape of correlation neighborhoods [1]. In this work, we propose a two-step domain partitioning that accounts for this anisotropy. An alternative approach to accelerating correlation evaluation is the use of correlation networks, which train a neural network to overfit to a specific dataset and support interactive queries [24]. Once trained, these networks can rapidly identify all cells correlated with a selected pivot element. Like correlation networks, our approach approximates pairwise correlations while avoiding the overhead of model training.

Pivot-based correlation clustering. CC is a longstanding problem in machine learning, with growing interest in the visualization community. The standard CC problem formulation involves grouping graph vertices to minimize the number of correlated pairs that are separated and uncorrelated pairs that are clustered together—an NP-hard problem [25]. The 3-approximation pivot algorithm [5] is widely studied for its simplicity, ease of implementation, and practical performance. Subsequent work improved the approximation guarantee to 2 using integer linear programming [26], [27], albeit at the cost of quadratic runtime. Parallelization has further improved speed [28], [29]. In this work, we demonstrate additional speedups by exploiting the spatial structure of the graph.

In pivot-based algorithms, scalar field ensembles are modeled as graphs where cells correspond to vertices and thresholded correlations define edges. The CN-Pivot algorithm produces clusters that reflect the anisotropy and strength of correlation neighborhoods [1], revealing both local and global relationships. Unlike other CC methods in visualization [2], [4], CN-Pivot does not require storing the full pairwise correlation matrix. It can also handle inverse correlation structures and hierarchical clustering. Furthermore, it can be extended to consider correlation cliques, which form more compact clusters by accounting for all pairwise relationships within a cluster [3]. Despite these advantages, CN-Pivot, like Pivot, faces quadratic runtime challenges. We demonstrate how LoGCC can significantly accelerate CN-Pivot, enhancing its practical utility.

3 Background

We consider an ensemble $S = \{S(1), S(2), ..., S(N)\}$ of N scalar fields on a domain in a Euclidean space. For discretization, the domain is decomposed into cells, *e.g.* by a regular grid, and each scalar field is represented by a real number per cell. We index the cells and describe each scalar field by a vector $S(n) \in \mathbb{R}^M$, which stacks the values the scalar field takes on all cells. Here, M denotes the number of cells. $S_i(n)$ denotes the i^{th} entry of $S(n)$ and $S_i \in \mathbb{R}^N$ the vector stacking the values $S_i(n)$ for all n. Fig. 1 (a) presents an example of the notation used for three cells. We describe the correlation of the ensemble at a pair of cells (i, j) by evaluating the correlation $\rho(S_i, S_j)$ of S_i and S_j. Different correlation functions ρ can be used. We use the Pearson correlation coefficient given by

$$\rho(S_i, S_j) = \frac{\sum_{n=1}^{N}(S_i(n) - \bar{S}_i)(S_j(n) - \bar{S}_j)}{\sqrt{\sum_{n=1}^{N}(S_i(n) - \bar{S}_i)^2 \sum_{n=1}^{N}(S_j(n) - \bar{S}_j)^2}}, \quad (1)$$

where $\bar{S}_i$ denotes the mean of the N entries of S_i.

Pivot-based CC algorithms operate on a complete undirected and unweighted graph $G = (V, E)$. Based on a given correlation threshold ρ_t, the edge set E is split into two disjoint sets: the correlated pairs of vertices, E^+, and the uncorrelated pairs of vertices, E^-. The goal is to cluster the vertices such that the sum of correlated pairs not in the same cluster and uncorrelated pairs within the same cluster is minimized [5], [25]. More formally, we represent the clustering by an array $I \in \{1, ..., K\}^M$, where K is the number of clusters and $I(i)$ indicates the cluster membership of vertex i. Then, the cost function

$$\begin{aligned} C(G, I) = &\sum_{(i,j)\in E^+} 1[I(i) \neq I(j)] + \\ &\sum_{(i,j)\in E^-} 1[I(i) = I(j)], \end{aligned} \quad (2)$$

where $1[\text{True}] = 1$ and $1[\text{False}] = 0$, is minimized.

To apply a pivot-based CC algorithm to an ensemble of scalar fields, the graph that has a vertex for each cell of the scalar fields' domain and the complete set of edges between all pairs of vertices is used. The subsets E^+ and E^- are given by $E^+ = \{(i, j) \in E : \rho(S_i, S_j) \geq \rho_t\}$ and $E^- = \{(i, j) \in E : \rho(S_i, S_j) < \rho_t\}$.

3.1 Pivot Algorithm

To construct a clustering, the Pivot algorithm [5] starts by randomly selecting an unassigned vertex as a pivot and forming a cluster around it by adding all other unassigned vertices with which it forms a correlated pair. This pivot assignment and cluster-building process is repeated until all vertices have been assigned (pseudocode is provided in the supplementary material). The algorithm's worst-case complexity is $\mathcal{O}(NM^2)$, assuming linear-time pairwise correlation evaluation, though runtime may be more favorable in practice depending on cluster sizes and the order of cluster construction. For example, if all M vertices belong to a single cluster, the algorithm runs in linear time, whereas performance deteriorates as the number of clusters increases, with the worst case occurring when each vertex forms its own cluster, requiring consideration of all vertex pairs and resulting in quadratic time complexity. Due to its randomized nature, clustering results may also vary across runs.

3.2 CN-Pivot Algorithm

The correlation neighborhood (CN)-Pivot algorithm [1] is an alternative to the Pivot algorithm designed for scalar field ensembles. Rather than directly minimizing the cost function

This article has been accepted for publication in IEEE Transactions on Visualization and Computer Graphics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2025.3630550

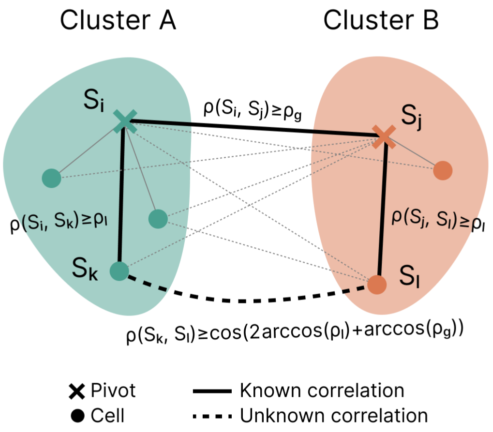

Cluster A
Cluster B
S_i
$\rho(S_i, S_j) \geq \rho_g$
S_j
$\rho(S_i, S_k) \geq \rho_l$
$\rho(S_j, S_l) \geq \rho_l$
S_k
S_l
$\rho(S_k, S_l) \geq \cos(2\arccos(\rho_l) + \arccos(\rho_g))$
× Pivot
● Cell
Known correlation
Unknown correlation

Fig. 2. Schematic of different types of relationships that arise when comparing two clusters. The within-cluster correlations between pivots and other cells (at least ρ_l) and between-cluster correlations between pivots (at least ρ_g) determine the values that cluster correlations between non-pivots (dashed line) can attain, as stated in Lemma 4.2.

in Eq. 2, it aims to produce clusters whose sizes reflect the strength of the correlation neighborhoods of their pivots. For a vertex i, the correlation neighborhood is defined as $\eta(i) = \{j \in V : (i,j) \in E^+\}$, i.e., all vertices with which i shares a positive relationship. Unlike Pivot, which selects pivots at random, CN-Pivot chooses the unassigned vertex with the largest correlation neighborhood; if all members of this neighborhood are unassigned, they form a new cluster, otherwise the vertex is skipped and the algorithm continues in decreasing order of neighborhood size. While CN-Pivot has the same worst-case time complexity of $\mathcal{O}(NM^2)$ as Pivot, its performance does not improve in practice, since determining the pivot order requires computing correlation neighborhoods for all vertices, which already incurs quadratic cost. Pseudocode for the algorithm is provided in the supplementary material.

4 Local-to-Global Correlation Clustering

In this section, we introduce the Local-to-Global Correlation Clustering (LoGCC) framework, which leverages the inherent structure of scalar field ensembles to accelerate correlation clustering (CC). We begin by discussing transitivity properties of correlation relevant to LoGCC's design.

4.1 Correlation Transitivity

Transitivity of correlation would mean that if the pairs (S_i, S_j) and (S_i, S_k) satisfy $\rho(S_i, S_j) \geq \rho$ and $\rho(S_i, S_k) \geq \rho$ for some threshold ρ, then also $\rho(S_j, S_k) \geq \rho$. This property is interesting for pivot algorithms; if S_i is a pivot of a cluster and S_j and S_k are members, then there would be a cost (see Eq. 2) if $\rho(S_j, S_k) < \rho$. Still, correlation is, in general, not transitive. Since pivot algorithms work well, however, there seems to be a possibly weaker form of correlation between S_j and S_k. The following lemma confirms this conjecture and establishes an explicit bound on the correlation of S_j and S_k. A proof can be found in the supplementary material.

Lemma 4.1. Consider a triplet $S_i, S_j, S_k \in \mathbb{R}^N$ and assume $\rho(S_i, S_j) \geq \rho$ and $\rho(S_i, S_k) \geq \rho$ for some $\rho \in [0,1]$. Then,

$$\rho(S_j, S_k) \geq \cos(2\arccos(\rho)). \qquad (3)$$

This lemma is interesting in the context of pivot algorithms as for a given correlation threshold ρ_t, one can choose a threshold $\rho_l > \cos\left(\frac{1}{2}\arccos(\rho_t)\right)$ to ensure that for all pairs (S_j, S_k) within a cluster, $\rho(S_j, S_k) \geq \rho_t$. As a result, no uncorrelated pairs are included in the same cluster. The second lemma concerns correlations between clusters.

Lemma 4.2. Consider a quadruple $S_i, S_j, S_k, S_l \in \mathbb{R}^N$ and assume $\rho(S_i, S_j) \geq \rho_g$, $\rho(S_i, S_k) \geq \rho_l$ and $\rho(S_j, S_l) \geq \rho_l$ for some $\rho_l, \rho_g \in [0.5, 1]$. Then,

$$\rho(S_l, S_k) \geq \cos(2\arccos(\rho_l) + \arccos(\rho_g)). \qquad (4)$$

Fig. 2 illustrates the relationships that arise between scalar field cells of two clusters. Local relations involve within-cluster cells (e.g., S_i, S_k) and global relations involve between-cluster cells (e.g., S_k, S_l). If S_k and S_l are two arbitrary elements from two clusters with pivots S_i and S_j, and we merge the clusters based on the correlation between their pivots, then lemma 4.2 provides a lower bound for correlations between other non-pivot cells like $\rho(S_l, S_k)$.

The lemmas introduced in this section constitute the basis of the LoGCC framework.

4.2 The Local Step

The aim of the first stage of LoGCC is to form local clusters, i.e. clusters that are each a connected subset of the domain. For this, LoGCC operates on a different graph than the complete graph used by pivot algorithms. While the graph also has a vertex for each cell of the domain of the scalar field, it only contains the edges that connect cells that are neighbors in the scalar field's domain. This neighbor graph is significantly reduced compared to the complete graph, as the number of edges is linear in the number of vertices while it is quadratic for the complete graph.

Algorithm 1 outlines the local step. It receives as input the scalar field ensemble and a local correlation threshold ρ_l. We denote by $\mathcal{N}(i)$ the operator that returns the indices of the cells adjacent to cell i. The algorithm incrementally partitions the spatial domain by forming clusters around randomly selected pivots. Starting from a random pivot cell, neighboring correlated cells are iteratively added to form a cluster. The traversal stops when the cluster is surrounded only by uncorrelated cells or cells already assigned to other clusters. This step is repeated with a new random pivot not yet in a cluster, only considering cells not yet in a cluster. The traversal ends when every cell is assigned to a cluster.

4.3 The Global Step

The global step finalizes the clustering by processing the local clusters generated in the previous step. This involves constructing a new graph, the hypergraph, where each vertex represents a cluster generated in the local step, and edges connect all pairs of vertices. As with pivot algorithms (see Sec. 3), the edge set is divided into two subsets: E^+ for correlated cluster pairs and E^- for uncorrelated ones. A global threshold ρ_g determines the correlation between

Algorithm 1 Local LoGCC

```
Require: $S, \rho_l$        ▷ Ensemble, local correlation threshold
 1: $\Pi \leftarrow$ uniform random permutation of cells
 2: $\Psi = \emptyset; I = \emptyset$        ▷ Local pivots and clusters
 3: $\mathcal{M} \leftarrow \emptyset$        ▷ Keep track of assigned cells
 4: while $\Pi \neq \emptyset$ do
 5:     $i \leftarrow pop(\Pi, 1)$        ▷ Pop first cell in $\Pi$
 6:     if $i \notin \mathcal{M}$ then
 7:         $\Psi \leftarrow \Psi \cup i; I_i \leftarrow \{i\}; \mathcal{M} \leftarrow \mathcal{M} \cup i$
 8:         $\mathcal{Q} \leftarrow \mathcal{N}(i)$   ▷ Initialize queue with $i$'s neighbors
 9:         $\mathcal{V} \leftarrow \emptyset$        ▷ Keep track of visited cells
10:         while $\mathcal{Q} \neq \emptyset$ do
11:             $j \leftarrow pop(\mathcal{Q}, 1)$
12:             if $j \in \mathcal{V}$ then
13:                 Skip $j$, go back to line 10
14:             end if
15:             $\mathcal{V} \leftarrow \mathcal{V} \cup j$
16:             $are\_corr \leftarrow \rho(S_i, S_j) \geq \rho_l$   ▷ I.e., $(i, j) \in E^+$
17:             if $j \notin \mathcal{M} \wedge are\_corr$ then
18:                 $I_i \leftarrow I_i \cup j$
19:                 $\Pi \leftarrow \Pi - \{j\}; \mathcal{M} \leftarrow \mathcal{M} \cup j$
20:                 $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{k : k \in \mathcal{N}(j) \wedge k \notin \mathcal{M}\}$
21:             end if
22:         end while
23:     end if
24: end while
25: return $\Psi, I$
```

Algorithm 2 Global LoGCC (Pivot Algorithm)

```
Require: $S, \rho_g, \Psi^l, I^l$        ▷ Ensemble, global correlation
    threshold, local pivots and clusters
 1: $\Pi \leftarrow \Psi$        ▷ Visit clusters in same order as pivots
 2: $\Psi = \emptyset; I = \emptyset$        ▷ Global pivots and clusters
 3: while $\Pi \neq \emptyset$ do
 4:     $i \leftarrow pop(\Pi, 1)$        ▷ Pop first pivot in $\Pi$
 5:     $\Psi \leftarrow \Psi \cup i$
 6:     $I_i \leftarrow I^l_i$        ▷ Initialize with cells of $i$'s local cluster
 7:     for $j \in \Pi$ do
 8:         $are\_corr \leftarrow \rho_*(i, j) \geq \rho_g$   ▷ $\rho_*$ compares clusters
    associated with pivots $i, j$
 9:         if $are\_corr$ then
10:             $\Pi \leftarrow \Pi - \{j\}$
11:             $I_i \leftarrow I_i \cup I^l_j$        ▷ Add cells of $j$'s local cluster
12:         end if
13:     end for
14: end while
15: return $\Psi, I$
```

Algorithm 3 Global LoGCC (CN-Pivot Algorithm)

```
Require: $S, \rho_g, \Psi^l, I^l$        ▷ Ensemble, global correlation
    threshold, local pivots and clusters
  $\eta_*(i) = I^l_i \cup \{I^l_j : j \in \Psi^l \wedge \rho_*(i, j) \geq \rho_g\}$        ▷
  Correlation neighborhoods for all $i \in \Psi^l$ based on inter-
  cluster correlation measure $\rho_*$
  $\Pi \leftarrow$ permutation of $\Psi^l$ in decreasing order of correlation
  neighborhood cardinality
  $\Psi = \emptyset; I = \emptyset$        ▷ Global pivots and clusters
  while $\Pi \neq \emptyset$ do
      $i \leftarrow pop(\Pi, 1)$        ▷ Pop first vertex in $\Pi$
      $empty\_intersect \leftarrow \eta_*(i) \cap \eta_*(j) = \emptyset \forall j \in \Psi$
      if $empty\_intersect$ then
          $\Psi \leftarrow \Psi \cup i; I_i \leftarrow \eta_*(i)$        ▷ Build cluster
      end if
  end while
  return $\Psi, I$
```

clusters. A pair of clusters is considered correlated if the correlation between their pivots exceeds ρ_g and uncorrelated otherwise, as illustrated in Fig. 2. In Algs. 2 and 3, the function $\rho_*(i, j)$ is used to evaluate the correlation between clusters i and j based on their pivots. Using pivots avoids costly pairwise comparisons between all cells in the clusters. It also allows us to reuse the established relationships between the pivots and other cells in their respective clusters from the local step. In the next subsection, we discuss how this approach provides a lower bound on the correlations between any pair of cells within a cluster formed during the global step.

LoGCC can be used to accelerate various CC algorithms, with each variant resulting in a corresponding global step tailored to the original method. We will discuss global steps modeled after Pivot and CN-Pivot. Alg. 2 shows our adaptation of Pivot. The algorithmic structure remains unchanged, but operations for forming and comparing clusters are adapted to the hypergraph's vertices and edges. Adapting the CN-Pivot algorithm requires an additional preprocessing step for defining the visiting order based on the cardinalities of the correlation neighborhoods. The resulting global step is listed in Alg. 3.

4.4 Choice of Thresholds and Runtime Analysis

As a consequence of Lemmas 4.1 and 4.2, for a target correlation threshold ρ_t, one can choose ρ_l and ρ_g such that the resulting clusters after the global step contain no uncorrelated pairs (i.e., edges from E^-). In our experiments, however, this proves too strict, and we obtain better results for less strict values. We choose the threshold ρ_l to be larger than ρ_g to obtain local clusters with high internal correlations (Lemma 4.1). This choice is motivated by the fact that ρ_l appears with a factor of two in the bound derived in Lemma 4.2. Our experiments suggest that setting ρ_l close to one, e.g., $\rho_l = 0.99$, and choosing the global step threshold equal to the target, $\rho_g = \rho_t$, are suitable values. Figure 3 shows the lower bound for the correlation of any pair of elements that are in one cluster for unaccelerated and LoGCC-accelerated variants of the Pivot method across different ρ_t values. One can see that the suggested choice of parameter values yields a stricter bound for the LoGCC-accelerated variant when $\rho_t < 0.96$.

LoGCC's runtime is the sum of the local and global steps. In the local step in Alg. 1, each edge of the neighbor graph is visited at most twice, so its runtime scales linearly with the number of cells M. The global step's runtime is determined by the chosen algorithm applied to a hypergraph with M_h vertices and M_h^2 edges, where M_h is the number of clusters produced by the local step. The Pivot and CN-Pivot methods in Algs. 2 and 3 scale quadratically with the number of vertices, meaning LoGCC's global step also

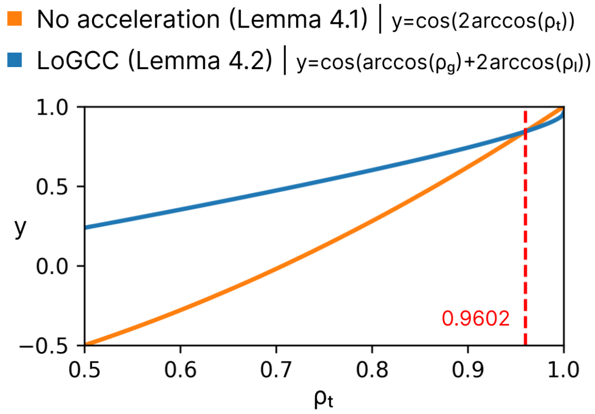


Fig. 3. Lower bounds for the correlation of any pair of cells in one cluster for the Pivot method without (Lemma 4.1) and with (Lemma 4.2) LoGCC acceleration. The y-axis denotes the bounds' values obtained by evaluating the lemma's equations with correlation thresholds $\rho_t \in [0.5, 1]$. We set $\rho_l = 0.99$ and $\rho_g = \rho_t$ for the LoGCC variant.

scales quadratically with M_h, which is a monotonically increasing function of the local correlation threshold ρ_l. A single run of LoGCC therefore has a worst-case time complexity of $\mathcal{O}(N(M + M_h^2))$, assuming a linear time cost for evaluating the correlation between N-dimensional cells. We experimentally confirmed that $M_h \ll M$ and $M + M_h^2 \ll M^2$, which results in a more favorable scaling behavior of LoGCC compared to unaccelerated algorithms.

Finally, when analyzing a range of thresholds, maintaining clustering stability is crucial for comparison. We found that reusing the pivot order from the local step significantly enhances stability.

5 Experiments and Results

In this section, we demonstrate how the LoGCC's framework can significantly accelerate existing pivot-based CC methods without compromising clustering quality.

5.1 Data

We use a synthetic dataset with adjustable parameters to demonstrate the speedups achieved by LoGCC, analyze its scaling behavior, and identify data characteristics that influence its performance. Since the outputs and performance of correlation clustering methods are highly dataset-dependent, we also evaluate LoGCC on three real-world datasets: one from meteorological forecasting and two from medical image segmentation.

Synthetic grid (SynthGrid). We define an ensemble of N square scalar fields of size $\sqrt{M} \times \sqrt{M}$, yielding M cells ($Rows = Cols = \sqrt{M}$). The spatial domain is partitioned into K unique local clusters using two mechanisms. First, we fix the cluster size and concatenate multiple clusters horizontally and vertically, producing progressively larger scalar fields—simulating scenarios such as scaling from the Meteo to the HaN datasets. Second, we fix the field size and subdivide it into more clusters, which reduces cluster size and mimics the effect of increasing correlation thresholds (higher thresholds yield smaller clusters). A final parameter controls the proportion of unique clusters, which determines global connectivity: a proportion of 1 assigns every local cluster a unique ID (resulting in as many clusters as disconnected components), while lower proportions reduce the label set and assign IDs randomly with replacement, thereby creating global connections.

After partitioning, the ensemble is generated in two steps. First, all cells within a cluster are initialized with the same N-dimensional random vector. Second, random scaling and translation transformations are applied to each cell's vector. Applying any CC method to SynthGrid should ideally recover a grid of equally sized squares. Further details and an illustrative example are provided in the supplementary material.

Meteorological dataset (Meteo). The Meteo dataset comes from the Ensemble Prediction System (EPS) of the European Centre for Medium-Range Weather Forecasts (ECMWF). The EPS produces ensembles of predictions for several variables like pressure, temperature, and precipitation. We consider the forecast from 00:00 UTC 15 October 2012 [9]. Each forecast ensemble has fifty members obtained by varying initial conditions and an additional control run, which results in $N = 51$ members. The resulting volumetric dataset comprises $101 \times 41 \times 62$ cells. The first two dimensions correspond to latitude and longitude. The last one corresponds to slices with different geopotential height levels. We extract a 2D ensemble of scalar fields by focusing on the geopotential height level of $500hPa$, which results in members with $M = 101 \times 41 = 4141$ cells.

Medical image segmentation datasets (HaN). Ensembles of medical image segmentations can be used to model and quantify uncertainty [30], to obtain consensus segmentations [31], and to enhance treatment plan robustness by considering multiple scenarios [32]. We use an ensemble of $N = 120$ predictions for i) the right parotid gland (ParotidR) and ii) the brain stem (BrainStem) of a head and neck patient treated at HollandPTC between 2018 and 2020. The IRB approved the research protocol for the use of patient data in research, and all patients signed an informed consent form. We refer the reader to [13] for more details about the architecture and the training process of the deep neural network (DNN) ensemble used to generate the volumetric predictions. Each member in the output ensemble is a 3D scalar field of probabilities, indicating the likelihood of a given voxel being part of the foreground. For our experiments, we use intermediate axial slices of the two organs, which are easier to visualize, resulting in 2D scalar fields with $M = 128 \times 128$ cells. Typically, DNN outputs are thresholded to obtain a final binary segmentation, but we apply CC analysis directly to the probability maps, as thresholding may discard informative structural details.

5.2 Setup

We compare the unaccelerated (NoAcc) and LoGCC-accelerated variants of the Pivot and CN-Pivot CC methods. The NoAcc variants are implemented as described in Sec. 3. The LoGCC-accelerated variants share a common local step based on Alg. 1, with custom global steps for Pivot and CN-Pivot detailed in Algs. 2 and 3, respectively.

Results are reported for the two variants of the Pivot and CN-Pivot CC methods across different datasets and

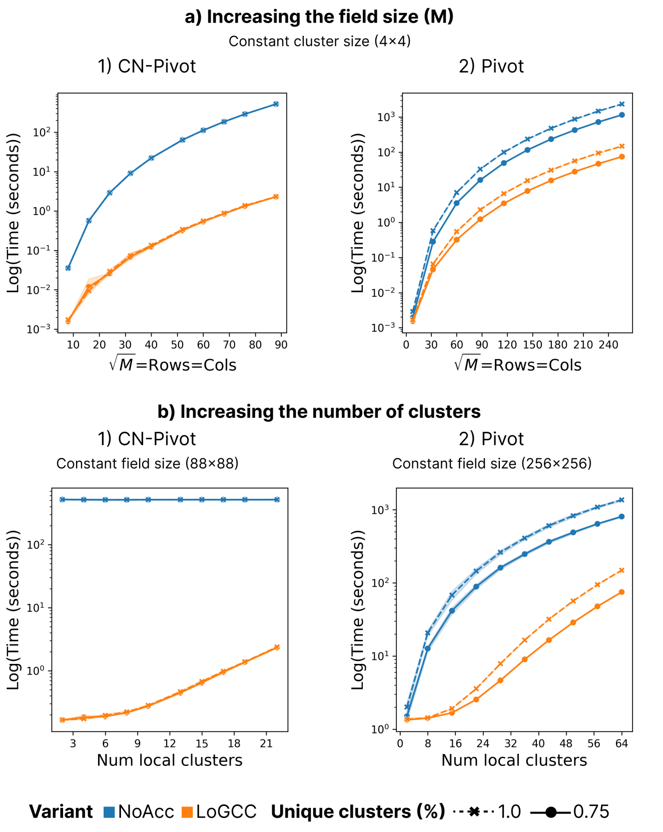


Fig. 4. Results of the scaling experiments on the SynthGrid dataset. We ran the CN-Pivot (1) and Pivot (2) algorithms with (orange) and without (blue) the proposed acceleration scheme for different scalar field sizes (a) and number of local clusters (b). Different line styles indicate the proportion of unique clusters in the grid (i.e., a cluster might contain multiple disconnected components). The x-axis encodes the changing variable and the y-axis encodes the time (logarithmic). The shaded areas indicate the 95% confidence interval obtained by performing ten trials for each method-variant combination per run.

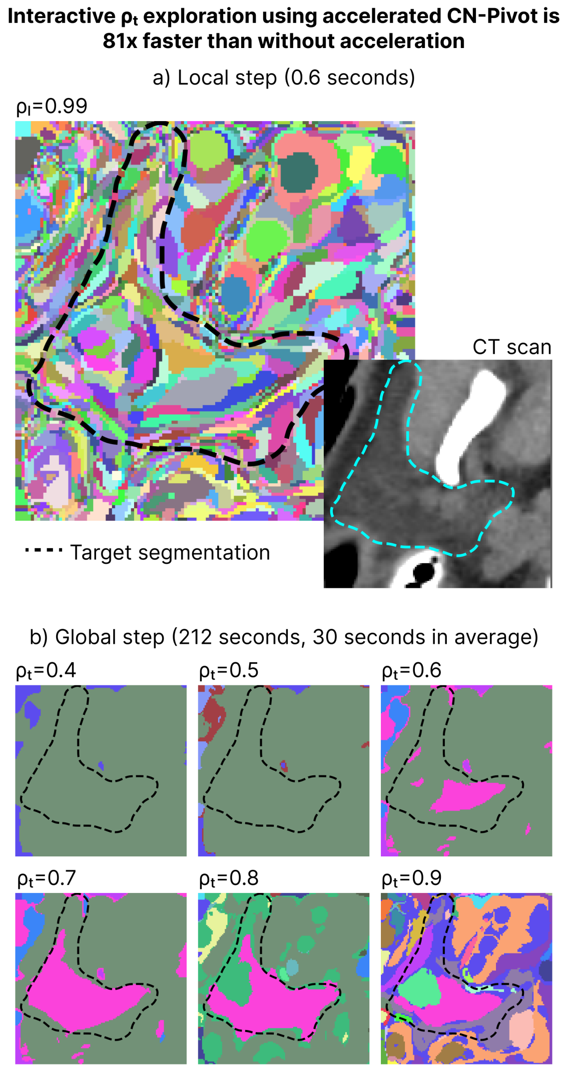

Interactive ρ_t exploration using accelerated CN-Pivot is 81x faster than without acceleration
a) Local step (0.6 seconds)
ρ_l=0.99
CT scan
Target segmentation
b) Global step (212 seconds, 30 seconds in average)
ρ_t=0.4
ρ_t=0.5
ρ_t=0.6
ρ_t=0.7
ρ_t=0.8
ρ_t=0.9

Fig. 5. Using the accelerated CN-Pivot method to explore the correlation structure of the HaN-ParotidR dataset using several correlation thresholds ($\rho_t \in \{0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$). The results of the local step (a) are re-used to compute several global clusterings (depicted $\rho_t \in [0.4, 0.9]$) in a fraction of the time (b). It takes the accelerated CN-Pivot method only 212 seconds to generate all the clusterings. In contrast, it takes the unaccelerated variant 17274 seconds, representing an 81× speedup. For context, we provide the CT scan slice for which the segmentation ensemble was computed and the target (ground truth) segmentation as a dashed line.

parameter combinations. For the Meteorological and Medical Image Segmentation datasets, the correlation threshold is varied as $\rho_t \in \{0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$. For the synthetic dataset, experiments vary the field size (with fixed local cluster size), the number of local clusters (with fixed field size), and the proportion of unique clusters. Smaller field sizes are used for the CN-Pivot variants, since their unaccelerated form can require hours even for moderate ensemble sizes. A unique combination of method, variant, and parameter values defines a CC run. Each run records the elapsed time—including local and global phases for the accelerated variants—and the resulting clusterings. Ten trials per run are executed to compute confidence intervals for scaling and quality metrics. Unless stated otherwise, $\rho_g = \rho_t$ and $\rho_l = 0.99$ are used for the accelerated variants. In the last experiment, we examine the effect of lowering ρ_l on LoGCC's performance.

To compare clusterings visually, we generate color-coded images where each cluster is assigned a unique color. One clustering is chosen as the reference, and the others are aligned to it by greedily matching clusters based on Jaccard overlap. We assign colors using the `distinctipy` library to ensure visual distinction and consistency across images. This helps track how clusters evolve across methods or parameter settings.

All experiments in this section run on a MacBook Pro (2022) with an M1 Pro processor (no GPU acceleration) and 32 GB RAM. All methods are implemented within a common Python framework[1] , using NumPy vectorized operations where possible. Optimized data structures track visited cells and cluster membership. For the local step of the LoGCC-accelerated variants, an efficient queue structure supports the breadth-first traversal.

[1] Code available at https://graphics.tudelft.nl/logcc

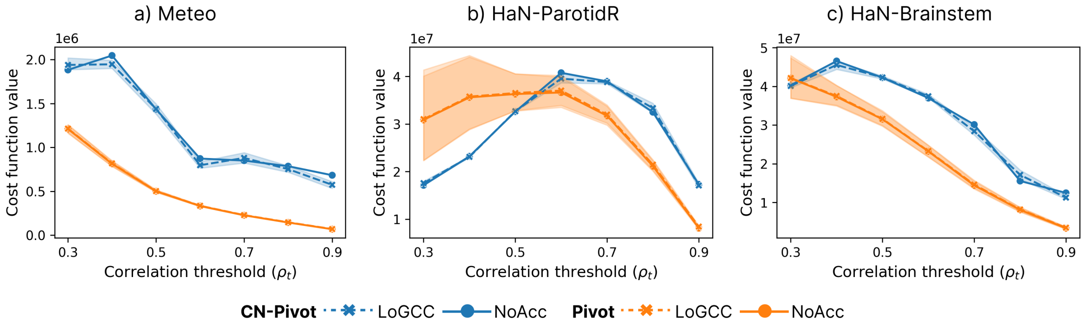


Fig. 6. Comparison of cost function values (Eq. 2) between accelerated (LoGCC) and unaccelerated (NoAcc) variants of the CN-Pivot and Pivot methods for different $\rho_t \in \{0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$ for the (a) Meteo, (b) HaN-ParotidR and (c) HaN-Brainstem datasets. The shaded areas indicate the 95% confidence interval obtained by performing ten trials per run.

5.3 Experiment 1: Scaling Behavior

To elucidate the scaling behavior of LoGCC, several ablation studies are conducted on the synthetic dataset, varying field size, correlation threshold, and number of connected components. The results are summarized in Fig. 4, where the x-axis denotes the parameter being varied and the y-axis shows the total runtime of each method in seconds (logarithmic scale).

Fig. 4(a) reports the results for varying scalar field size. The runtime gap between accelerated and unaccelerated variants widens as the field grows, with a particularly pronounced effect for CN-Pivot, which achieves more than two orders of magnitude speedup at the maximum field size $\sqrt{M} = 88$. For Pivot, the difference is smaller but still substantial: at $\sqrt{M} = 256$, the accelerated variant is more than an order of magnitude faster, with maximum observed speedups of 225× for CN-Pivot and 15× for Pivot. The accelerated variants deliver substantial runtime reductions for large fields, highlighting their effectiveness in practical settings.

Fig. 4(b) presents the scaling results as the number of local clusters increases, which corresponds to increasing the correlation threshold ρ_t while keeping the field size fixed. Consequently, increasing the number of local clusters reduces their size in pixels/cells. For CN-Pivot, the runtime gap between variants is largest when there are fewer clusters, as most of the computation occurs in LoGCC's local step, which scales linearly. As the number of local clusters grows, the global step must process more clusters, and in the extreme case where each cell forms its own cluster, the performance of both variants converges. In practice, this limit is unlikely due to gradually fading local correlations in scalar fields.

The Pivot method exhibits a slightly more complex scaling behavior due to its more favorable practical scaling. Performance tends to converge both when the number of clusters is very small or very large. For the latter, the explanation mirrors the CN-Pivot case. For the former, with few clusters, both variants have similar runtimes since the unaccelerated variant only evaluates a limited number of irrelevant cell pairs.

Finally, Fig. 4 also examines the effect of the proportion of unique clusters, simulating non-local connections. Global connections have negligible impact on CN-Pivot's runtime, whereas both Pivot variants show a systematic increase in runtime. This occurs because Pivot methods do not revisit cells already assigned to another cluster. Therefore, increasing the number of non-local connections has a similar effect to reducing the number of local clusters.

TABLE 1
Median run times in seconds for the accelerated (LoGCC) and unaccelerated (NoAcc) variants of the CN-Pivot and Pivot CC algorithms for $\rho_t = 0.9$.

Method	Dataset	NoACC (s)	LoGCC (s)	Speedup (×)
CN-Pivot	Meteo	154.59	43.81	3.53
	HaN-ParotidR	2491.61	30.94	80.52
	HaN-Brainstem	2480.31	39.10	63.44
	HaN-ParotidR-3D	-	390.08	-
	HaN-Brainstem-3D	-	1968.73	-
Pivot	Meteo	5.33	3.36	1.59
	HaN-ParotidR	5.05	1.41	3.57
	HaN-Brainstem	8.52	2.03	4.21
	HaN-ParotidR-3D	1191.07	151.61	7.86
	HaN-Brainstem-3D	3417.92	236.01	14.48

[*] The 3D datasets used with CN-Pivot consist of a subset of those used with Pivot due to the former's significantly higher runtime.

5.4 Experiment 2: Performance on Real-World Datasets

LoGCC retains favorable scaling behavior on real-world datasets. Comparing runtimes across thresholds and dataset sizes highlights the impact of these parameters on the unaccelerated variants. Table 1 reports the median total elapsed times (in seconds) for the Pivot and CN-Pivot CC methods on the three real-world datasets with $\rho_t = 0.9$. In all cases, LoGCC-accelerated methods outperform the unaccelerated counterparts. The contrast is particularly pronounced for the quadratic-time CN-Pivot algorithm, achieving speedups of up to 80×. The Pivot method also exhibits substantial runtime reductions across datasets, though these gains are more modest compared to CN-Pivot, consistent with the synthetic data experiments.

To illustrate how LoGCC enables the analysis of larger datasets, we evaluate the 3D versions of the HaN datasets. For Pivot, the full volumes, each comprising 64 axial slices of resolution 128 × 128, yield over one million spatial locations (voxels) per volume. In contrast, CN-Pivot is significantly more computationally demanding, so evaluation is limited to ten vertically centered slices. As shown in Tab.1, both the HaN-Brainstem-3D and HaN-ParotidR-3D datasets benefit

This article has been accepted for publication in IEEE Transactions on Visualization and Computer Graphics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2025.3630550

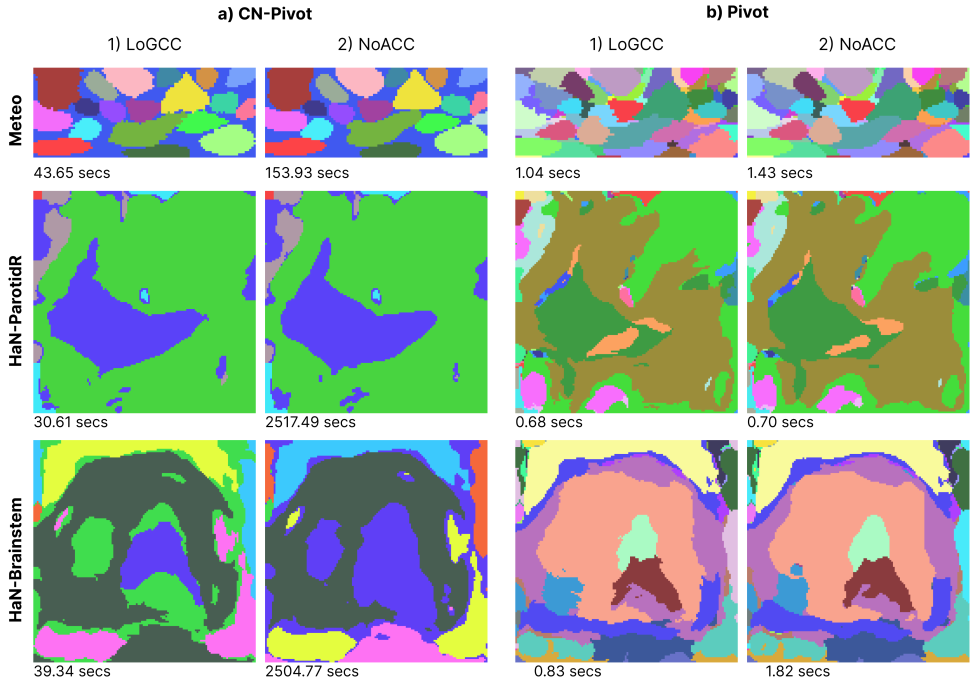


Fig. 7. Qualitative comparison of the clustering results of the accelerated (LoGCC) and unaccelerated (NoAcc) variants of the CN-Pivot (a) and Pivot (b) methods for the Meteo and HaN datasets using $\rho_l = 0.99$ (for the local step of accelerated variants) and $\rho_t = 0.7$.

substantially from LoGCC acceleration. Notably, the unaccelerated CN-Pivot variants cannot be run, as preliminary estimates indicate runtimes of approximately six hours per run. For Pivot, the observed speedups remain consistent with the order-of-magnitude improvements seen in the synthetic experiments. These gains become especially pronounced for volumetric datasets, which are otherwise infeasible to compute on the hardware used within a reasonable timeframe.

The observed speedups have a significant impact when analyzing larger datasets or exploring multiple correlation thresholds, where the extra time required by unaccelerated methods quickly accumulates. Fig. 5 illustrates this exploratory workflow on the HaN-ParotidR dataset using the LoGCC-accelerated CN-Pivot method. Fig. 5 (b) shows the sequence of global correlations derived from the same local clustering in (a). Without LoGCC's acceleration, this workflow would require over four and a half hours to complete; LoGCC reduces the processing time to under five minutes. At high correlation thresholds, the clusters closely follow the target segmentation boundary (dashed line). Combined with cell-wise statistics, these CC clustering results can support selection-based interactive segmentation workflows, enabling the extraction of the target segmentation from the ensemble. Another example of this workflow applied to the HaN-Brainstem dataset, along with a table summarizing performance differences between variants, is provided in the supplementary material.

To evaluate clustering quality, the cost is computed using Eq. 2, which measures the number of wrongly clustered cell pairs [25]. The line plots in Fig. 6 show results for different datasets and correlation thresholds. Overall, there is no significant decrease in clustering quality for either CN-Pivot or Pivot, as the lines for the accelerated variants remain close to their unaccelerated counterparts. In some cases, the accelerated variants even achieve slightly lower (better) cost values. The CN-Pivot method exhibits higher cost function values compared to Pivot, marking the first experimental comparison of these methods to our knowledge. This suggests that CN-Pivot's focus on predominant correlation structures trades off some clustering quality. Another notable observation is the increasing size of the confidence bands for the Pivot methods as ρ_t decreases. Examining the two components of the cost function reveals that this is due to greater variability in the cost associated with misassigning related cells (i.e., positive edges across clusters).

We also qualitatively analyze the clustering results. Fig. 7 shows one run ($\rho_t = 0.7$) across datasets and method-variant combinations. Overall, both variants produce similar clusterings, with LoGCC-accelerated versions recovering clusters of comparable size and shape. This is especially evident in

TABLE 2
Average speedups of the accelerated variants of CN-Pivot and Pivot (across ρ_t and runs) for different local thresholds relative to their unaccelerated counterparts.

Method	Dataset	Speedup (×)		
		$\rho_l = 0.99$	$\rho_l = 0.7$	$\rho_l = 0.5$
CN-Pivot	Meteo	3.48	946.97	1350.48
	HaN-ParotidR	75.79	5844.94	6426.89
	HaN-Brainstem	60.94	5685.81	6158.26
Pivot	Meteo	1.28	14.24	18.80
	HaN-ParotidR	1.70	4.36	4.58
	HaN-Brainstem	2.27	7.29	7.91
	HaN-ParotidR-3D	4.56	19.75	21.57
	HaN-Brainstem-3D	8.65	53.10	58.53

This article has been accepted for publication in IEEE Transactions on Visualization and Computer Graphics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2025.3630550

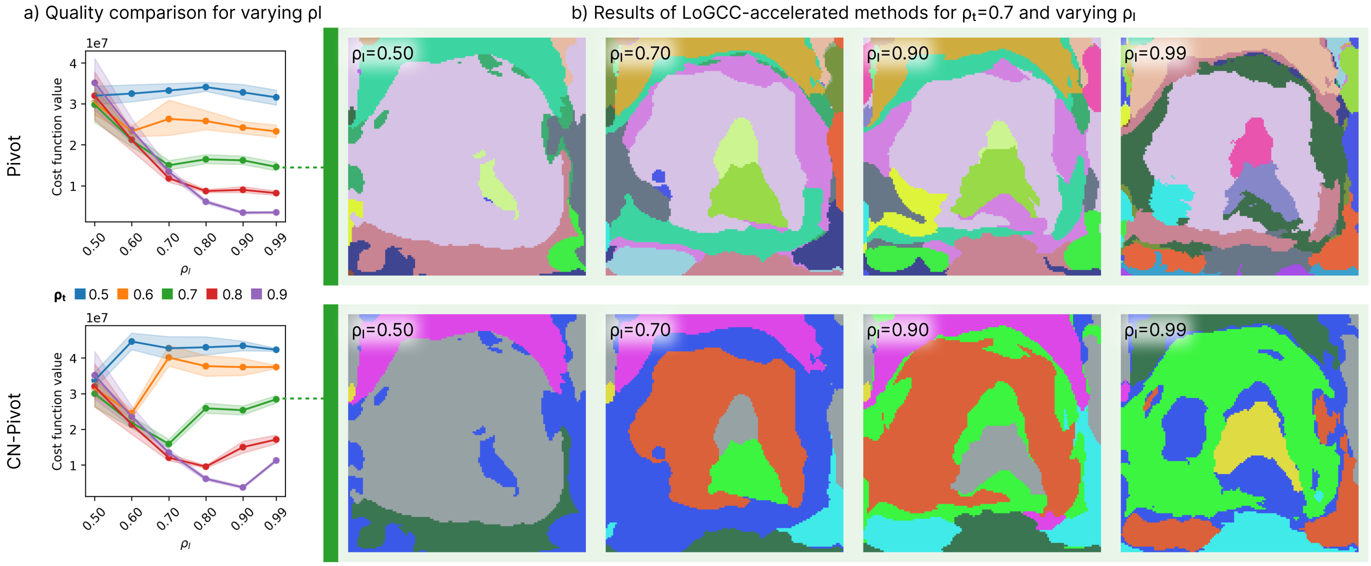


Fig. 8. Results of ablation study on the sensitivity of LoGCC to variations of ρ_l for the HaN-Brainstem dataset using the Pivot and CN-Pivot methods. (a) presents the clustering's cost according to Eq. 2. We connect points if they share ρ_t. The shaded areas indicate the 95% confidence interval obtained by performing ten trials per run. (b) shows a qualitative comparison of clustering results for $\rho_t = 0.7$ for the different ρ_l.

the HaN datasets, where large structures—such as the light and dark blue areas in HaN-Brainstem for CN-Pivot and the red area in HaN-ParotidR for Pivot—align closely. The accelerated variants exhibit slightly jagged cluster borders and minimal deviations in composition; for example, the red cluster in the lower-left corner of the Meteo dataset for CN-Pivot shifts slightly to the right with LoGCC acceleration, due to using a coarser grid in the global step. Notably, results for the two CN-Pivot variants remain highly consistent, whereas Pivot results show more variability because of the algorithm's randomized nature and sensitivity to pivot selection.

5.5 Experiment 3: LoGCC's Sensitivity to Decreasing ρ_l

So far, we have used a conservative high threshold of $\rho_l = 0.99$ for the local step, which ensures strong correlations among the members of a local cluster. As Tab. 2 shows, lowering this threshold produces additional speedups because local cluster sizes typically grow, resulting in fewer primitives to merge during the global step. In particular, reducing ρ_l from 0.99 to 0.7 generates substantial speedups, which continue to increase as the threshold decreases further. Due to the coarsening of the local step's results, we expect a tradeoff between these speedups and LoGCC's ability to produce comparable clustering. This subsection examines this tradeoff to better understand LoGCC's behavior and guide the selection of the local correlation threshold.

Fig. 8 shows the results of the sensitivity study for the clustering of the HaN-BrainStem dataset using the LoGCC-accelerated CN-Pivot and Pivot algorithms. (a) compares the cost function values for different ρ_l across global thresholds $\rho_t = \rho_g$. In the case of Pivot, we observe that the quality of the clustering remains stable until $\rho_l = \rho_t$. Lower thresholds lead to a noticeable decrease in quality (i.e., an increase in cost function value). The picture is slightly more complex for CN-Pivot. We observe that initial reductions in ρ_l preserve quality. Nevertheless, using $\rho_l = \rho_t$ yields an unexpected bump in performance (i.e., reduction in cost function value). Further reducing ρ_l does not lead to better quality, which instead degrades.

The qualitative analysis in Fig. 8 (b) provides a clearer understanding of these results. For this discussion, we focus on $\rho_t = 0.7$. For Pivot, increasing ρ_l above ρ_t increases the number of clusters while largely preserving the original structures. When $\rho_l < \rho_t$, significant detail is lost, consistent with the increase in cost function value discussed earlier. For example, at $\rho_l = 0.5$, only the largest correlation structures remain distinguishable, while smaller structures are "absorbed" by larger clusters. Similar patterns occur with the CN-Pivot method, which generally exhibits higher cost function values than Pivot. Interestingly, reducing ρ_l makes CN-Pivot clusterings increasingly resemble those of Pivot, explaining the sudden drop in the cost function at $\rho_l = \rho_t$. Overall, the sensitivity study indicates that LoGCC clusterings remain robust to reductions in ρ_l up to a point. While lowering ρ_l produces substantial speedups, it also introduces a tradeoff, as the quality of clustering deteriorates when transitivity relationships between correlations become less informative.

6 Discussion and Conclusion

In this work, we introduce the Local-to-Global Correlation Clustering (LoGCC) framework, which accelerates pivot-based CC techniques by leveraging the inherent spatial structure of scalar fields. LoGCC's local step scales linearly with the number of cells in the scalar field, resulting in significant reductions in computational overhead. The global step, while maintaining the same scaling properties as the adapted CC algorithm, operates on a reduced set of primitives generated by the local step, thus enabling substantial speed-ups without compromising the quality of the resulting clusters.

Our experiments demonstrate that LoGCC can significantly accelerate existing pivot-based CC methods such as

Pivot or CN-Pivot, preserving quality. In our experiments, we observe speed-ups of up to $15\times$ for Pivot and $200\times$ for CN-Pivot. Based on the theoretical lower bounds for correlations between clusters and empirical results, we recommend using high local correlation thresholds ($\rho_l \geq 0.96$). Our experiments show that lowering ρ_l beyond the suggested value significantly increases time gains at the cost of potentially increasing clustering variability.

LoGCC's modular design offers several advantages. For instance, it allows one to reuse the local step's results across multiple runs of the global step, reducing overhead. This feature is valuable when exploring various correlation thresholds in scenarios such as user-driven meteorological analysis [3] or connectomics [33], where precomputing the local step and determining global relationships on demand can enhance workflow efficiency. Additionally, LoGCC's two-step design enables hierarchical clustering [1], ensuring consistency in cluster memberships across different global correlation thresholds, further enhancing its utility in complex applications.

Looking ahead, there are several avenues to improve LoGCC. Our worst-case error analysis could be refined with tighter bounds, providing a more accurate theoretical performance estimate. Although our experiments focus on positive correlation, negative correlations can be considered for correlation clustering as well. An interesting direction for future work is to explore whether the proposed local–to-global methodology extends to this setting. While we used the Pearson correlation coefficient in this study, LoGCC can adapt to other correlation measures if they support the transitivity required for error bounding. When a notion of transitivity is not available or is more difficult to obtain, like in the case of non-linear measures such as Kendall's Tau and Mutual Information [34], LoGCC can still be applied but with fewer guarantees. Future research could explore applying other pivot-based heuristics, such as correlation cliques [3] and extending LoGCC to data types like high-dimensional or time-varying fields, spatio-temporal ensembles, and polygonal meshes. Additionally, accelerating LoGCC's local [35] and global [28] steps for real-time ensemble analysis remains a promising direction.

ACKNOWLEDGMENTS

This research was funded by Varian, a Siemens Healthineers company, through the HollandPTC-Varian Consortium (grant ID 2019022), and was partially supported by the Surcharge for Top Consortia for Knowledge and Innovation (TKIs) from the Ministry of Economic Affairs and Climate.

REFERENCES

[1] T. Pfaffelmoser and R. Westermann, "Visualization of global correlation structures in uncertain 2d scalar fields," *Comput. Graph. Forum*, vol. 31, no. 3pt2, pp. 1025–1034, 2012.

[2] T. Liebmann, G. H. Weber, and G. Scheuermann, "Hierarchical correlation clustering in multiple 2d scalar fields," *Comput. Graph. Forum*, vol. 37, no. 3, pp. 1–12, 2018.

[3] A. Kumpf, M. Rautenhaus, M. Riemer, and R. Westermann, "Visual analysis of the temporal evolution of ensemble forecast sensitivities," *IEEE Trans. Vis. Comput. Graph.*, vol. 25, no. 1, pp. 98–108, 2019.

[4] M. Evers, K. Huesmann, and L. Linsen, "Uncertainty-aware visualization of regional time series correlation in spatio-temporal ensembles," *Comput. Graph. Forum*, vol. 40, no. 3, pp. 519–530, 2021.

[5] N. Ailon, M. Charikar, and A. Newman, "Aggregating inconsistent information: Ranking and clustering," *J. ACM*, vol. 55, no. 5, 2008.

[6] J. Wang, S. Hazarika, C. Li, and H.-W. Shen, "Visualization and visual analysis of ensemble data: A survey," *IEEE Trans. Vis. Comput. Graph.*, vol. 25, no. 9, pp. 2853–2872, 2019.

[7] K. Pöthkow and H.-C. Hege, "Nonparametric models for uncertainty visualization," *Comput. Graph. Forum*, vol. 32, no. 3pt2, pp. 131–140, 2013.

[8] M. Zhang, Q. Li, L. Chen, X. Yuan, and J.-H. Yong, "Enconvis: A unified framework for ensemble contour visualization," *IEEE Trans. Vis. Comput. Graph.*, pp. 1–1, 2022.

[9] F. Ferstl, M. Kanzler, M. Rautenhaus, and R. Westermann, "Visual analysis of spatial variability and global correlations in ensembles of iso-contours," *Comput. Graph. Forum*, vol. 35, no. 3, pp. 221–230, 2016.

[10] T. Athawale, E. Sakhaee, and A. Entezari, "Isosurface visualization of data with nonparametric models for uncertainty," *IEEE Trans. Vis. Comput. Graph.*, vol. 22, no. 1, pp. 777–786, 2016.

[11] K. Pothkow and H.-C. Hege, "Positional uncertainty of isocontours: Condition analysis and probabilistic measures," *IEEE Trans. Visual Comput. Graphics*, vol. 17, no. 10, pp. 1393–1406, 2011.

[12] R. T. Whitaker, M. Mirzargar, and R. M. Kirby, "Contour boxplots: A method for characterizing uncertainty in feature sets from simulation ensembles," *IEEE Trans. Vis. Comput. Graph.*, vol. 19, no. 12, pp. 2713–2722, 2013.

[13] N. F. Chaves-de Plaza, P. P. Mody, M. Staring, R. v. Egmond, A. Vilanova, and K. Hildebrandt, "Inclusion depth for contour ensembles," *IEEE Trans. Vis. Comput. Graph.*, vol. 30, no. 9, pp. 6560–6571, 2024.

[14] N. Chaves-de Plaza, M. Molenaar, P. Mody, M. Staring, R. van Egmond, E. Eisemann, A. Vilanova, and K. Hildebrandt, "Depth for multi-modal contour ensembles," *Comput. Graph. Forum*, vol. 43, no. 3, p. e15083, 2024.

[15] Z. Zhang, K. T. McDonnell, E. Zadok, and K. Mueller, "Visual correlation analysis of numerical and categorical data on the correlation map," *IEEE Trans. Vis. Comput. Graph.*, vol. 21, no. 2, pp. 289–303, 2015.

[16] A. Biswas, S. Dutta, H.-W. Shen, and J. Woodring, "An information-aware framework for exploring multivariate data sets," *IEEE Trans. Vis. Comput. Graph.*, vol. 19, no. 12, pp. 2683–2692, 2013.

[17] N. Sauber, H. Theisel, and H.-p. Seidel, "Multifield-graphs: An approach to visualizing correlations in multifield scalar data," *IEEE Trans. Vis. Comput. Graph.*, vol. 12, no. 5, pp. 917–924, 2006.

[18] T. Pfaffelmoser and R. Westermann, "Correlation visualization for structural uncertainty analysis," *Int. J. Uncertain. Quantif.*, vol. 3, no. 2, 2013.

[19] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "Slic superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274–2282, 2012.

[20] M. Berenjkoub, R. O. Monico, R. S. Laramee, and G. Chen, "Visual analysis of spatia-temporal relations of pairwise attributes in unsteady flow," *IEEE Trans. Vis. Comput. Graph.*, vol. 25, no. 1, pp. 1246–1256, 2019.

[21] Y. Su, G. Agrawal, J. Woodring, A. Biswas, and H.-W. Shen, "Supporting correlation analysis on scientific datasets in parallel and distributed settings," in *Proc. 23rd Int. Symp. High-Perform. Parallel Distrib. Comput.*, ser. HPDC '14. New York, NY, USA: Association for Computing Machinery, 2014, p. 191–202.

[22] C.-K. Chen, C. Wang, K.-L. Ma, and A. T. Wittenberg, "Static correlation visualization for large time-varying volume data," in *IEEE PacificVis*, 2011, pp. 27–34.

[23] C. Neuhauser, J. Stumpfegger, and R. Westermann, "Adaptive sampling of 3d spatial correlations for focus+context visualization," *IEEE Trans. Vis. Comput. Graph.*, vol. 30, no. 02, pp. 1608–1623, feb 2024.

[24] F. Farokhmanesh, K. Höhlein, C. Neuhauser, T. Necker, M. Weissmann, T. Miyoshi, and R. Westermann, "Neural fields for interactive visualization of statistical dependencies in 3d simulation ensembles," *arXiv preprint arXiv:2307.02203*, 2023.

[25] N. Bansal, A. Blum, and S. Chawla, "Correlation clustering," *Mach. Learn.*, vol. 56, no. 1, pp. 89–113, 2004.

[26] S. Chawla, K. Makarychev, T. Schramm, and G. Yaroslavtsev, "Near optimal lp rounding algorithm for correlationclustering on

complete and complete k-partite graphs," in *ACM Symp. Theory Comput.* ACM, 2015, p. 219–228.

[27] V. Cohen-Addad, E. Lee, and A. Newman, "Correlation clustering with sherali-adams," in *Proc. - Annu. IEEE Symp. Found. Comput. Sci. FOCS*. IEEE, 2022, pp. 651–661.

[28] X. Pan, D. Papailiopoulos, S. Oymak, B. Recht, K. Ramchandran, and M. I. Jordan, "Parallel correlation clustering on big graphs," in *NeurIPS*, vol. 28, 2015.

[29] S. Behnezhad, M. Charikar, W. Ma, and L. Tan, "Almost 3-approximate correlation clustering in constant rounds," in *Proc. - Annu. IEEE Symp. Found. Comput. Sci. FOCS*. IEEE, 2022, pp. 720–731.

[30] P. P. Mody, N. C. de Plaza, K. Hildebrandt, R. van Egmond, H. de Ridder, and M. Staring, "Comparing Bayesian models for organ contouring in head and neck radiotherapy," in *Proc. SPIE Med. Imaging*, vol. 12032. SPIE, 2022, p. 120320F.

[31] M. Mirzargar and R. T. Whitaker, "Representative consensus from limited-size ensembles," *Comput. Graph. Forum*, vol. 37, no. 3, pp. 13–22, 2018.

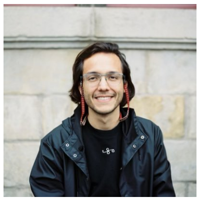
[32] E. W. Korevaar, S. J. Habraken, D. Scandurra, R. G. Kierkels, M. Unipan, M. G. Eenink, R. J. Steenbakkers, S. G. Peeters, J. D. Zindler, M. Hoogeman, and J. A. Langendijk, "Practical robustness evaluation in radiotherapy – a photon and proton-proof alternative to ptv-based plan evaluation," *Radiother. Oncol.*, vol. 141, pp. 267–274, 2019.

[33] A. K. Ai-Awami, J. Beyer, D. Haehn, N. Kasthuri, J. W. Lichtman, H. Pfister, and M. Hadwiger, "Neuroblocks – visual tracking of segmentation and proofreading for large connectomics projects," *IEEE Trans. Vis. Comput. Graph.*, vol. 22, no. 01, pp. 738–746, 2016.

[34] G. J. Székely, M. L. Rizzo, and N. K. Bakirov, "Measuring and testing dependence by correlation of distances," *Ann. Stat.*, vol. 35, no. 6, pp. 2769 – 2794, 2007.

[35] H. Liu, H. H. Huang, and Y. Hu, "iBFS: Concurrent breadth-first search on GPUs," in *ACM SIGMOD Int. Conf. Manag. Data*, ser. SIGMOD '16. ACM, 2016, p. 403–416.

7 Biography Section



Nicolas F. Chaves-de-Plaza is a PhD student at the Department of Intelligent Systems at the Delft University of Technology. He works on developing visualization and interaction tools to support clinician-driven segmentation of 3D medical images in the context of Adaptive Proton Therapy.

Renata G. Raidou is Assistant Professor (Tenure Track) at the Research Unit of Computer Graphics of the Institute of Visual Computing & Human-Centered Technology, TU Wien, Austria. She works on the interface between Visual Analytics, Image Processing, and Machine Learning—with a strong focus on medical applications. Her specific domains of expertise are Comparative Visual Analytics and Uncertainty Visualization.

Prerak Mody is a PhD student at the Division of Image Processing (Dutch abbreviation LKEB) at Leiden University Medical Center. His research focuses on clinically-applicable bayesian and interactive deep learning techniques for medical image segmentation.

Marius Staring is a Full Professor and vice director of LKEB at the Leiden University Medical Center. He and his team develop machine-learning methods for automated image analysis in clinical and life sciences. He serves as an Associate Editor of IEEE TMI and is a program committee member for MICCAI, IEEE ISBI, SPIE MI, and WBIR. He frequently open-sources his work, such as the image registration package Elastix (https://elastix.dev/).

René van Egmond is an Associate Professor of Cognitive Ergonomics at the Faculty of Industrial Design Engineering at the Delft University of Technology. His expertise lies in the fields of Product Sound Design & Perception and Informational Ergonomics. His research is focused on understanding how people process information streams in complex environments and how people process this information and deal with this complexity.

Anna Vilanova is a Full Professor in visual analytics at Eindhoven University of Technology. She leads a research group on visual analytics focusing on high-dimensional data, explainable AI, and medical visualization. She has been a member of international program committees, chair, and editor of conferences & journals in visualization. She is an elected member of the EUROGRAPHICS executive committee since 2015 and is currently president of EUROGRAPHICS.

Klaus Hildebrandt is an Associate Professor in the Computer Graphics and Visualization Group within the Department of Intelligent Systems at Delft University of Technology. His research focuses on Visual Computing, Geometric Data Processing, Physical Simulation, and Computational and Discrete Differential Geometry.