

# In-situ Visual Exploration and Analytics over Big Data\*



**rawVis**  
www.rawvis.net

## Introduction

### - Scenario

A user wishes to visually interact & analyze large data files that do not fit in main memory using commodity hardware

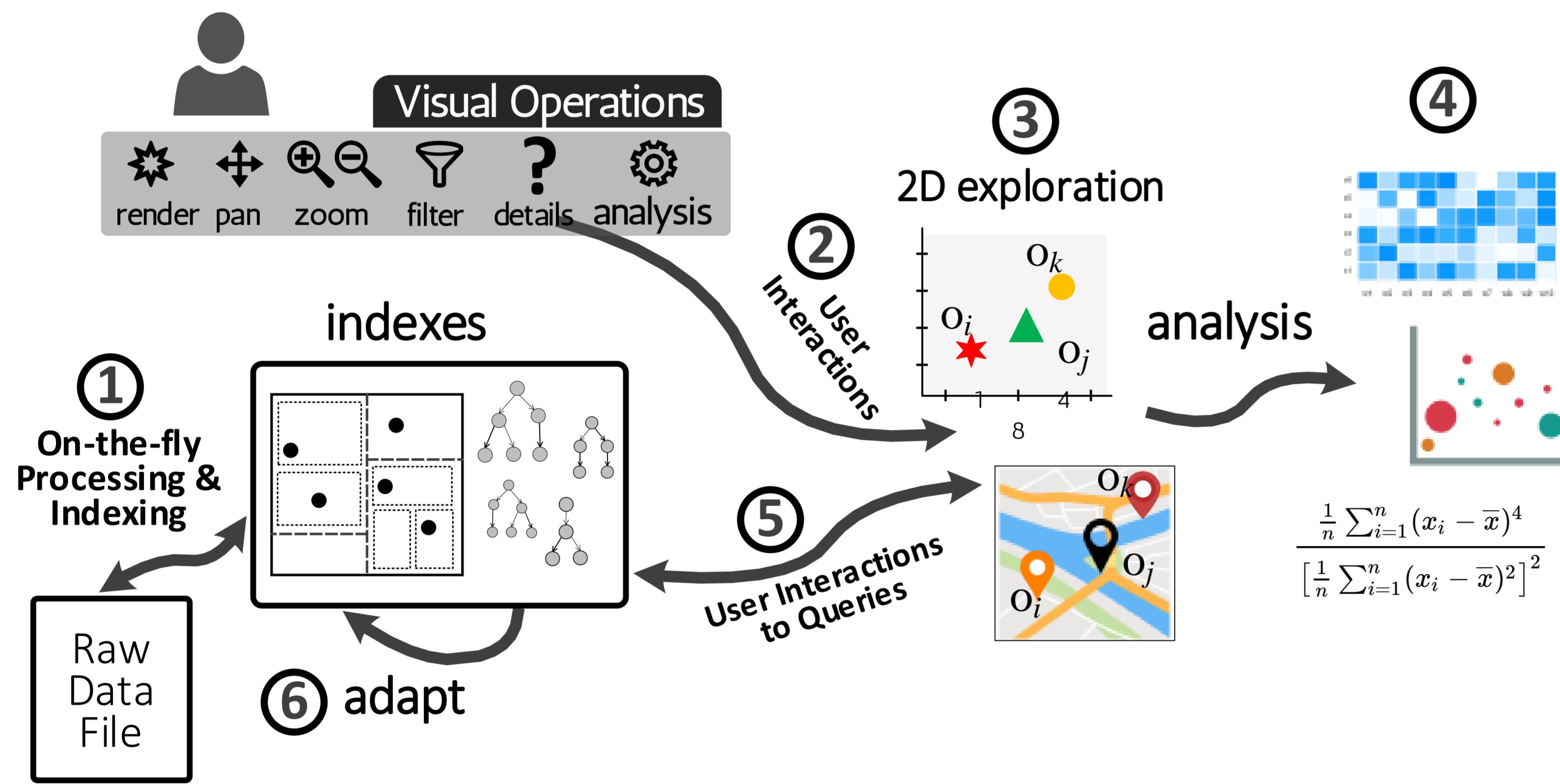
### - Setting

- > Big raw data in files, e.g., csv that do not fit in main memory
- > Limited hardware resources - commodity hardware, e.g., scientist's laptop

### - Main Challenges

- > Minimize Data-to-analysis time
  - no preprocessing phase, e.g., DBMS loading & indexing
- > Support interactive environment: response time < 1sec
  - ↓ I/O's + computations

## Working Scenario



- > **Data:** numeric, spatial, categorical
- > **2D Visual Exploration:** map, scatter plot, etc.
- > **Visual Interactions:** Pan, Zoom, Filter, Details, Analyze
- > **Visual Analysis:** bar charts, heatmaps, scatter plots, parallel coordinates, etc.
- > **Statistics:** e.g., variance, Pearson correlation, covariance

## RawVis System

### - Exploration Model

- > User interactions → Data-access operators over the index

### - Hybrid main-memory indexing scheme

- > Combines tile & tree structures
- > Constructed on-the-fly

### - Adaptive techniques

- > Based on user interaction methods progressively adjust the index structure, update & enrich index metadata
- > Adaptive-based query processing

### - Resource-aware index initialization

### - Efficient & Scalable

- > Response time: < 0.04sec (e.g., 45GB Data / 2GB RAM)
- > vs. competitors: ~ 100x faster & 2 orders of magn. fewer I/O's

### - Open-source system & Online tool

## More info

- > Resource-Aware Adaptive Indexing for In-situ Visual Exploration and Analytics, VLDB Journal, 2023
- > RawVis: A System for Efficient In-situ Visual Analytics, ACM SIGMOD, 2021
- > In-Situ Visual Exploration over Big Raw Data, Information Systems Journal, 2021
- > Home page: [www.rawvis.net](http://www.rawvis.net)

## Index Initialization

- > Index is constructed on-the-fly based on the first user interaction
- > Initially a crude index is constructed to allow fast construction
- > More fine-grained near the initial query to improve query evaluation during the initial stages of the exploration
- > Resource-aware initialization mechanism
  - The initial structure (e.g., number of tiles, stored metadata) of the index is adjusted based on the available memory
  - NP-hard problem: efficient approximation algorithms

## Hybrid Indexing Scheme

### - VETI Index (Visual Exploration Tile-Tree Index)

- > Combines a multilevel tile-based index with a categorical-based tree
- > Constructed on-the-fly
- > Stores metadata, e.g., statistics
- > Exploits metadata: ↑ memory-based computations  
↓ I/O's (↑ sequential I/O's)
- > Incrementally adapted based on user interactions
  - index structure & metadata are updated

### - Multilevel Tile Structure

- > In-memory 2D tile-based multilevel index
- > Organizes data objects into hierarchy of tiles
- > Enables efficient interactive exploration in the 2D plane, e.g., Pan, Zoom

### - Categorical-based tree Structure

- > Lightweight, memory-oriented, trie-like tree structure
- > Organizes data objects & computes statistics based on categorical attributes
- > Enable efficient categorical-based operations & analytics, e.g., group-by, filter

## Index Adaptation

### - Index Adaptation Types

- > Structure Adaptation: split & merge Tiles & Trees, expand trees, etc.
- > Metadata Adaptation: enrich & update

### - Interactive User-based Adaptation

- > Considering the locality-based characteristics of the exploration scenarios, adaptations are performed around the areas that the user exploration focuses
  - ⇒ more fine-grained indexing & rich metadata in these areas

## Index Example

