All questions are made by me. No guarantee of correctness. They might not even come to the exam. Note that there are a lot of spelling mistakes. OOPS....

SLIDE 1: INTRODUCTION

Human–Data Interaction and response-time thresholds

1. **Why do Miller/Nielsen-style response-time thresholds (0.1s, 1s, 10s) matter for visualization design?**

   People have different breaking points of how slow a visualisation is. 0.1s means that the response is instant. 1s means that the response time does not interrupt the workflow. 10s is the maximum user attention span.
   100ms should be the maximum an interactive visualization needs.

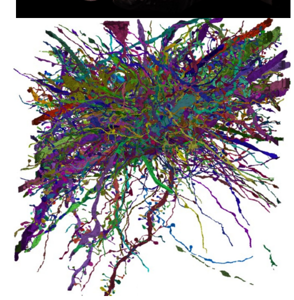
2. **Brushing & Linking vs Panning/Zooming: Pick one and explain what „real-time" means.**

   Brushing & Linking: When you select a section then everything linked to the selected part can be highlighted or filtered.
   Panning & Zooming: When you zoom in, the view transforms and there are more visible regions which leads to LOD or tile loading or re-rendering.

3. **When the system has a response time of ~500ms, which interactions still „feel" interactive? Which start to break?**

   Elements that dont need 30 fps or more will still feel interactive. Like if you click somewhere and something updates. 500ms is okay. But when you Pan or Zoom and have 500ms between every frame... That makes it unusable in most scenarios.
   500ms is below 1s so discrete actions can feel okay, but it's too slow for continuous manipulation that expects ~100ms-ish updates.

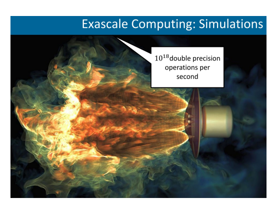
Data size, scalability, and why "better hardware" isn't the whole story



4. **Compare the Head Volume (30MB) with the Microscopy Volume (>1TB). What changes fundamentally in how you visualize them.**

   With more data, you will have to precompute some things, if possible. Maybe sample that data. Out of core streaming. You might have/need distributed storage networks to store all that data.

5. **Why can't we visualize Exascale Computing Solutions?**



   Exascale is 10^{18} double precision operations per second. Even if we try to compute them, it takes moving a lot of hindreds of Terrabytes around to accurately visualize all this data to where it can be processed/rendered. Then we need to run filtering/mapping over it too...

6. **Give two reasons why buying a faster machine might not fix interactivity for large tasks**

   Large Datasets need a very large bandwidth and the data must be stored somewhere.
   Algorithms get very complex for high dimensional data.
   When it gets too big, we as humans can not percieve the full picture anymore.
   When the dataset gets too large, we hit a limit of how much we can actually transfer. Also, we need to make the filtering, aggregating and mapping a lot faster or else all these better machines wont help anything.

Definitions and the visualization pipeline

7. **What is the pipline diagram of a visualisation?**

   Acquisition -> Filtering -> Visual Mapping -> Rendering

8. **If you speed up only Rendering, when will the user still feel lag?**

   When Filtering, Aggregating, fetching, mapping, ...
   Changing a filter recomputes aggregates. So even if you lag, the rendering is fast.

9. **Explain why some efficiency strategies target rendering only, while others target mapping and rendering together.**

   Sometimes different things cause bottlenecks. Like GPU, mapping, caching, bandwidth:
   If geometry or pixels are the bottleneck, switch to effiecient rendering, LOD, tiling
   If computing the visual representation is the bottleneck, them accelerate filtering, aggregation, mapping,etc.

Web-based interactivity: d3 vs WebGL

10. **Why can WebGL render more different individual dots in a scattrerplot than d3 without hitting it's limit?**

    D3 has a lot of overhead like DOM and SVG. WebGL directly uses the GPU. Batching, draw calls, animation loops.

11. **You must build an interactive scatterplot with 5 million points. Which architecture would you use and what interaction compromises might you consider.**

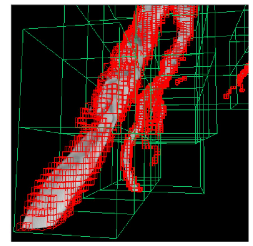
    WebGL. With Aggregation or density plots, progressive refinement and a level of detail slider.
    You can also use spatial indexing, server side aggregation or showing density and then drilling down.

Strategy selection under constraints

12. Given a scenario: interactive exploration of a global map at many zoom levels, which two strategies are most natural and why?
LOD, tiles, visibility management, aggregate visualization, data cubes, parallel processing, progressive visualization, prefetching/caching, in-situ

Tiles & caching. You dont want to render everything. Only where on the map you are right now. So only these tiles should be rendered. And when looking around, the chance is quite high that the user will come back to a previusly visited part of the map to for example compare.

13. What is empty space skipping and how does it help volumen rendering more than in for example bar charts?



[Ruijters & Vilanova, WSCG 2006]

Empty space skipping is a technique in 2D or 3D Visualisations. Basically, it tries to create a quadTree (2D) or octTree (3D) to only render/look at the sections that are really neccessary. We don't need to look over all raster but just at the ones that actually contain information.
It's especially valuable for volume rendering because you'd otherwise sample along rays through lots of empty/transparent space.

14. When do aggregate visualisation improve accuracy of insight, not just performance?

When there are so many points that we overplot. Or when we want to look at the density or find trends.

15. What is a progressive visualization?

It's about showing partial results of costly computations and refining over time. Show an approximate or partial result quickly, then refine as more computation finishes so the user can still keep exploring.
You use it because sometimes visualizing a huge dataset (>10TB) is way too costly.

16. What is prefetching and caching?

Predicting what the user might request next to send the request and cache the result before the user initiates the request.

17. What is in-situ visualization? Benefits? Drawbacks?

When the data is being processed andvisualized while the data is being generated. It avoids running huge data but it is lessflexible. It adds a lot of overhead and complexity. It is not easy to debug.

Slide 2: Fundamentals

Core definitons and scope

18. How can we efine Visualizataion?

When using a computer-supported (interactive) visual representation of (bastract) data to amplify cognition.

19. Explain, what each of these terms imply in parctise: computer supported, interactive, visual representation, (abstract) data, amplify cognition

Computer supported: Not hand drawn
interactive: We can interact with it. Like filters, tooltips, zooming, sliders ...
Visual representation: Visual perception as the representation channel. Not like text or touch.
Abstract data: Data without any spacial geometry like tables, networks, hierarchies, ...
Amplify cognition: We want to enhance edcision making to make informed decisions.

Areas

20. Differentiate Scientific Visualisation, Information Visualisation an Visual Analytics/Vsual Data Science

SciVis: Volume, Flow, Tensors, ...
InfoVis: Abstract Data like graphs and tables
VDS: automate analysis, interactive vis, human in the loop

Resource Limits

21. The slides explicitly mention limitations of computers, humans an displays. Give one concrete example.

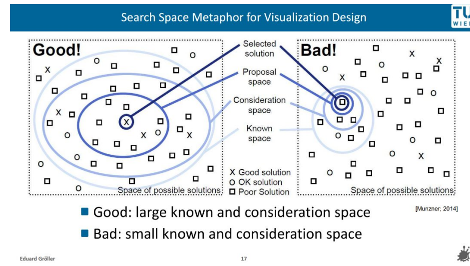
Computers: Latency, IO, Banwith, ocmputing power, memory, power
Humans: Perception, cluttering, low attention span, slow computing
displays: resolution, size, color, pixel density

22. Most visual encoings are ineffective. Give reasons as to why.

Visualizing the wrong thing, ineffective encodings, cluttering, wrong color scheme, misleading chart choice, slow interaction/ineffective, bad abstraction choice

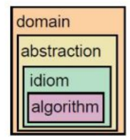
Search Space Metaphor

23. In the Search Space Metaphor graph from the slides, what do “known space”, “consideration space” and “proposal space” mean?



Know Space: broading the horizon to know what is out there. SO you are aware what exists
Consieration: Not everything that is out there can be usefull all of the time. It
s the options that you actively consider.
Proposal: Are the conidates that ypu generate and test.

Four nested level of visualization desing

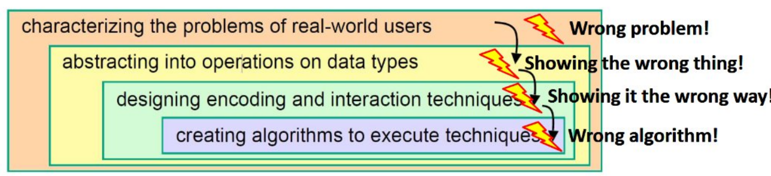


24. Explain the four nested levels Domain, Abstraction, Idom, Algorithm:

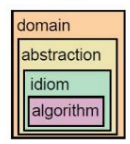
Domain: Who is the target. Depending on the target, we need to visualize differently.
Abstraction: Translate from omain specific vocabulary to that of Viz
Idom: HOW/WHAT should we show? How do we visually encode it? How do we interact with it?
Algorithm: Create an efficient algorithm for the problem.

25. What are the problems in these nessted levels? What can go wrong in each?

Domain: Identifying the wrong problem.
Abstrction: SHowing the wrong thing.
Iodm: Showing it the wrong way.
Algorithm: Wrong/inefficient algorithm.

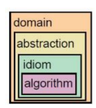


26. Imagine this: A team buils a super fast visualisation of an Aircraft carrier for a game. But noboy uses it or knows what to do with it. Which level was one wrong?



Likely the domain or maybe abstraction. They could have misunderstoo dwho the target auience was. Or they might have not been able to show the correct thing.

27. What questions must you ask yourself for each layer?



WHO? Who is the target?
WHAT? WHY? What is being shown. Why this this?
HOW? How should we show it?

28. What kind of decisions are being made in the abstraction layer? What asks the WHAT question?

How does the dataset look like? types like tables, network, trees.
How do the variables look like? Categorical? Numerical? Ordinal or Quentitative?

29. What kind of decisions are being made in the idom layer? What asks the WHY question?

We want to know the puropose of the viusalisation. Like discover trends, locate outliers or compare trends. Basically we want to decide for a matching design choice.

30. What kind of decisions are being made in the idom layer? What asks the HOW question?

It’s like how we visualize it. Encodings, manipulations, facets an reuction.

31. Explain the difference between the visual encoding idom and interaction idom.

Encoding: How we draw the data
Interaction: How users should manupulate the data and how to they should interact with the vis.

Principles / Slogans

32. Explain the slogan “No Unjustifie 3D”? Give an example for unjistified and justified 3D.

We as humans can not really comprehend the volume of 3D objects. If possile, use as little dimensions as possible.
Unjustified: Making a bar chart 3D
Justified: VOlumetric data, Flow charts, 3D anatomy

33. Explain the slogan “Eyes over memory” and give an example.

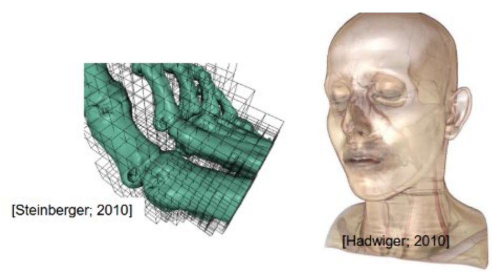
Reuce mental load by keeping comparisons visible. Like side-by-side views

34. Explain the slogan “Get it right in black and white”. What is this guaring aginst. Give a failure mode.

It explains the overreliance on color. If you cant understand the visualisation without needing colors, you should probably reconsider and reevaluate. An example would be using non-colorblind colors

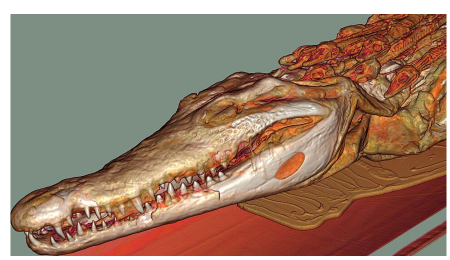
Data centric techniques

35. What is a scalar field visualisation?



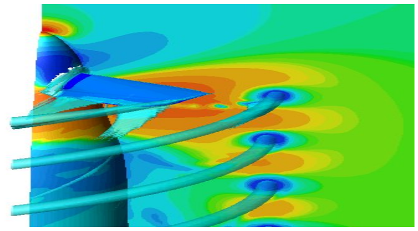
A scalar field maps each point in space (and sometimes time) to a single valie (temperature, presure, density).
It is also used to sompare surfaces with each other. Like the températeure on a 3D object like a bunny or a jet engine.

36. What is volume rendering?



We try to render only „certain parts“ of a 3D object. The „certain parts“ are specified by a transfer function to map values like color or opacity. With this, we can visualize things like the veins inside a human. Or only all bones. Basically a 2D projection of a 3D space.

37. What is isosurface rendering?

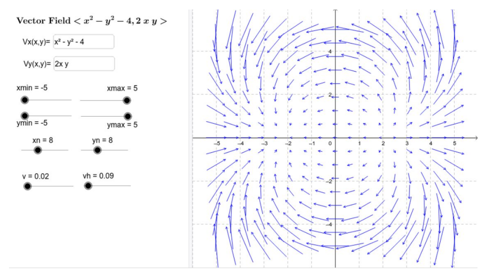


We render points on a surface that all have a specific value within a volume of space. In other words a level set of a continious function.
Examples are temperaatures inside a jet engine, stress levels of a mechanical part (like in PolyBridge), or distance to a certain object in space.

38. For a scalar field, when would you pick isosurfaces and when would you pick volume rendering? Name a tradeoff.

Isosurfaces: Highlighting a specific threshold or visualizing temperatures.
Volume rendering. Show the internal structure of a human body.
Trade-offs: Occlusion, parameter sensitivity, performance, interpretabiltiy

39. What is a vector field visualization?



The purpose is to visualize vector fields. A vector field is a collection of many many vetors. A vector is most commonly representeed by an arrow. If we then place a lot of vectors in a 2D space like a map of Europe we can visualize certain variables like wind speed. In 3D, we can simulate/visualize how objects ike cars or planes are affected liquids or gases.

40. Explain why vector field visualization needs different methods than a scalar field

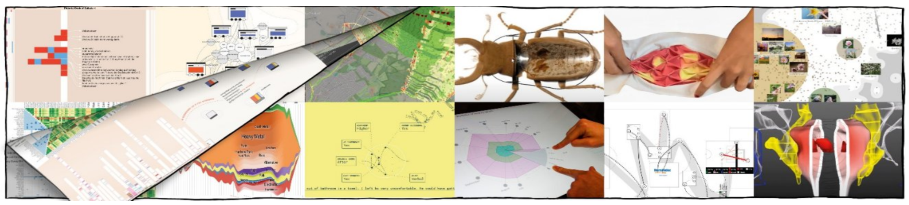
A vector field needs direction. A scalar field can only show the temperature or stress. But it can not show us from where the liquids or gases are coming from. Furthermore, it is hard to show these changes over time.

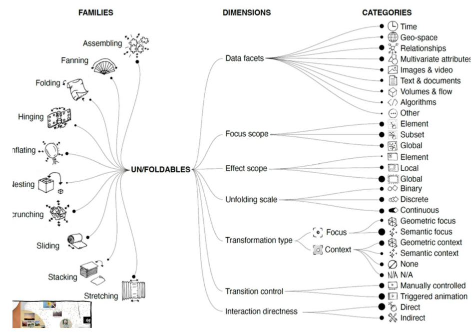
SLIDE 3: SURVEY

Fluidity revelaing Information (Un/Foldables)

41. What is an „un/foldable“ visualization?

It‘s like a tree structure, where fold in or fold out certain sections of the graph. The idea is to make a cluttered tree-like graph more readable by collapsing certain parts of the graph. It‘s an interactive mechanism that reveals or reduces information via fluid transformations.





42. What makes an „un/foldable“ visualization different from a plain filter or a hard drill-down?

They „remove“ items by hiding information. In contrast, filters remove infromation. They keep context through gradual representations in an integrated way. Instead of simply removing items or replacing the view with a deep level (hard drill-down)

43. Explain the focus score and effect scope

The focus scope is what region or items the user targets. „How much is actually un/folded?“. Effect scope is how the whole visualization changes. „Does the change stay local or does it affect the whole thing?

44. What does „unfolding scale“ mean? Why would having more states matter?

When using an unfolding scale, you can specify how deep the unfolding should occur. If you only have two states, then either everything is collapsed or nothing. But when having more than that, you can choose how deep the unfolding should be carreid out. Basically, you can gradually reveal information instead of a harsh collaps/expanse toggle.

SLIDE 4: WEBVIS TUTORIAL

Real-Time Web-Based Visualizations

45. When would you pick a SVG, canvas or WebGL for an interactive scatterplot?

SVG is very easy to implement but it has a lot of overhead and lots of nodes=lots of DOM elements. So it is only an option, when there aren‘t that many points to visualize. It has poor performance but an convenient DOM-based interaction. Canvas has better performance. But you don‘t get the DOM events fro „free“ like in SVG. WebGL has incredible perofrmance since it can use the GPU, as long as the data can be processed by the GPU in parallel. GPU-accelerated rendering scales many marks/animations at the cost of adding more complexity.

46. At 300% zoom, what differences should you see?

SVG are vector graphics. So they scale perfectly even at incredible zoom levels. Canvas and WebGL are bitmaps so they won‘t be as sharp when you zoom in.

47. Fill out this chart

	SVG (d3.js)	2D canvas	WebGL	WebGPU
Performance				
Visual scalability				
Event handling				

	SVG (d3.js)	2D canvas	WebGL	WebGPU
Performance	poor	better	Very good	Even better (?)
Visual scalability	Vector graphics	Bitmap	Bitmap	Bitmap
Event handling	Easy (DOM elements) *	Complicated *	Complicated *	Complicated *

Use whenever possible (SVG) — Little reason to use it ☺ (2D canvas) — Use whenever necessary (WebGPU)

* Depends on API

48. What makes a Network Visualization different from a regular scatterplot of what needs to be drawn and interacted with. Which technology (SCG, canvas, WebGL, ...) would you use?

In a network graph, in addition to scatterplot-like dots, there needs to be drawn the connections between nodes. With a lot of nodes and connection, this can easily be too much to handle for a SVG, canvas, and CPU. So, the best choice would be WebGL (three.js)

Slide 5: CUDA Tutorial

CPU vs GPU

49. Compare CPU computing vs GPU computing

A CPU has few cores but it excels at complex tasks. GPUs have a thousands of smaller cores who can all work in parallel. It can also do complex tasks but if the workload has many different operations, they are doing wery poorly.

50. When can data be parallelized?

GPU(s) with multiple cores, Seperable and streamable data. The data must be able to be split up and it must be able to be processed independently.

51. What different types of parallelization are there?

Task partallelisation: Decomposition into smaller tasks
Data prallelism: Apply the same operation but on different data

52. What is AMdahl‘s Law?

It specifies the theoretical speedup that you can gain by running your code in parallel.
Fuormula: S(s)=1/((1-p)+(p/2))

53. How does CUDA work (hardware perspective)?

There are multiple CUDA Grids in the GPU, depending on the model. They might consist of 4 blocks. Each block might consists of 128 Threads. But the total numbers depends on the model. The CPU talks to the GPU for instructions. The CPU allocates/copies data to GPU memory, launches a kernel with a chosen grid/block configuration, blocks get scheduled on the GPU, threads run the kernel, then results are copied back.

54. If a block has 128 threads, how many warps is that? And why does the unit „warp“ matter for performance and control flow?

128/32 = 4 Warps. Warps are the execution/scheduling unit. Warps matter, because they are the execution unit. If threads in a warp take different braches, you get a warp divergence, which reduces throughput.

Slide 6: WebGPU Tutorial

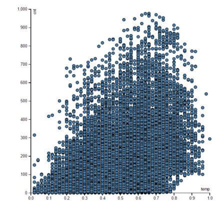
Vis WebGPU

Nothing relevant

SLIDE 7: ALGORITHMS

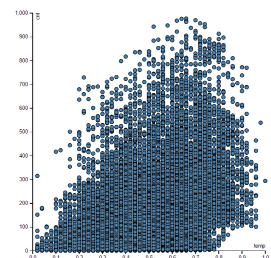
Efficient visualization algorithms and scalability factors

55. What is the difference between perceptual scaleability and interactive scaleability?



Perceptual scaleability means that ar some point, we can‘t just add more datapoints. The whole graph will get cluttered and you will not see anything. It‘s just a waste of resources. We, as huamns, can‘t even read the graph anymore.
Interactive scaleabilty is mainly about rea-time response time constraints. So many datapoints are very constly to compute and visualize. Especially when they are not even visible. This makes everything laggy.

56. Why is overplotting not just a readabilty issue? How can we fix this graph?



Of course this is a readabilty nightmare. But there are other problems like occlusion, added complexity, waste of resources, interaction lag, pointelss drawing of datapoints taht aren‘t even visinble.
We could fix this graph by using a different graph. Like a density map of hexagons, where the color of each hexagon says how many datapoints there are in this area of the hexagon.

Aggregate visualizations

57. Contrast a point map versus an aggregate map. What changes in the visual structure and what changes in the computational workload?

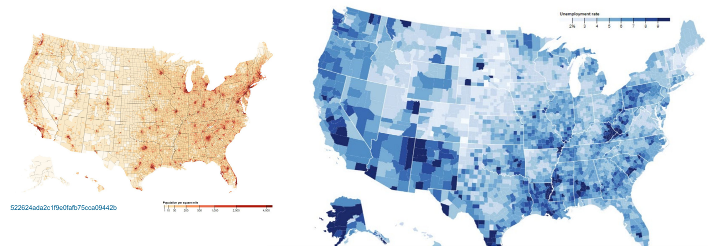
A point map is good when ther aren‘t much datapoints. It is good for a quick overview and to find trends, clusters, etc. But they hit a limit when the there are too many datapoints. That‘s where the aggregate maps come in. They aggregate (or bin) data and display a graphic that is (normally) not cluttered and can easily be interpreted. They can do that via hexagonal binning, histogram, aggregating by a common feature (like US State or county), choropleth maps, geographic binning, kernel density estimation, violin plots, density scatterplot, etc. Which makes it way faster to draw.
The visual structure gets refined and you can actually read something. But it adds complexity since you practically have an additional abstraction layer. So, there needs to be more computing involved. But it might be worth it because the resulting visualization will definitely be faster. Since, for example, many invisible points wont be drawn.
Zoom & Filter often require re-computing aggregates thus making it slower an computationally more expensive.

58. Explain the benefits of a hiostogram for aggregating.

A histogram is a 1D aggregate view where scalability comes from controlling the number of bins

59. What is a Choropleth Map?

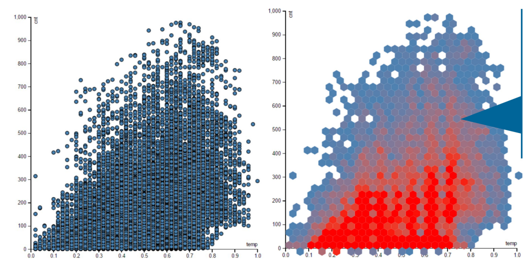
I tbrings together two datasets: Spacial data representing a partition of geographic space into stistinct districts and a variable aggregated within each district. They use a gradient colormap.



Binning

60. What is Hexagon Binning?

It is when you have a point map, that has so incredibly many poionts, that it represents a scatteplot. But most points are not visible, because other points are overlapping them. Hexagon binning means, partitioning the whole graph into hexagon bins and then counting how many values fall into each bin. Here is a before after image:

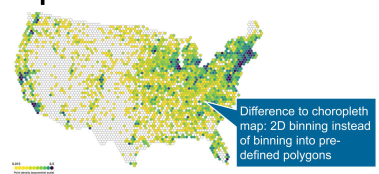


61. Why use hegaon binning instead of square binning?

Hexagons hava nearest-neighbor symmetry and max polygon sides for a regular tessellation. Squares dont have that. They do not align with each other perfectly. It‘s easier to see diffrences and gradients than in squares. Hex are more uniformly adjacent to their neighbors than squares.

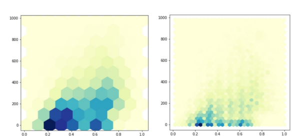
62. What is being binned in geographic binning versus a Choropleth Map?

Geographic binning: 2D binning into a regular gird/hex layout
Choropleth: binning/aggregating into predifined polygons like states or countries.

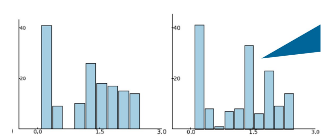


63. What is the problem with binning?

When changing the Bin size, the resulting aggregation may also change.



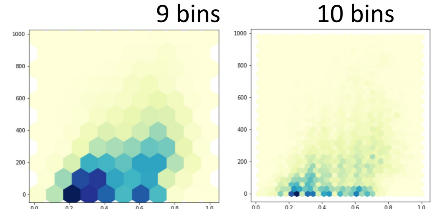
10x10 bins 22 30x30 bins



9 bins 10 bins

64. In this example, what parameters can flip just by changing the bin size from 10x10 to 30x30?

The whole map gets more granular. With this comes boudaries shifting counts, peaks appearing or disappearing, false hotspots created by discretization.



10x10 bins 22 30x30 bins

65. If you had to pick a bin size for an interactive visualization, what signals would tell you that your current bin size is too coarse or too fine?

Try out different bin sizes from coarse to fine. If the structure changes, then further investigate because coarse hides structure. In contrast, too fine just creates noise. Too fine often gives sparse and noisy bins. Too coarse hides structure.

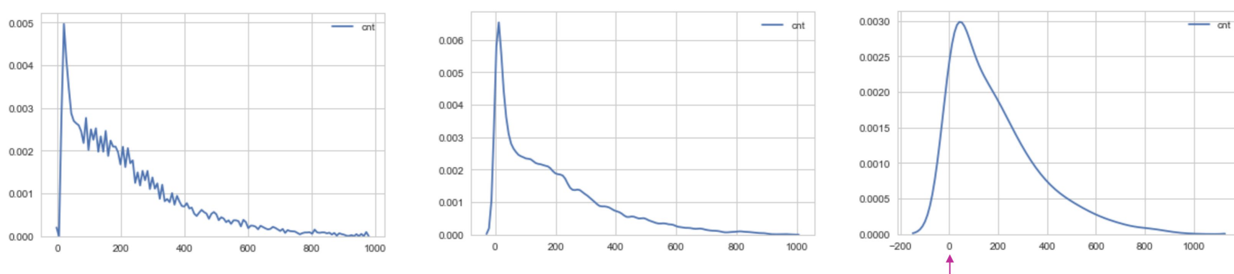
Kernel Density Estimation (KDE)

66. What is a Kernel Density Estimation?

It tries to compute the PDF (probability density function) of a certain variable. This can then be used to see what value of a variable is the most probable.

67. Explain Bandwidth in KDE using the “undersmoothed vs oversmoothed“ idea. What errors do each extreme create for interpretation?

Bandwidth means how smooth the curve should be. If the bandwith is extemely low, it might look like an Audio track. If the bandwidth is too hight, the curve looks like one continiouse curve. A moderate bandwidth should be chosen. With undersmoothing, there is a chance that the graph gets way too spikey and predicting will be quite hard and there is the problem that too much noise might distort predictions. With oversmoothing there is a chance, that no predictions can be made since the whole graph ist just one smooth curve. And sometimes the curce can even get below 0 in for example miles driven in a car, which is a violation of the Kernel function! It is not allowed below 0.



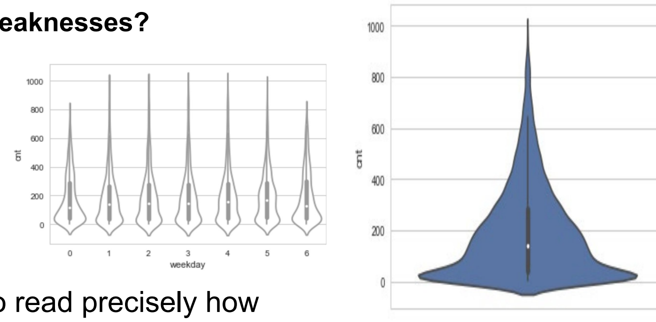
undersmoothed oversmoothed

68. When would you prefer binned aggregation over KDE (and vice versa)?

Bins are fast to calculate and can be done without much computing power. They are explicit and count based. KDE is continious and smoother put parameter sensitive, more complex and take longer to compute. It really depends on the task.

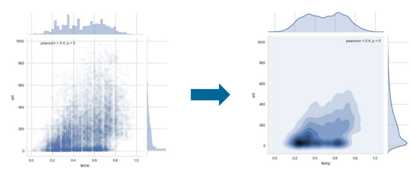
Plots

69. What is a violin plot and what are it's stengths/weaknesses?



A violinplot is like a boxplot but without quantiles. It uses a 2nd dimension (x-axis) to convey how many datapoints are in a certain region. The more, the broader the violin plot is. It's strength is that compared to a boxplot, the violin plot very clearly tells us where the biggest clustering of datapoints is. It is harder to read precisely how much is at a ceratain value. It also does not support a third dimension like color, shape, etc.

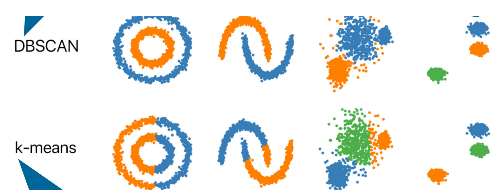
70. What is a density scatterplot?



It's like a scatterplot where the datapoints are aggregated into bins (densities). Basically, it only looks at where the scatterplot is dense and where not. It then paints only those denseity zones instead of dots like in the scatterplot.
But actually, it's a 2D KDE

Clustering

71. How is a cluster Map different from binning or KDE?



It tries to maximize the distance between two points in the same neighborhood. It doesnt try to average the data but it tries to visualize the sections that have things in common. In other words, clusters. (Answer probably wrong!)

72. For density-based clustering, what kind of problems/failures can occur?

Problems include wrong k-means threshold, too small fragments, incompatible data, too large merges, wrong noise handling.

Interactive Scaleability

73. How should you implement zooming?

Just like Google Maps. You have a very coarse overview first and only when you zoom in (or semantic zoom or drill down), you get more and more details.

74. How can you encode more information instead of just zooming?

Tooltips. When hovering or clicking over a certain object, it should display some kind of tooltip with additional information.

Spatial data structure for fast spatial queries

75. What is a KD-Tree?

It's a k-dimensional tree that recursively splits space into half spaces.

76. What is a BSP-Tree

Binary Space partitioning. It's a binary tree but with arbitrarily oriented hyperplanes. That means, if you have a 2D area that you want to partition, you don't have to partition the area in halves but in arbitrary sizes.

77. What is a Quad-Tree?

It divides Space into four euqal quadrants. It recursivly subdevides wach square until only one point is in one square.

78. Compare kd-tree, BSP-tree, and quadtree: what's the splitting rule for each, and what breaks when the data is highly dynamic (moving points)?

BSP-Tree: Binary = split in two arbitrarily big spaces
Quad-Tree: split in four equal quadrants
KD-Tree: Axis aligned split, alternating dimensions
Octree: Split into 8 equally sized quaders.

79. Given the "flat table" vs "multi-dimensional table" idea, what operations correspond to slicing, rollup, and projection—and what does brushing & linking trigger in that pipeline?

Slicing: selecting dimension values
Rollup: Aggregating up hierachy
Projection: dropping dimensions
Brushing triggers filter and re-aggregation

Progressive Visualizations

80. What does progressive visualization promise during user interaction?

It promises partial results in a short time. Like if the dataset is way too big to compute all at once, only a subset is chosen and computed. Then more data is being computed after that. That means the visualization is computed while the suer is exploting. This helps give us an understanding even though the computation is not yet completed.

81. Why should we randomly pick the datapoints that needs to be processed first?

Imagine if we pick only the first 10% to process. There might be hidden patterns like the time of day that can skew our initial visualization. Like we want to visualize car accidents. If we only take 0am to 6 am, then we dont really have that many accidents since there aren't many cars on the streets.
The proximity to the final result can be expressed as a running confidence interval.

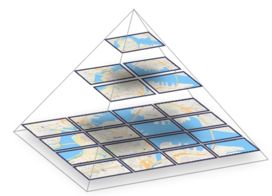
Data Tiles

82. What are data tiles?

Data Tiles are a pyramid of prerendered views across zoom levels. It‘s when you pre-render certain parts of a map so that it is available when the user wants to go there. It‘s basically a question of „What will the user look at next“.

83. Explain the tile pyramid for zooming. What gets precomputed per zoom level? How does a quadtree show up implicitly?

Imagine you have a map of the world. You don‘t want to load TB of data and display everthing. So you need LOD. When zooming into a certain section of the map, the more finer tiles will be computed and displayed. The further you go down, the more details will be shown. When you scroll, then one layer below will be precomputed. You end up with a quadtree wven though you didn‘t specify by zooming in an loading new areas.



s-coordinates-tile-bounds-projection/

[Fisher, TVCG 2007]

84. What is the difference between raster tiles and vector tiles?

Raster Tiles are bitmaps. You can not zoom in infinitely. But with vector tiles you can do that. Bitmaps are easier to create and render. Vector Tiles can be computationally expensive and they are more complex. But sometimes needed since they don‘t care about resolution when zooming in.

85. Why does can we prefetch during „idle time“?

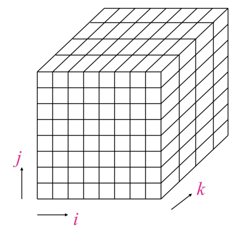
The user needs time to interpret and explore. During this time, we can already preload more fine data. Also, we can try to calculte what the user wants to do next based on their previous interactions.

Slide 8: VolVis

Volumetric Data Basics

86. **In the voxel grid (i,j,k) with one scalar L, what does it mean to „sample" the volume along it's way? What are you sampling and where do these positions come from.**

We shoot a ray from the camera out through every pixel of the „screen layer" (which is out final image in the end). If we hit something, we then determine the density. Things in the foreground with a high density will always overshadow the lower densities in the background. We can also then later use an activation function to filter only specific parts.
It's a 3D scalar field where we sample points along the way of the ray.



87. **The slides list examples like density, pressure, temperature, velocity, etc. Why is velocity tricky if the slides claim one scalar value L per data point?**

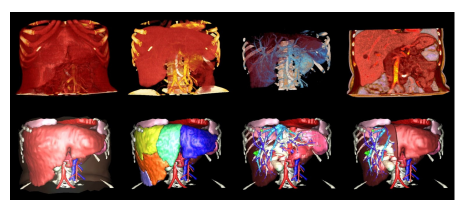
Velocity is tricky, because it is not one value. It's a vector and not a single data.

88. **What are ways to render a volume?**

2D: Slice based view (like in CT or MRT)
3D: Indirect Volume Rendering, Durface rendering, direct volume rendering.

89. **Compare indirect (surface) rendering with direct (volume) rendering in terms of what gets assigned color/transparancy.**

In indirect rendering, the exact geometry is rendered first and then only is the surface/mesh assigned. They can be rendered as one wishies. For example, if we only want to look at the bone structure of a person, we use indirect Volume Rendering because then we can see only the bones and nothing else (If we set the density to bones).
With Direct Volume Rendering, we assign color and opacity to voxels via a transfer function. If we want to set teh density to bones, we need the TF.



90. **Looking at the image grid, what kinds of anatomical/context information might you lose when moving from direct rendering to segmented/label-based surface views (or vice versa)?**

If something is uncertain or some structure have not been segmented, they will not be shown.

91. **When would a slice based view be preferable to a 3D DVR?**

When we know exactly what to look for. When we want to precisly inspect a certain part. It is often used in a clinical environment.

Direct Volume Rendering (DVR)

92. Walk us through the DVR Pipeline (Shoot a ray -> Smapling -> Classification -> Compositing) and say WHAT data is being processed.

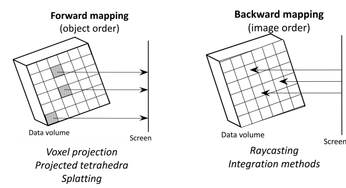
First we shoot a ray.
Second we look at where the ray intesects the Voxel grid. So, we sample position and values.
Third, each intersection is classified and mapped to optical properties (color, opactiy,...).
Lastly, they are accumulated into their final pixel value.

93. In Raycasting, what controls quality and speed?

The numbers of pixels, transfer function, sampling size, and artifacts (like missing structure) influence the quality and speed.

94. Why are people calling raycasting „Image-order (backward mapping)“ and voxel projection „Object Order (forward Mapping)“?

Image Order: We have X pixel, and we cast rays in the direction of the voxels and see what we hit.
Object order: We project the voxels on the screen.



Transfer functions

95. What is a transfer function?

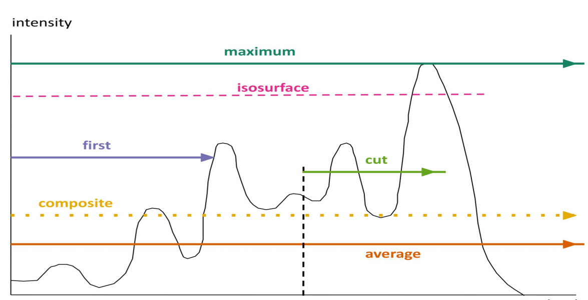
A transfer function maps density values to RBG and opacity values. You can specify at which density what kind of RBG value should be displayed.

96. Why is choosing a transfer function an iterative process? What does „find boundaries by thresholding“ mean in practise?

It really depends on what you want to visualize. If you want to visualize only the bones, then your density function needs to only look at denser parts. But materials overlap in scalar ranges. Lots of manual tuning is needed to find the best ranges.

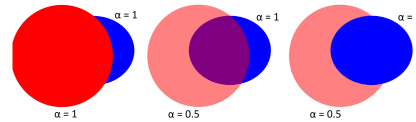
97. Compare MIP, avarage, first-hit, isosurface-like, cut, and full compositing: What does each one do in the context of transfer functions?

MIP: only show the maximum intensity
avarage: Integrate all intensities along a ray. „X-Ray-ish“ look
first-hit: Return the first hit you found above certain threshold
isosurface-like: Emphasizes boundries
cut: Render a slice/cut away region
full-compositioning: Color weighting with opacities in transfer function



98. In alpha compositioning, what does it mean for opacity to „accumulate along a ray“?

It means that when we cast a ray, we look at the first, second, third, ..., n-th Intersection and determin their alpha values. We then merge the semi transparent objects in depth oder, starting with the first. We do this for all intersection until either the alpha value >= 1 or we run out of intersections (=we are done).



Real-Time DVR & Parallel rendering

99. What are the primary problems of real-time DVR?

Mainly GPU memory and illumination. It is computationally expensive to run path-tracing. And even if we hit something, it is hard for us humans to understand. So we need some kind of shadows to understand depth. Classic bottleneck are sampling/compositioning cost & memory bandwidth

100. The slides define 2 requirements for parallel processing: Sperable and streamable. Use raycasting to explain both.

Spereable: Data can be split up. We can fire as many rays as we have pixel at the same time and process each ray and its intersections seperately.
Streamable: Data can be processed independently. For example we need to process every pixel on their own. When no pixel influences another one, we can make use of multicore processing.

Spatial data structures

101. Why is an octree described as a 3D generalization of a quadtree, and what does "regularly divide into eight equal octants" imply for adaptivity?

2D -> 4 children; 3D -> 8 children. Regular means that every division is always the same (percentual) size. 2D: 25% of the area; 3D: 1/8 of the volume.

102. The bricking slides list two purposes: fit in GPU and visibility culling. Explain the link. How does bricking enable culling?

The volume is split into chunks. We cull whole bricks instead of voxel level decisions. And then only render the needed bricks.

103. Compare octree (fixed subdivision) versus multi-resolution bricking (arbitrary subdevision). What tradeoff do you expect in efficiency?

Arbitrary subdivision is better when there are clusters of points who are very close to each other but the clusters are sperated. With arbitrary, you could clearly define the regions. But you must always calculate in which precise subtree you want to go next. This adds complexety and computational requirements.
An Octree has fixed indexing. You know exactly where to go next if you want to go deeper. There are no complicated constraints to follow for traversion.

Visibility Management

104. Distinguish view-frusum culling, occlusion culling, backface culling, and empty space culling

View-frustum culling: Cull everything that lies ouside of our FOV.
Occlusion culling: Cull everything, where the view of it is obstructed.
(Backface Culling: No need to render the backface of an object.)
Empty Space Culling: Cull transparent or irrelevant regions

105. Why does the slide say “opacity of a voxel is view-independent” but “opacity of a pixel is accumulated along a ray (view-dependent)”?

The opactiy of a voxel is defined by a transfer function. The opactiy of a pixel is determined by the accumilative opacities of the rays that hit each voxel in the way.

Empty Space Skipping

106. How can we skip empty space without looking inside a node to determine if it is really empty?

We use a min-max aggregate. It is like a histogram that tells us what is inside the node. If the transfer function maps the node‘s entire min-max range to zero, then skip.

107. What extra power do histograms give over min-max?

It gives better prediction and ccan detect most non-empty bins. But depending on the size, it costs a lot of storage and needs lots of preprocessing.

108. Why might a KD-Tree be better at empty space skipping than a octree?

WIth a KD-Tree, we can clearly define the boundaries. Sometimes in octrees we need to go tens or even hundreds of layers deep, with a good defined kd-tree we only need one or two layers. We can carve out uniform regions more tightly than with octrees.

Out of Core Rendering

109. What is out of core rendering?

It is a technique that is used to render/process scenes that are too large to handle for CPU, GPU, and/or RAM.

110. Define „working set“ in bricked single-pass rendering. Also explain why it‘s view dependent and output sensitive.

A working set is the visible set of bricks, which are stored in the brick cache. But we need those bricks to construct the current image. If we change the view, then completely different bricks will be put into the cache. It is output sensitive, because it uses technieques like empty space skipping. Thus it depends on what we actually touch and what we just pass through.

111. What problem does the „address translation from virtual space to brick cache" solve?

It maps local voxel to an address in the cache that can then be precessed and translated back into virtual memory. It is also upported by a query structure.

112. What happens, when the working set is too large for even the brick cache?

Either interrupt while rendering or use progressive rendering. You can also use thrashing and LOD fallback.

113. Ray guided rendering determines the working set on the GPU. Why is that attractive compared to CPU driven requests? What is the catch?

GPUs can process many many seperable and streamable computations at the same time. As long as the Cache does not change, we can send instructions to the GPU to use the cache and render as many bricks as possible at the same time. CPU is a major bottleneck! Problem is when bricks are missing, where then a lower LOD is used or brick is completely skipped.

114. If a brick is missing, the slide suggests „substitute with a lower resolution". Explain how ctrees make this substitution easy.

In an octree, a parent node holds coarser data about it's children. We can still render the parent brick while the children are either still precessed or bricked. 😏

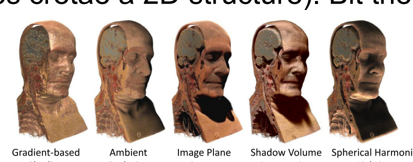
115. Compare LRU and MRU in regards caching in this setting. When might MRU beat LRU for interactive navigation?

Least recently used: Is good for when the user wants to look at many different scenes in sequence. It's great for exploring.
Most recently used: This is perfect for when a user wants only one specific thing and knows where to find it. Like if a doctor has a look at a CT scan of a tumor. He isn't interested in any other part except this one. Here it makes sense that this whole section is cached.

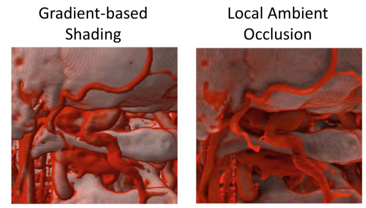
Realtime Volumetric Illumination

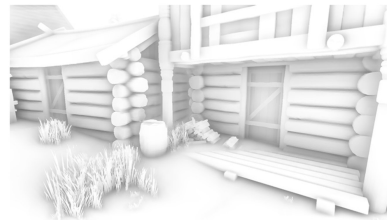
116. The slides claim that illumination helps understand structure, size, and depth. Especiall for diagnosis planning, Explain WHY those tasks benefit more than, say, a casual volume screenshot.

Humans can see depth only with either shadows or 3D (two eyes cretae a 2D structure). Bit the screen „has only one eye", wo we need shadows to convey depth. But not any kind of shadows. The best thing would be to simulate how the light works in real life. With illumination, we can easily orient in space lighting reduces ambiguity in dens semi transparent volumes. If there is only one color and no volumentric lighting, then we can only see „one plane".



117. Contrast local ambient occlusion versus dynamic ambient occlusion in terms of computation, precomputation , and what information is sacrificed.





It is an approximation technique to determine how much ambient light is reaching a certain part.
Local Ambient Occlusion is computed from the local neighborhood at runtime. Minor precomputation costs. It improves realism by darkening crevices, corner, etc ...
Dynamic ambient occlusion uses precomputation (like clustering or histograms) so runtime can support multiple effects. It sacrifices spacial detail.

Slide 9: Parallel Distributed Vis

Out-of-core visualization & Multiscale & LOD

118. If a volume dataset does not fit into GPU memory, what are concrete tactics to get around that?

Use bricks. Empty ones don‘t need to be in memory. We don‘t need full resolution for all bricks. We can use LOD.

119. Why is „Not all sub-volumes need to be rendered in full resolution“ a valid design choice?

detail can depend on view, importance, and screen-space contribution; you can refine progressively while navigating. Why waste resourrces on rendering something that we can‘t even see anyways. The detail is entirely dependent on the viewpoint. We can always progressively render the rest while the user is exploring.

Post-hoc & In-situ

120. Define post-hoc vs in-situ processing and give a practical consequence for storage and I/O.

Post-hoc means that the data is being processed AFTER being generated.
In-situ means that the data is being processed AS it is generated. With this, we can avoid writing big RAW files into storage.

121. What does steering mean? What does it add to the pipeline?

It means that you dynamically make adjustments to your configuration to get a better visualization. It‘s a feedback loop, where you „steer“ your the simulation by adjusting the configuration parameters in a way that produces a useful result. But it needs a data translation pipeline to connect the simulation to the visualization.

Coupling

122. Explain tight (inline), loose (in transit), and hybrid coupling.

Tight coupling: Simulation and Visualization runs code on the same process
Loose coupling: Simulation transfers the neccessary data over a network onto another processes or machines that then do the processing and visualization.
Hybrid coupling: A mixture of both. Some processing and visualization on the same machine as the Simulation.

123. What is a problem of tight coupling?

When the visualization crashes, then the simulation will go down too. Sharing the same process or machine drastically reduces the computing capabilities. There is a hard limit on what can be done on one single machine. Even a supercomputer is no match against thousands of GPUs in parallel

124. What is the problem of loose coupling

Timing and syncing problems, network outages, missing time slices and transfer coordination.

125. In this image, would you use tight or loose coupling? And Why?

Simulation time (sec.)	
Total	8.1659
Reaction rate evaluation	3.8779
Mixture average diffusion calculation	1.2728
Derivative evaluation	0.9246
Other temporal derivative calculation	1.7332
Runge-Kutta integration	0.3472
Tracer advection	0.0102
I/O time (sec.)	8.2675 (101.24%)
Visualization time with $2,048^2$ image resolution (sec.)	
Total	1.7699 (21.67%)
Boundary particle exchange	0.0012
Particle rendering	0.1255
Boundary voxel exchange	0.0023
Volume rendering	0.0834
Image compositing	1.5575

Here, tight coupling would be better since the I/O operations would take as long as running a whole simulation. Writing Data dominates runtime. We can not tore every timestep. This we mist compute the images/features during the run.

126. When should tight and loose coupling be used?

Tight: When we want to minimize data coordination.
Loose: When we want to maximize flexibility, and stability.

Parallelism

127. What does „streamable" and „seperable" mean:

Sperable: We can split the data into multiple pieces
Streamable: We can process the independent pieces in parallel

128. Compare data parallelism vs functional/task parallelism vs temporal parallelism: what caps scalability in each one?

Data parallelism: We split data into multiple data streams with identical rendering units. Capped by number of nodes.
Task parallelism: We split the rendering process in functions and let each function run in a different thread. Capped by the slowest stage
Temporal parallelism: We distribute the rendering of animation frames onte different machines. Capped by whether frames can be done independently and by latency/ordering constraints.

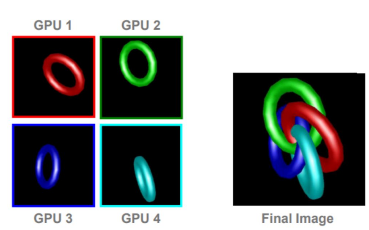
129. „send data vs send geometry vs send images". Where would you use each of those.

Send Data: When we need to remotely compute something.
Send Geometry: When the data is bigger than the geometry.
Send Images: When we want to remotely render something and want to display it somewhere else.

Visibility Management

130. What does „sort-last" actually sort?

It does not sort by last. It says, that sorting will be the last thing to do. That means everything is processed before and then in th last step sorted and merged. Like if we split up the rendering of a complex structure into its substructures and let multiple processes render each substructure seperately, then „sort-last" means that the substructures will only be sorted, when every other task has been completed.

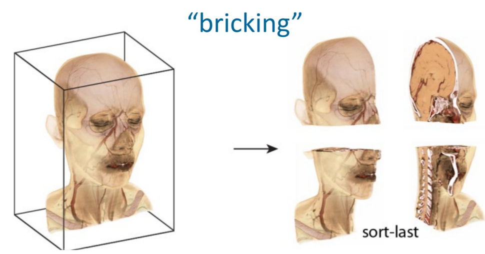


131. When do use Depth (Z-Buffer) compositioning vs Alpha (RGBA buffer) compositioning?

Depth: It allows us to overlay multiple composition based on a depth map.
Alpha: We have different RGBA Values that then get blended over to create the final image.
We use Depth for complex scenes where Alpha would be way too costly to run. Like in a forest scene where it would be needed to re-render the whole forest when using Alpha compositioning. With Depth, we could just use the depth map to re-calculate the pixels without needing to re-render the whole forest.

132. Why does „bricking“ matter specifically for sort-last volume rendering?

We can use bricks to divide our volume into small bricks that can then be rendered in seperate processes or sequentially. We split our volume into managable parts. Sort last is important so that every brick has been completely rendered and can be stiched together when everything is done. Lest we get a semi finished visualization.



133. In the slides, they mention that the „Over“ operator is associative. Wha is associativity so important for parallel computing?

Reminder: Associativity: a+(b+c) = (a+b)+c.
Because a ray can be broken into segments that then can be composited individually and combined into one final composition step.

134. Compare full-frame vs sparse merging. What drives communication cost in each, and what new bottleneck can spare merge introduce?

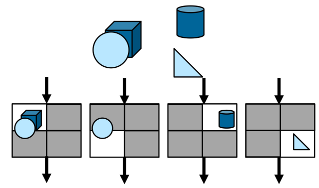
Full-frame is when you merge the full frames. The cost comes from the number of nodes and image resolution. Not the content.
Sparse merging in contrast refers to only merging a certain subset of pixels. This means only a fraction of the screen is re-rendered. This can lead to node loade imbalance during compositioning. Communication cost depends only on the amount of pixels send and the number of primitives.

135. Compare Direct-Send vs Binary-Tree compositioning. Interpret the cost n(n-1) vs log(n)

Direct-Send: Each node is responsible for one part of the image. After calculating, it sends the pixels to the correct node where it is being stiched together and sent further. This creates a lot of traffic and complexity. O n(n-1)
Binary-Tree: With each step that we traverse we halve the numbers of active nodes each step. O log(n)

136. Why do tiled displays maturally match sort-first rendering? What gets assigned to nodes?

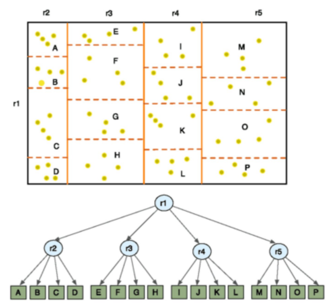
Sort-fist classifies screen regions into buckets. If we set each screen as one bucket, we get sort-first redering. Each node renders one tile. Just like NVIDIA had once the feature SLI, which specifies what region of what screen gets rendered by which GPU. Thus letting multiple GPUs render one screen.



137. Why does sort-first need screen-space bounding boxes? And why do large primitives break performance?

Bounding boxes are needed to classify primitives into screen-region buckets pre-transformation, which adds overhead. Large primitives cross boundaries, thus multiple nodes need to process them.

138. How can you balance the nodes so that everyone has roughly the same workload?



You can imagine the screen as a kd-tree. Where each node has roughly the same to do even though the areas are vastly different. It adaptively splits the screen into equal per-tile workload. Because particles often cluster, static tiling performs poorly.

Slide 10: Immersive Analytics

Immersive analysis

139. Engagement vs immersion vs embodied interaction. What is each one and what would a concrete example that fits one but not the others.

Engagement: Something that captivates you where you cant stop. Like reading a exciting book.
Immersion: Involving the user conceptually and perceptually. Surrounding their senses. IMAX
Embodied Interaction: Using ones own body to interact with a visualization. Like haptic feedbck.

140. The slides list „Natural Input -> Intuitive interaction". Where can this backfire?

Something that is for me intuitive can be super unintuitive for another person. Somtimes gestures can be amigue. Also, there is the problem with precision and accidential activation.

New Oppertunities in immersive analytics

141. Explain situated analytics with your own words. Give an example where linking data to physical objects changes the analysis outcome.

Situated analytics = analytics that is tied to the real environment / place / object and gains meaning from that context (factory, operating room, shop floor). Example: overlay sensor history directly on the specific machine component that is failing —> faster diagnosis vs reading charts elsewhere.

142. Why might „spatial arrangement of information" help (or hurt) analytic work compared to a traditional desktop.

The problem is, that not everyone has a device that supports complex spatial arrangment visualization like a VR-Headset. But it helps a lot since we can encode a third dimension. Plus the user can interact with a visualization, pick it up, and look at it from all sides without projecting 3D space onto a plane like with a Desktop Monitor.

143. The slides mention multisensory presentation. What is that? Name an analytic task that can benefit and one where it only adds nose.

It‘s immersing the user by using haptics and sound. It‘s good for looking into machines without actually opening them. But it struggle with precise quantitive reading.

Hardware and immersive interaces

144. Which hardware capabilities are „foundational“ for immersive analytics, and what breaks if one is missing?

Large FOV: It is uncomfortable if the FOV is smaller than that of the eyes.
Stereo images: We can only see 3D, if both eyes have a slightly different image.
Head, Hand, Full body tracking: Without head tracking on an HMD, we loose balance and fall.
Multi sensory presentation (audio, haptics, etc.): This enhances immersion

145. Compare Fish Tank VR, CAVE and HMDs in terms of viewpoint control and collaboration potential.

Fish Tank: You have a mechanical arm on your head that is hooked up to a computer which tracks to movement fo the head. You still control everything with a mouse. Collaboration potential: 0
CAVE: You stand in an environment where you are surounded by either projections or displays. You are wearing a special kind of goggle, which has two different polarisations for each eye. And everywhere around you, there are two images with two different polrisations projected. With the special goggles, you only see one projection for each eye. Thus creating a 3D image. Collaboration potential is decent since multiple people can be in the same room at the same time.
HMD: A display strapped to your head. The viewpoint is eather controlled by base stations or inside-out tracking. Collaboration potential is immense since everyone with a headset can collaborate.

Applications

146. Where can HMDs be used?

Medical Visualizations, Molecular Vis, Flow Vis, CLimate Data, Badminton Tactics, Trajectories

147. Across medical, molecular, flow, climate, and sports tactics examples—what common need makes immersive environments attractive?

The most important thing is that the data must inherently be 3D. Like with flow, you want to visualize how water flows around a submarine. Or air flows around a plane. And when you want to collaborate.

148. Choose one application (e.g. Flow, Climate, ...) and explain what „being inside the data“ means that a 2D desktop struggles with.

It means that you go inside. Lie you have a structure in front of you and you can explor it from all sides, includin the insides. Also, we can actually image the scales since we can see 3D in front of us in contrast to a monitor. We can also move around and (hopefully) interact with the vis. Moving

149. When is 3D justified?

When the structure is inherently spacial. DONT visualize tabular data in 3D. Humans are extremely bad at judging and comparing volumes. We want to understand the shapes of inhgernetly 3D structures not a „looks cool".

Embodied interaction

150. Explain the core idea of ImAxes and what new operations it enables compared to mouse-based axis controls.

We can manipulte the axes directly in 3D space. Thus we can see what nad how the graph changes.

151. Tilt Map: Why would „change maooing based on viewing angle" be useful, and what is a danger of andle-dependent encoding?

It is useful for switching encodings without some kind of slider or buttons. Just take the vis and tilt it a bit.
Problem is, that it is not entirely intuitive. If the mapping is wrong or flawed, then it gets a hassle to use and get useful information. Also there is a risk, that the user doesnt find all visualizations.

152. What are Scaptics and how can they be used?

It is when you have haptic vibration to convey certain information like the density of a scatterplot.

153. What are input modalities for immersive analystics?

Direct touch, gaze input (eye tracking), gestures, proxemics, wearables, speech input

Network data

154. "Networks have no inherent dimensionality." So what is the actual argument for trying true 3D network layouts in immersive systems?

Even though they hav eno inherent dimensionality, 3D gives us a better understanding of how they work. A 2D network (especially a big one) overlaps very often. But this is quite hard with a 3D Network. We can easily identfy clusters in 3D rather than in 2D.

155. Summerize the reported benefits and caviats of immersive 3D visualizations for abstract data and give a plausible reason for each caviat.

Benefits: Helps us better understand 3D Data, IF it is inherently 3D spatial. We can see the overall graph structure way easier.
Caviats: There is a chance that a 2D Vis could have been better. Another problem is cognitibe overhead and problems with spatial memory. It can also be that we dont get such a good immediate overview.

Platforms

156. Roomsized vs Tablesized vs Egocentric. What are those and what are the benefits?

Roomsized: You have a room and can walk inside the room. It‘s good for getting a feeling of the scale of everthing. Like a life-size flow vis of a car. Problem is that you have to move around and probably miss a lot of fine details.
Tablesized: This focuses purely on details. But you have to consider the reach of a person.
Egocentric: When the visualization is around you. It is better for vis where you dont really want to user to move around at all. It gives a very good overview.

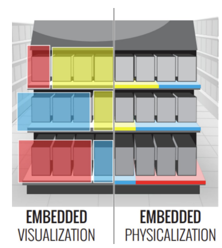
157. What are trade-offs of immersive visualizations?

Searching for certain visual features is very inefficient. Navigation and spacial orientation can be hard for some people. Motion sickness is a very real problem.

158. What’s the conceptual shift when moving from VR analytics to AR analytics?

AR just overlays information over the real world. It augments it. This opens new doors for orientation, information gathering, exploration.

159. Differentiate embedded visualizations vs embedded physicalization, and give an example of each.



Embedded visualization: visual marks are directly aligned with the object (one-to-one correspondence / registered overlay). Example: overlay stress values exactly on the corresponding part of a turbine blade.
Embedded physicalization: the data representation is physically integrated into the object/material. Example: a device casing whose geometry/texture physically encodes sensor readings.

SLIDE 11: VISUAL TEXT ANALYTICS

160. Why is "text information is unstructured" not just a storage/format issue but a visualization problem?

Storage issue: Unstructured data (like text) can not be converted into a csv table.
Vis Problem: We want to structure the data somehow. We can not visualize unstructured data. We need to derive a structure thourgh mapping, aggregating, etc.

161. What is lexical text analysis?

We split the sentences into tokens (characters, words, sentences, paragraphs, commonly occurring sequences, ...). Then we can take those tokens and use them as nodes in our tree. With this we can visualize word occurances, repetitions, context of certain words/phrases. Example: We try to build a word tree to for prediction.

162. What is syntactic text analysis?

It's like the lexical but each token has a label attached to ait according to its syntactival function. It can be used to find out grammar/structure.

163. What is semantic text analysis?

This extends the syntactic analysis. It also tries to deferr meaning and relationships with Named entities by like tagging phrases into Date, Person, Origranisation, ... Meanings and relationships are also analyized. Like Syntactic dependencies: Nound compounds, adjective modifiers, passive nominal subject. Furthermore, it also resolves references of the same item. Like „...organisation ... it ...", where the „it" references the organisation.
You can then see how certain things are connected with each other. Like people in the Panama Papers.

Single Document Visual Analysis

164. In a word cloud, why does „frequency = important" break down?

In English, there are many filler words like „the", „to", „of" that need to be filtered out. These are not important. Domain common words, redundancy (synonyms), lack of context and frequency can reflect a writer's style and not context.

165. Wordl/Word clouds can not be compared between different documents. What would you change so that two word clouds CAN be compared?

We need stable layouts and exes. Then we need a shared baseling and consistent ordering. And explicit encoding for differences. You can use color and position to encode where which word gets used more.

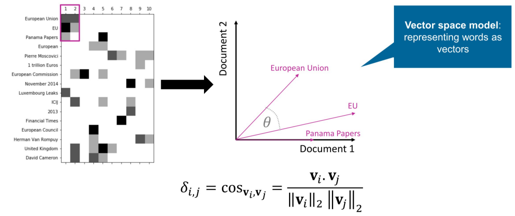
Corpus-level structure

166. What is term co-occurence?

It means that if a term occurs often with another term, then those two might be connected.

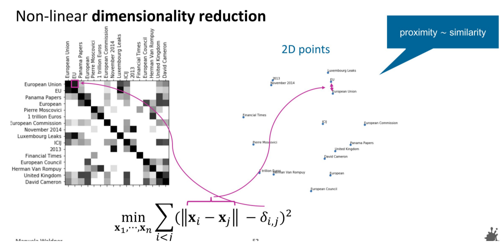
167. Explain the term-document matrix and cosine similarity in your own words.

A document matrix is an orderless representation of terms. rows = terms, columns = documents (or vice versa), cells = weight (count, tf-idf). The corpus (x-axis) corresponds to the documents and each values of each cell to the number of proximity (or some arbitrary constraint). Cosine similarity: treat each document as a vector over terms; similarity is the cosine of the angle between vectors (scale-invariant).



168. How does a similarity matrix turn into a 2D Map?

A similarity Matrix is a 2D matrix that describes the similarity of two words/tokens. These are calculated by a certain function. But if we take the similarities and then apply dimensionality reduction (e.g., MDS / t-SNE / UMAP) to place items in 2D so that distances approximate similarities, we see similar words appearing next to each other.



169. In corpus maps, what can go wrong if the proximity is interpreted too literally?

We could cluster unrelated words and could distort the projection.

170. How can we specify what words are important in some dokument?

If a word in a document appears very often but does not rally appear in most other documents, then we know that we have found a very important word. The more often it occures in out document but the less often in appears in all other documents tells us the significance.

Word embeddings

171. What are synonyms?

If there are other words for the same thing. Like Lawyer – attorney - solicitor

172. What is Polysemy?

If a word is the same but the meaning is different. Like Kiwi (fruit) and Kiwi (animal)

173. Bag-of-words struggles with synonyms and polysemy. How do embeddings help, and what do they still lose compared to context-aware models?

Static embeddings (word2vec/GloVe) do not reliably separate kiwi-fruit vs kiwi-animal — they give one vector per word form, so polysemy is still blended. They help with synonyms because similar contexts → nearby vectors.

Context-aware models (ELMo/BERT/transformers) give different vectors per occurrence, so "kiwi" in different sentences can separate.

Slide 12: Mol Vis

Ist this even relevant?

Real-Time Rendering Tricks

174. What is an impostor?

It's a pre-rendered 2D image that replaces polygons. It is used to improve perforance in real-time rendering. Especially for far-away places that don't have a high fidelity anyways.

175. Why do „standard impostors“ and sinple 2D billboards break down in dense molecular senses? And what does „correct depth output“ fix?

Billboards break because they don't write the right per-pixel depth for the represented 3D shape, so overlap/occlusion is wrong. Correct depth output makes depth testing and screen-space effects behave as if real geometry were rendered.

176. What are procedual impostors?

They are being generated analytically per pixel in a shader during runtime. In contrast to regular imposters, they are created in real-time based on the current camera viewpoint.

Scaling to huge models

177. What artifacts can appear at LOD transitions and how would you reduce them?

Popping, temporal instability, artifacts. You can mitigate them by like blending, cross-fadeing, ...

178. Explain the occlusion culling pipeline. Why render previously visible molecules first, and then build a hierachical Z-buffer with mip levels?

First, we render previously visible molecules, then we create the Z-buffer and Mip-Map and lastly the Hierachical buffer.
We render previously visible molecules first, since the camera probably doesn't change that much from frame to frame, since we know that most of them will probably still be relevant.

Illumination, Color, and Visibility Management

179. In dense data, why are cutaway views and “which proteins are visible?” a hard problem. What makes it an optimization problem in multi-instance settings?

In big models, there is MASSIVE occlusion. We have competeing goals (context & target visibility). We need to select which instances we remove and which to show.

Slide 13: Spatial Temporal Data

Since tomorrow is exam and I covered most of it already, I will keep this section very short

Statistical Attributes on Maps

180. Why is it so important in Choropleth Maps to normalize the values to make them realtive? Give a concrete example where absolute counts produce a misleading conclusion.

In a choropleth, we color areas. If we map absolute counts (extensive quantities), big or highly populated regions will often look "high" mainly because they have a bigger denominator (people, area, households), not because the phenomenon is more intense.
Choropleths work best with rates / ratios / densities (intensive quantities): per capita, per km², per household, etc.

181. „Choosing the correct mapping" what is the difference between sequential and diverging color maps? And what property of the data tells you which one to pick?

Sequential goes from Color A to Color B sequentially. Diverging has a middle Color from which two Colors sequentially diverge. Like Red->White->Blue for USA elections.
If we have a meaningful midpoint, we use diverging.

182. Spatial resolution: explain how changing the spatial aggregation unit (coarser vs finer) can flip the story a choropleth tells, even if the underlying raw data didn't change.

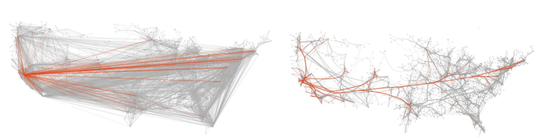
With coarser aggregation, local hotspots can get averaged away, and sharp differences can disappear.
With finer aggregation, you may reveal pockets of high/low values—but also get more variance/noise if counts are small.

Connections

183. Why does drawing all connections on OD-relations as individual edges quickly break down? What can we do?

If we draw all relations, we get a clusterfuck and can only see lines. Nothing else. What we can do, is we can bundle the Edges into bundles. Whith that we can see connections without cluttering the whole screen. We can also bundle the edges based on the forces that connect them.

184. Explain the idea of "bundle edges based on forces" in your own words and name one parameter/design choice that would strongly affect the outcome.



Force-directed bundling treats each edge like a flexible polyline with intermediate control points. Edges that are compatible (similar direction, close in space, similar length, etc.) exert attractive forces on each other's control points, pulling them into shared bundles. At the same time, forces keep edges anchored to their endpoints, so they still start/end at the correct locations.

SLIDE 14: GEN AI

Automatic Chart generation

185. Compare chart recommendation systems „from data“ vs “from natural language”. What inputs do they need, and what are typical failure modes for each?

From Data: They need data. They take the data and then decide which plot is probably gonna fit best. You just have to plop in the data and te reccommendation system will take care of the rest.
From Language: They need the data and they need the instructions what to do and they need to know the conext. So, the need to break down what your prompt actually includes and what entails. It then must be able to accurately execute the identified steps. A LLM visualization is useless, when it fails at the step „count all X that are > Y“.

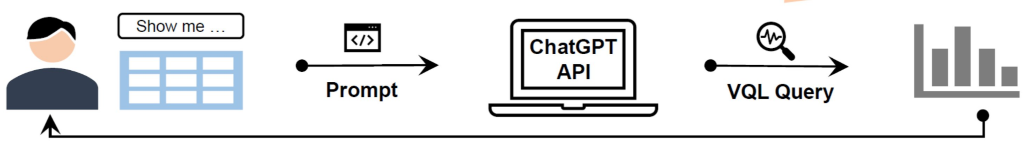
Big-picture map of “LLMs in Data Visualization”

186. The slides split the topic into: generating visualizations, understanding visualizations, and workflow integration. What does what mean? What are the problems? What formats are recommended?

Generating Vis: You write a prompt „show me ...“ and attach data and it creates you a visualization for the provided data. Problem is that LLMs also have limitations like context length. You can‘t just add all the data into the LLM. It needs the structure and then should generate code for the vis. Use machine readable formats like Json, XML, ...
Understanding Vis: The LLM needs the ability to read and understand. There can be encoding errors, interpretation errors, or multi-step reasoning errors
Workflow Integration: An LLM can act as a Connector, Simulator, programmer, and assistant. A problem is, that the LLM might not know the domain specific knowledge. It has a relatively low context window.

NL2Vis

187. In the NL2Vis pipeline diagram, where do you see the „translation boundry“ that most often causes errors?



It is during the VQL-Query step. LLMs can hallucinate and give wring answers.
But the main problem i sthat the LLM has to map ambiguous language to formal queries. Then ist must choose aggregates or bins. If the schema isn‘t directly provided in SQL, then the LLM can have problems understanding the schema itself

188. Token limits is a major problem. How would you do to process a big dataset?

You can split it into multiple chunks. Or you could use context engineering or aggregation.

189. What can go wrong with chunking, context engineering, and aggregation?

With chunking, you could use global patterns.
Context engineering can make the output biased depending on what was being provided.
Aggregation can hide outliers.

190. Explain why machine readable formats like SQL or XML help more than a list of columns.

SQL is extremely precise. Everything that is needed is written in the SQL. The LLM sees the whole structure instead of just a snapshot of the data. When using columns, the LLM does not know the relationships between the columns. Furthermore, the LLM has been trained on SQL and not on column data. It also lerans the variables (categorical/numerical) and the typical value ranges.

191. Failure analysis says that most failures are with Data rather than with Visualizations. Why? Give two examples.

Errors can happen at any point during processing the prompt. But if the LLM doesn‘t know (or hallucinate) the underlying data, it can use wrong operands like joining, or binning, or filtering.

192. What can fail during graph creation, when an LLM creates a visualization?

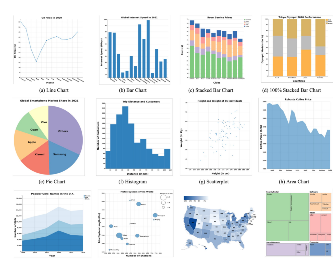
It could mislabel axes, it set opacity of important elemts to 0, it could use wrong marks or encodings. It could also swap axes.

193. The prompt-iteration strategy self-repair outperforms role-playing. Why is that?

Role-playing is good for starting a new chat. But self-repair makes the LLM focus on a certain feature that did not work. You explicity target errors. It‘s an iterative refinement instead of a one-off reasoning.

LLM Visualization Literacy

194. Define LLM Visualization Literacy



It means that the LLM can accurately extract all needed features from a graph. It must be able to read the labeles (and if there, also values). It needs to understand differences in color, height, shape, etc. Finally, it needs to take all this information, combine it, and then reason and interpret what the best answer for the question is.

195. What is the VLAT/Mini-VLAT Test?

It is a test that aims to see how good a person (or LLM) can interpret graphs, In other words, how literate the user is in reading Visualizations.

196. What can Vision Language Models do what LLMs can‘t?

VLMs can look at images and interpret them including graphics and text.

197. **The error types (where LLMs (and sometimes humans) are failing) are grouped into data, encoding, and reasoning. Give one example each, and explain which type you expect to increase most when charts use nested marks or layered encodings**

Data: Look-up/extraction errors.
Encoding: misreading labels
Reasing: If the context is poisoned or pruned, there might be a chance that the LLM goes into a whole different direction than intended.

LLMs inside workflows

198. **What is human in the loop?**

It is when a human still needs to check over the outputs of the LLM

199. **The slides list LLM roles: connector, simulator, programmer, assistant. Pick one role and describe a realistic scenario where that role helps without replacing the human analyst.**

In all of these roles will a human still be neccessary (human in the loop). Human judgement (for now) will be a vital part.
Programmer: A programmeing assistant can help solve complex tasks or even create visualizations from scratch, so the user can focus on more important or fun tasks.

200. **Give reasins why AI collaboration might fail**

Missing project background, limited interaction modalities.

Slide 15: 3D Terrain Visualization

Probably not relevant?

Slide 16: Flood Simulation

Prohbably not relevant?

Slide 17: Large Complex Graphs

Sorry no time anymore, but I will try with ChatGPT... Exam is in 30 minutes... These are all AI generated. I am sorry

Foundations

201. **A graph vis decision can be framed as What? Why? How? Pick a concrete analysis goal (e.g., "find clusters" vs "trace paths") and argue how it changes (a) the target (nodes/edges/paths/groups) and (b) the encoding choice (node-link vs matrix vs hybrid).**

If the goal is trace paths / reachability, the target is paths/edges, so a node-link view (plus highlighting) works well.
If the goal is find clusters / groups, the target is groups/communities, so you often want aggregation + node-link (meta-nodes) or a reordered matrix to make blocks pop out.

202. **Given a messy node-link diagram, choose two interaction categories from the slide (Select/Explore/Reconfigure/Encode/Abstract/Filter/Connect) and explain what each would let you do that directly addresses the mess.**

Filter: hide low-weight edges or irrelevant node types to cut clutter.
Reconfigure: change layout / collapse to super-nodes so the same data becomes readable at a higher level.

Graph basics & terminology + data structures

203. **Using the terminology slides, explain the difference between a path, a circle, a clique, and connected components—and give one quick "what might I use it for?" per term in analysis.**

Path: sequence of adjacent edges from A to B → used for routing / influence chains.
Circle (cycle): path that starts and ends at the same node → used to detect feedback loops.
Clique: every node connected to every other → used to find tightly knit groups.
Connected components: maximal sets where every node can reach every other → used to see disconnected "islands" in the data.

204. **You have a graph with millions of nodes and you repeatedly ask: "who are the neighbors of v?" and occasionally: "is there an edge (u,v)?" Compare adjacency matrix vs adjacency list vs edge list for these queries.**

Adjacency matrix: edge test (u,v) is O(1), but memory huge for millions of nodes; neighbor iteration is slower (scan a row).
Adjacency list: neighbor query "neighbors of v" is fast (iterate list), memory proportional to edges; edge test (u,v) is slower unless hashed.
Edge list: simplest storage, but both queries are typically slow unless indexed.

Complex graphs as sets & hypergraphs

205. **When would you prefer an Euler diagram over a Venn diagram for set-typed data, and what do the aesthetic principles (well-matchedness / well-formedness / area proportionality) buy you?**

Use Euler when many intersections are empty: it draws only existing regions, reducing clutter. Aesthetics: well-matchedness (shape reflects meaning), well-formedness (no weird self-intersections), area proportionality (area ≈ size) → makes comparisons and reading regions easier.

206. **The slides highlight why many-set Venn/Euler diagrams don't scale. Explain the 2^n regions issue and give two distinct failure modes that appear as n grows.**

Regions blow up as 2^n (possible intersections).
Two common failures: (1) diagram becomes unreadable/overlapping clutter, (2) it becomes hard or impossible to lay out while keeping good aesthetics (and area proportionality).

Matrix-based set/graph encodings

207. **The matrix slides list when matrices are useful vs when to avoid them. Give one scenario for each side (use / avoid) and justify using the exact constraints mentioned (reordering, scale, pixel resolution, audience familiarity).**

Use when you can reorder rows/cols to reveal blocks/patterns and the graph is dense or hairball-y.
Avoid when the dataset is so big you hit pixel limits, or when the audience expects node-link and must read paths explicitly.

208. **Compare a Venn/Euler view with an UpSet/OnSet-style view for many intersections: what changes in what the viewer can do (tasks) and what is lost?**

UpSet/OnSet scales to many intersections by listing + ranking them; great for "largest intersections?" tasks.
You lose the immediate "spatial set overlap" intuition (less direct sense of containment/shape metaphor).

Bipartite graphs + projections + hypergraph conversions

209. **Explain how (a) a biadjacency matrix relates to a bipartite graph, and (b) how a hypergraph can be viewed/converted into a bipartite representation.**

Biadjacency matrix: rows = left partition, cols = right partition; a 1 marks an edge across partitions.
Hypergraph → bipartite: treat hyperedges as nodes on one side, original vertices on the other; connect a vertex to the hyperedge-node if it's a member.

210. What is a bipartite projection, and what kind of information can it introduce or distort compared to staying bipartite?

Project onto one side: connect two nodes if they share at least one neighbor on the other side (optionally weighted by how many).
Distortion: can create very dense cliques and hides which intermediary nodes caused the connection.

Very large graphs

211. Define node aggregation into super-nodes/meta-nodes and name the three bases for aggregation listed (hierarchies, attributes, topology). For each, give a quick example of when it's the right choice.

Replace many nodes by one "group node," and summarize edges between groups.
Bases: hierarchy (org chart), attributes (same type/region), topology (community detection).
Use whichever matches your question: reporting structure vs similarity vs connectivity-driven communities.

212. Contrast "aggregation of edges" vs "aggregation of nodes (and edges)": what visual problem does each target, and what new ambiguity can each introduce?

Edge aggregation/bundling mainly reduces edge clutter but can hide exact endpoints.
Node+edge aggregation reduces both node and edge clutter but can hide internal structure ("what's inside this super-node?") and blur specific relationships.

Communities, clustering, biclustering

213. The slides describe communities via modularity (dense inside, sparse between) and show a hierarchical layout. Explain why a hierarchy is useful even if the underlying graph is "flat."

It gives multiscale navigation: overview of big communities, then drill down, while keeping a stable mental model of "where you are" in the network.

214. What is biclustering in the context of a biadjacency matrix, and what does "rearranging rows and columns to maximize modularity" actually accomplish visually?

Biclustering groups rows and columns together (both partitions) to find co-structures.
Reordering to maximize modularity makes dense blocks appear along the matrix, so communities/co-occurrence patterns become visually obvious.

Edge bundling & routing for readability at scale

215. Compare edge bundling vs an edge density map: what question does each answer better, and what detail does each hide?

Bundling answers "what are the main connectivity corridors?" but hides individual links. Density map answers "where do many edges pass?" but tells you even less about who connects to whom.

216. Edge routing can use grids/quadtree/voronoi and represent edges as polylines/Béziers. Pick one spatial structure + one curve type and argue how that combination affects (a) bundling strength and (b) geometric faithfulness.

Quadtree + Béziers: quadtree encourages stronger bundling by snapping to shared spatial channels; Béziers make bundles smooth/readable, but both can reduce geometric faithfulness (edges no longer follow the true straight-line geometry).

Data tiles / multiscale pre-rendering for real-time exploration

217. Explain the core idea of data tiles for graphs (multiple zoom levels) and identify what must be precomputed/stored to make panning/zooming responsive.

Precompute multiple zoom levels of the graph (overview aggregations down to details). Store, per level: aggregated nodes/edges (and maybe bundles), plus label sets and which links appear at that scale, so pan/zoom just swaps tiles/levels instead of recomputing everything.