Tips for Giving Clear Talks

Kayvon Fatahalian

Last update: November 2023

My motivation

- I have found I give nearly the same feedback over and over to students making talks
  - It is not profound feedback, it is just an application of a simple set of techniques and principles that are consistently useful when making talks

- I am hoping these slides serve as a useful checklist you can refer to vet your own talks before delivering them to others
  - Don't worry: I still make these mistakes all the time when creating first drafts of talks

Why give talks?

Two reasons to give talks

1. **Helping others: convey what you have learned in your research to your peers**

   (By "what you have learned"… The goal is not to tell the room what you did. Tell the room the most important things you think they <u>should</u> know, but probably do not, because they haven't spent as much time as you on your specific research problem.)

2. **Helping yourself: get feedback from others to advance your own research goals**

   (Goal: put smart people in the best position to help you.)

Two BAD reasons to give talks

1. You believe it is a reward for:
    - Finishing a project
    - Getting a paper accepted
    - Being clever enough to get a good result

2. You were signed up to speak in a research lab meeting or in CS197

A good attitude: presenting is an opportunity, not an obligation

Consider the costs of a bad (unclear) talk

- **To the audience:**
  - **1 hour x 50 people = 50 person-hours of valuable time lost**
  - **General unhappiness**

- **To you:**
  - **Missed opportunity for feedback or quality discussion**
  - **Missed opportunity to identify new collaborations**
  - **Diminished impact of your work**

Benefit TO YOU of a good (clear) talk

- **Non-linear increase in the impact of your work**
  - **Others are more likely to remember and build upon your work**
  - **Others are more likely to adopt your ideas**
  - **Others are more likely to come up to you after the talk**

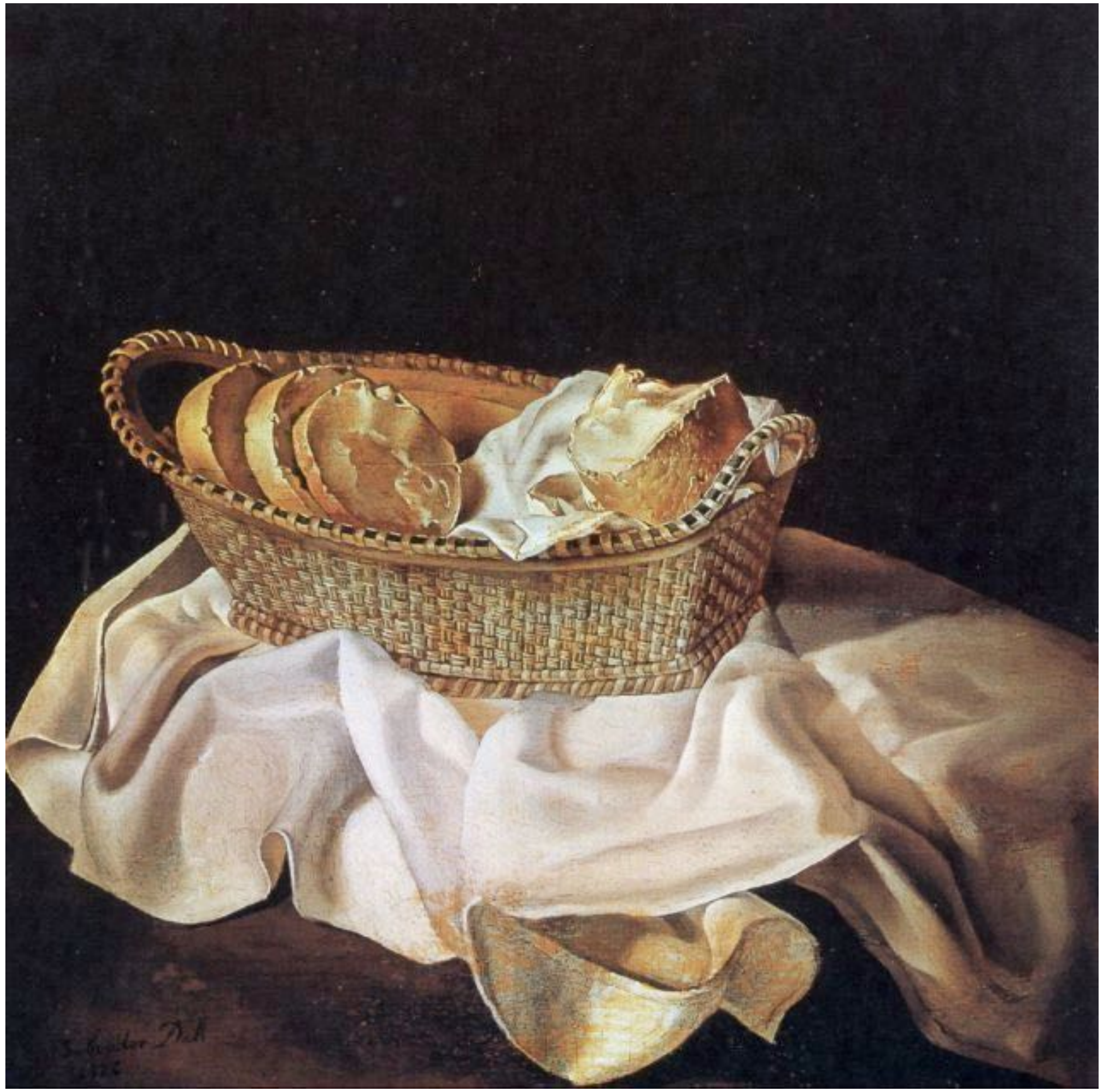
- **Clarity is highly prized in the world: the audience will remember clear communicators**
  - ***"Hey, that was a great talk yesterday... are you looking for a job anytime soon?"***
  - ***"Hey, that was a great talk, I'm working on something that you might find helpful."***

Roadmap

- The rest of this talk is structured as a list of principles and tips
    - It is not a comprehensive guide to making a talk

- So let’s get started…

Who painted this painting?



Salvador Dali (age 22)

My point: learn the basic principles before you consciously choose to break them

Tip 1

Identify your audience

Strive for clarity for that chosen audience

- You should aim for your target audience to understand everything you say in a talk
  - If they won't understand it, why are you saying it?

- This means you have to put yourself in your audience's shoes
  - Even if you are targeting experts (e.g., your research advisor or peers), experts haven't been thinking about your problem 60 hours a week
  - The ability to analyze your own talk from the perspective of others is a skill ~~young researchers~~ (all researchers, including professors) struggle with tremendously.

Tip: recite a sentence out loud to yourself. Do you really expect someone who has not been working with you everyday on the project to understand what you just said? *

* I find hearing myself say something out loud makes it easier to parse it from an audience's perspective.

Tip 2

A good principle for any talk (or paper):

“Every sentence matters”

What are you trying to say?
What technical story are you trying to tell?
What is point you are trying to make?

Is what you just said making that point? (If not, remove it)

If you can’t justify how it will help the listener, take it out.
If it won’t be understood, take it out.

An example of applying “every sentence matters”:

The talk intro

Intros and background are often very poor

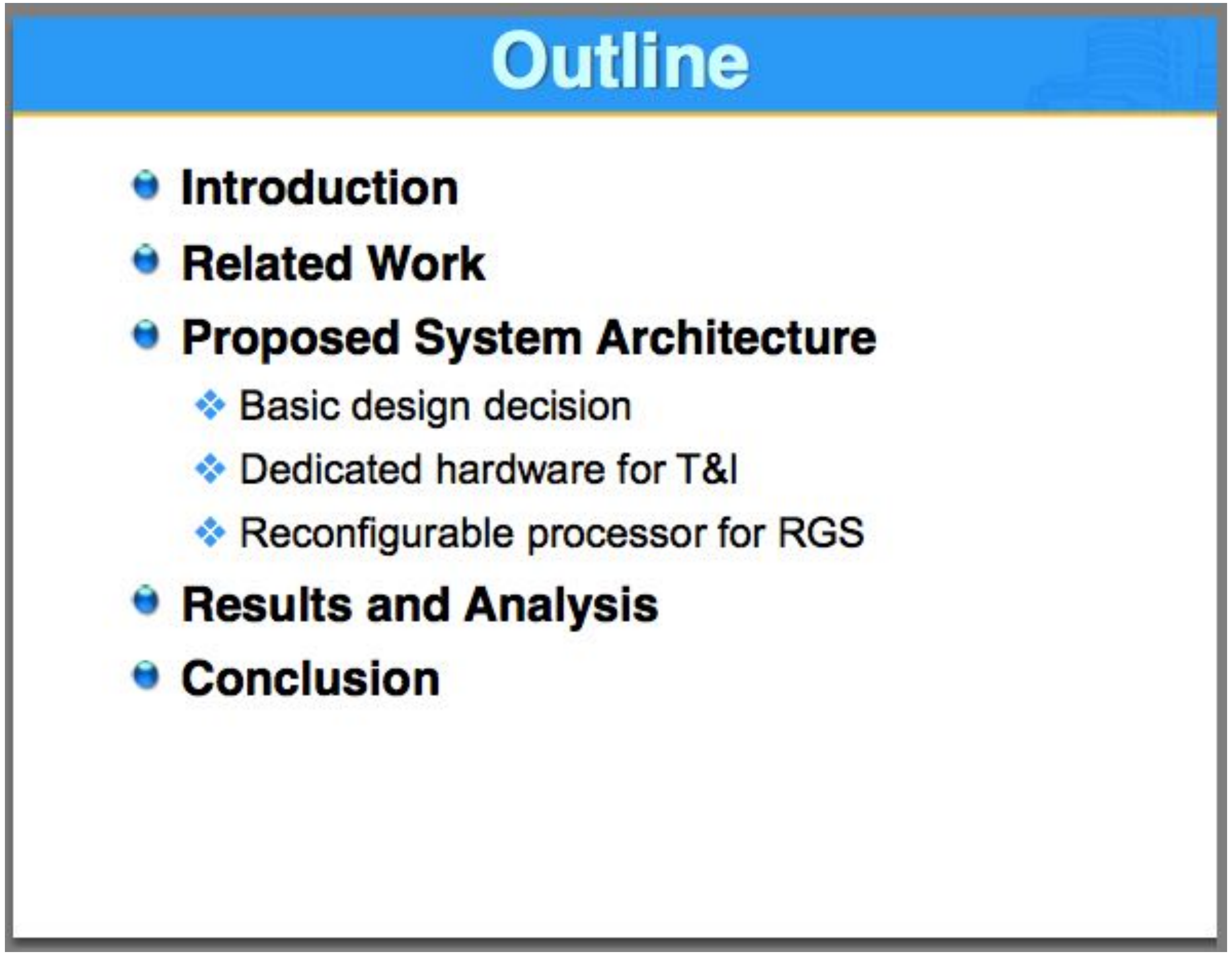
- Too many talks have introductions and related work that suggest little thought was put into them
  - Which is bad — those are critical parts of the talk!
  - Example of useless intro in 2023: “LLMs have been shown to have good task performance on a variety of tasks…” Everyone knows. So don’t say it.
- Unlike a paper, related work in a talk does not exist for academic completeness
  - No one cares if citations are comprehensive (will address this later)
- The intro/related work in a talk exist to set the context for the technical components of the talk. Specifically...

The goal of the intro and background is to tell the listener:

“In this talk, here is the way I want you to think about the problem I am trying to solve.”

Bad example 1

- **Never ever, ever, ever do this!**



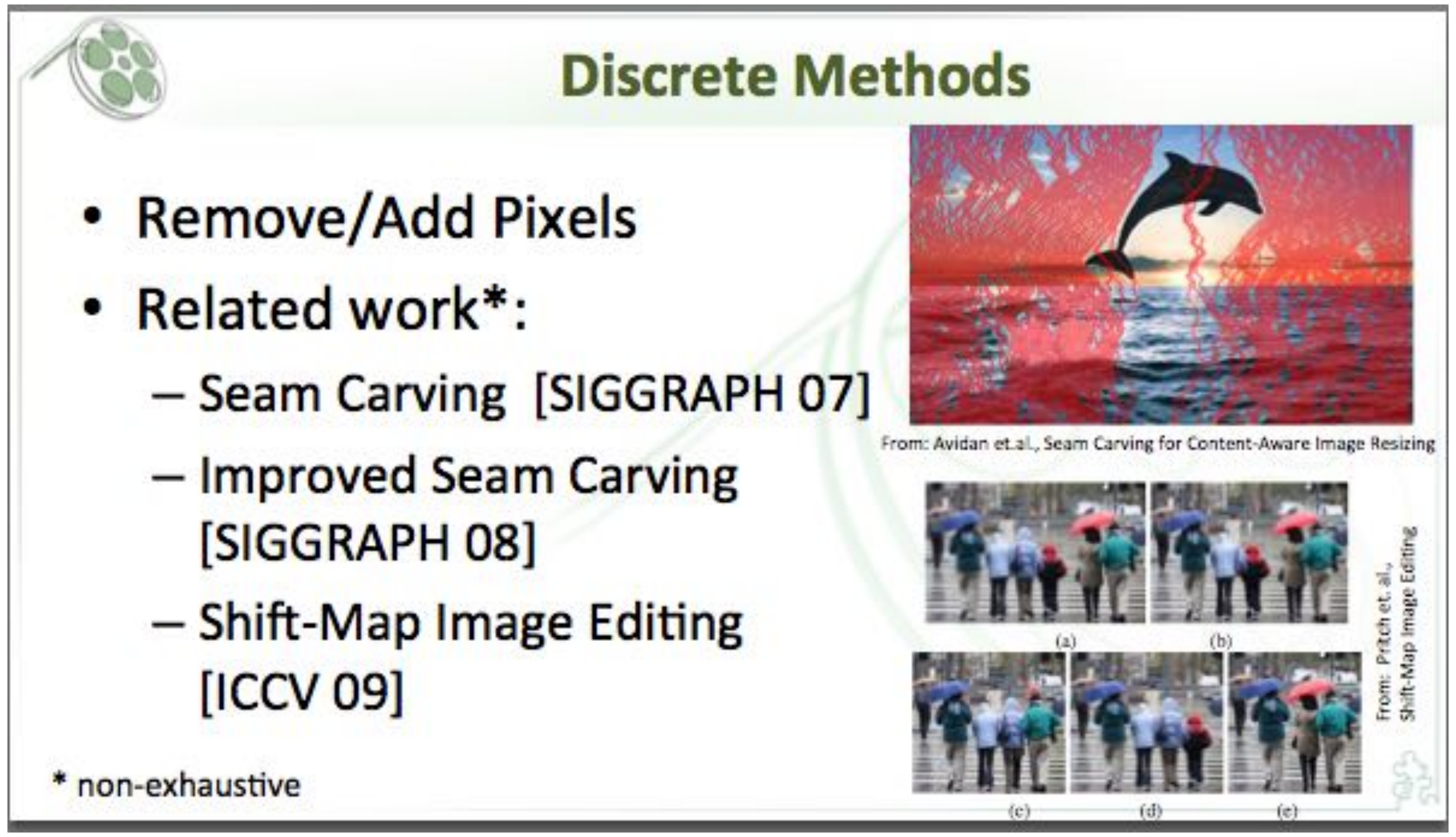
Outline

- Introduction
- Related Work
- Proposed System Architecture
  - Basic design decision
  - Dedicated hardware for T&I
  - Reconfigurable processor for RGS
- Results and Analysis
- Conclusion

This slide just told me this talk will have an introduction and a conclusion. :-(

Bad example 2

Who is the audience for this? How do these slides benefit that audience?



Discrete Methods

- Remove/Add Pixels
- Related work*:
  - Seam Carving [SIGGRAPH 07]
  - Improved Seam Carving [SIGGRAPH 08]
  - Shift-Map Image Editing [ICCV 09]

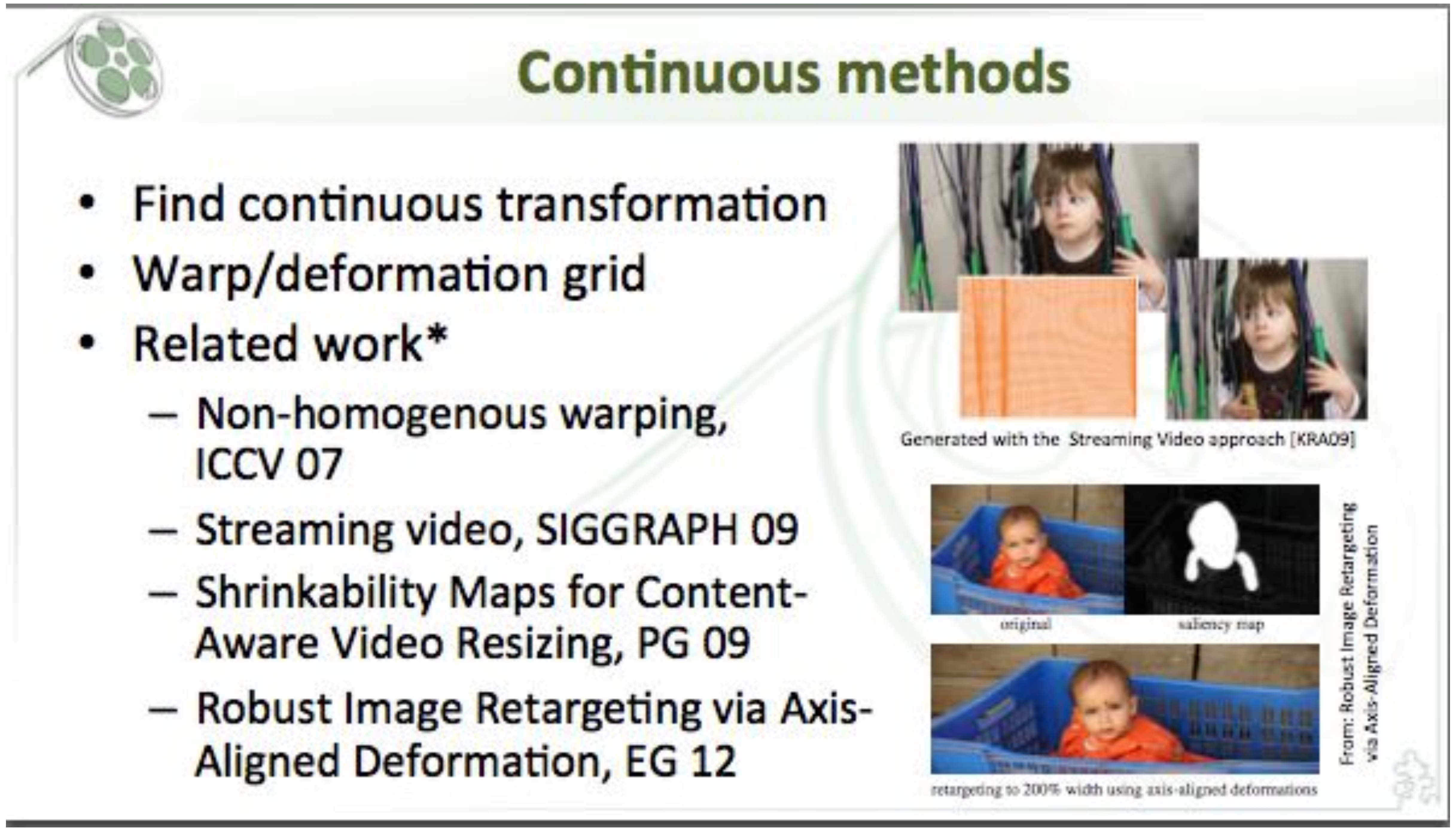
* non-exhaustive

From: Avidan et.al., Seam Carving for Content-Aware Image Resizing

From: Pritch et. al, Shift-Map Image Editing

- **Experts on the topic?**
  - **They likely know these papers exist. These slides don't tell them what about these papers is most relevant to this talk**
- **Non-experts?**
  - **They won't learn the related work from these two slides**

- **This type of related work section says little more than "others have worked in this area before".**
  - **I suspect your audience already assumes this is the case**
  - **Every sentence matters: if it doesn't provide value, take it out (or replace it with comments that do provide value)**



Continuous methods

- Find continuous transformation
- Warp/deformation grid
- Related work*
  - Non-homogenous warping, ICCV 07
  - Streaming video, SIGGRAPH 09
  - Shrinkability Maps for Content-Aware Video Resizing, PG 09
  - Robust Image Retargeting via Axis-Aligned Deformation, EG 12

Generated with the Streaming Video approach [KRA09]

original
saliency map

retargeting to 200% width using axis-aligned deformations

From: Robust Image Retargeting via Axis-Aligned Deformation

The goal of the intro and background is to tell the listener:

“During this talk, here is the way I want you to think about the problem I am trying to solve.”

An excellent strategy to catch the audience’s attention and frame the story is to make them aware that there is something they didn’t know they didn’t know.

(“You might think you know this, but here’s a new angle on it”)

Good intro example #1: [Mullapudi 2019]

The talk intro simply asked the audience to consider the following question:

Why are we working so hard to train general ML models that work in many different situations?

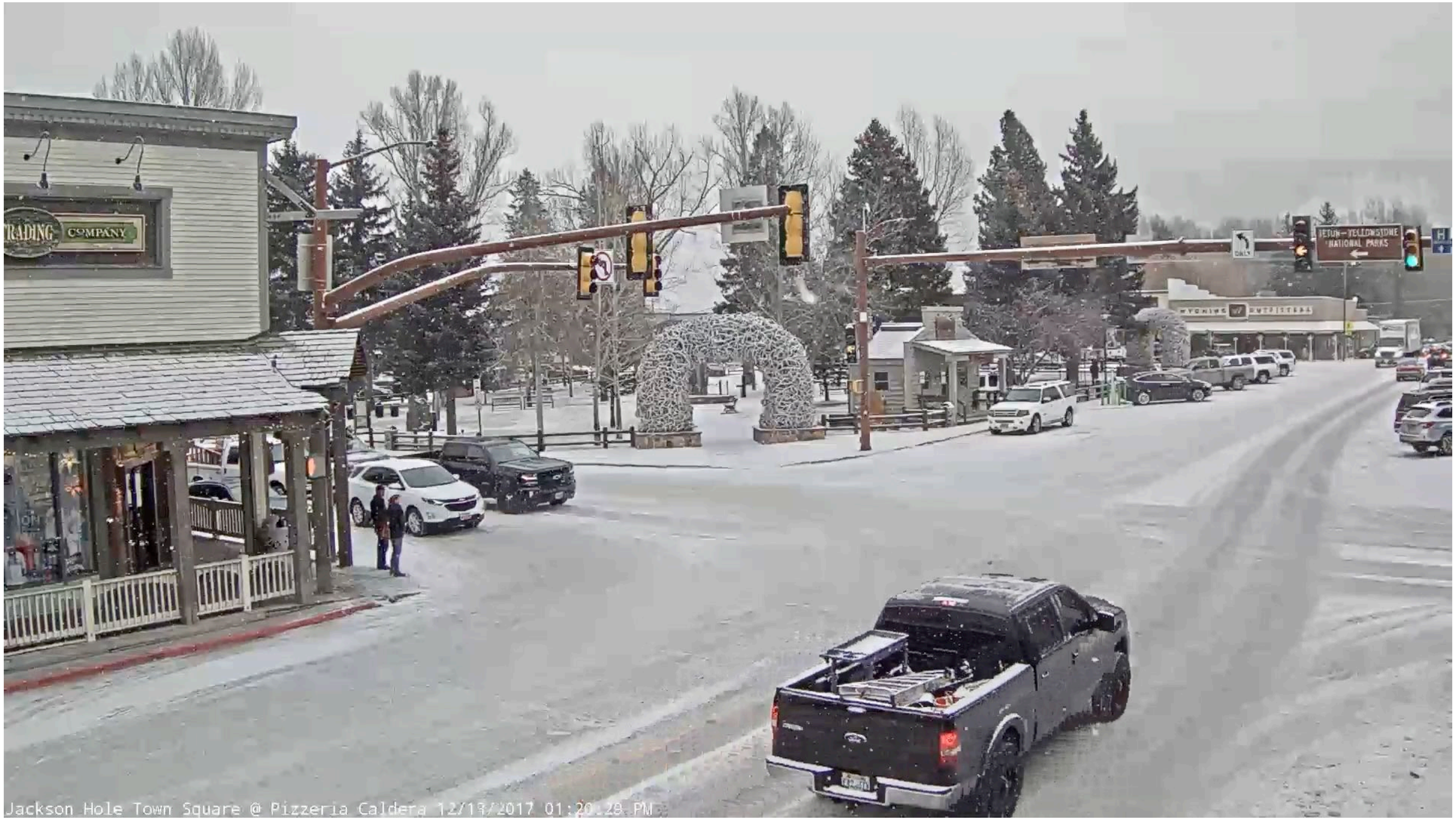
Why don’t we just use simple, less general ML models and retrain them constantly for the specific conditions at hand?”

🤔

Here’s the sequence used to do that…

This is what a traffic camera sees

It seems like it should be easy to train a simple, low-cost ML model to detect cars/pedestrians/etc in this one scene… right?


TRADING COMPANY
TETON–YELLOWSTONE NATIONAL PARKS
ONLY
Jackson Hole Town Square @ Pizzeria Caldera 12/13/2017 01:20:28 PM

Problem: distribution shift

But any one scene can change dramatically over time… weather, time-of-day, types of vehicles in view, etc…

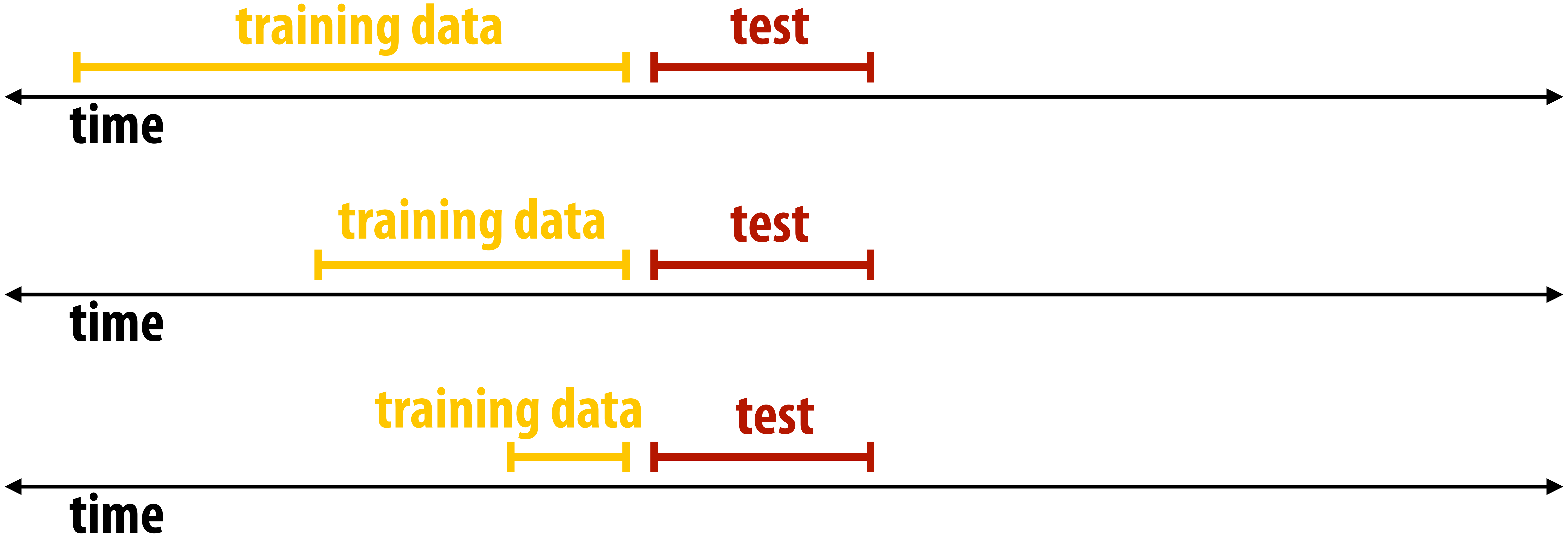
We need to acquire a training set for this diverse set of conditions… right?

Jackson Hole Town Square @ Pizzeria Caldera 12/09/2017 12:44:18 PM
Jackson Hole Town Square @ Pizzeria Caldera 12/15/2017 04:16:44 PM
Jackson Hole Town Square @ Pizzeria Caldera 12/13/2017 01:27:54 PM
Jackson Hole Town Square @ Pizzeria Caldera 12/13/2017 12:01:30 PM

Consider this experiment

- Plop camera down in a new environment
- We want a specialized model for processing the stream from this camera
- How much training data is needed to train an accurate model?

It makes the audience think a bit!



training data

test

time

training data

test

time

training data

test

time

Answer: the small amount of <u>recent</u> training data worked best.

So let's embrace continuous adaptation of ML models to the conditions at hand!

- Our premise: low cost efficient models can retain high accuracy for complex tasks in challenging environments *if they are continuously specialized to the contents of video streams*

- A.k.a. Don't worry about carefully curating the perfect training set up front, just make sure you can adapt quickly online when you see it

Audience member: "But how do I adapt an ML model quickly on the fly with little data or supervision? That sounds hard."

- Ah, that's the point of the talk…
- This speaker has something to say, maybe I shouldn't check email.

Now that we've framed the story, the related work can now be discussed <u>in the context of this framing</u>.

Establishing this framing is the primary value of the talk intro.

In this example: how have other folks tried to quickly adapt ML models to new contexts?

> *The "meta learning" field tries to train models that can be quickly adapted later using a few examples…*
>
> *There's a whole body of work on "domain adaptation": taking a model that works well in context X and modifying it to work well in Y…*
>
> *Instead we adopt an approach based on "model distillation", training a low cost student model to mimic the output of a general purpose teacher model.*

Tip 3

Set up the problem:
establish inputs, outputs, and constraints
(goals and assumptions)

Establish goals and assumptions early

- "Given these inputs, we wish to generate these outputs…"
- "We are working under the following constraints"
  - Example: the outputs should have these properties
  - Example: the computer graphics algorithm...
    - Should run in real time
    - Should be widely parallelizable so it can run efficiently on a GPU
    - Should not require artist intervention to get good output
  - Example: the system…
    - Need not compile all of Python, only this subset that we care about…
    - Should realize about 90% of the performance of hand-tuned code, with much lower development time

Why is knowing the goals and constraints important?

Your contribution is typically a system or algorithm that meets the stated goals under the stated constraints.

Understanding whether a solution is "good" requires having this problem context.

Tip 4

Show, don’t tell
It’s much easier to communicate with figures and images than text

(And it saves the speaker a lot of work explaining)

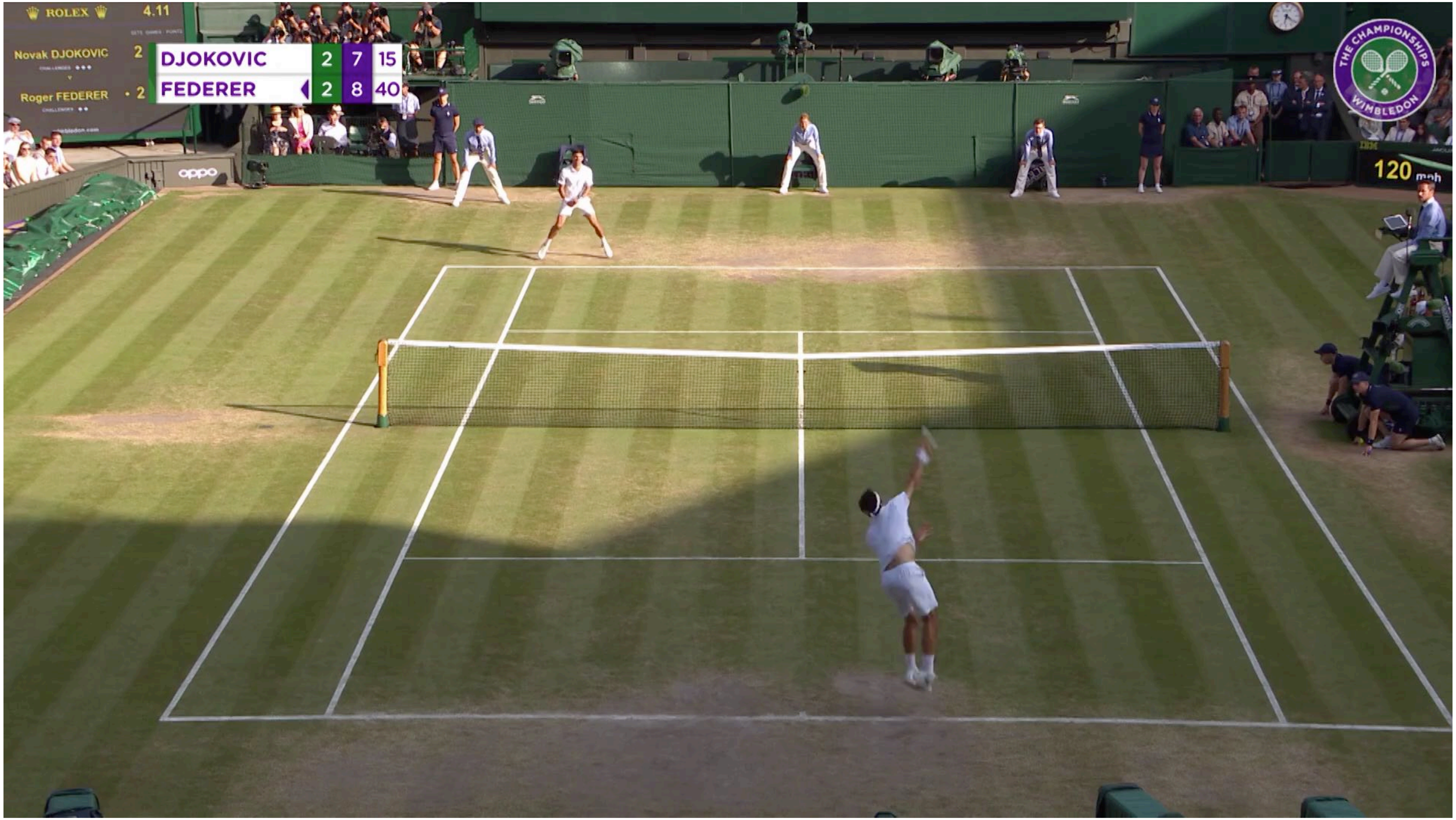
Example:

- In a recent project, we asked the question… given enough video of tennis matches of a professional athlete, could we come up with an algorithm for turning all this input video into a controllable video game character?

- Words are ambiguous. Consider possible listener questions?
  - What does this video contain? Is it easy to analyze?
  - What do you mean by controllable characters?
  - What output are you trying to create? Video? 3D animation?

Compare the description above to the following sequence…

Here's an example of that source video

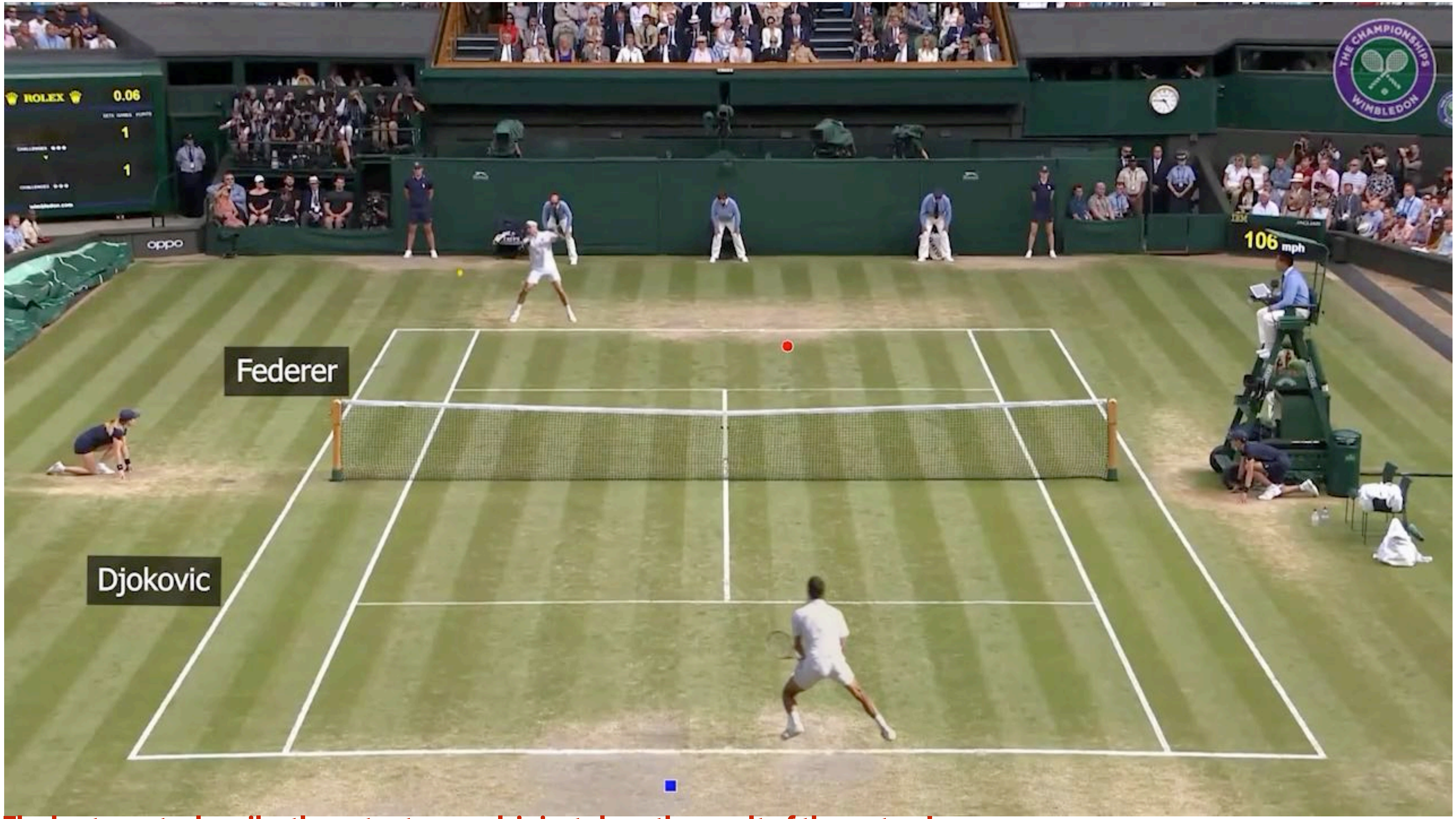


The best way to describe the input data is just show it! ("This is what the input looks like!")

And there's a lot of it out there!

Showing a lot of thumbnails is a visual metaphor for "a lot" .

And here's an example of controllable output



The best way to describe the output we seek is just show the result of the system!
("The user clicks to specify a target ball location, and the player hits the incoming ball back to the red dot")

Another example:

- We had a problem in this project: input videos were taken in different lighting conditions, and these lighting differences were the cause of bad results.

- An anticipated audience question: “what do you mean by lighting differences?”

The problem (lighting differences)

After the fix

Another example: we recently created a game engine that simultaneously executes many games at the same time

Tip 5

The audience prefers not to think (much)

An audience has a finite supply of mental effort

- **The audience <u>does not</u> want to burn mental effort about things you know and can just tell them.**
  - They want to be led by hand through the major steps of your story
  - They <u>do not</u> want to interpret any of your figures or graphs, they want to be directly told how to interpret them (e.g., what to look for in a graph).
  - They want to be told about your key assumptions

- **The audience <u>does</u> want to spend their energy thinking about:**
  - Potential problems/limitations with what you did (Did you consider all edge cases? Is your evaluation sound?)
  - Implications of your approach to their work
  - Connections to their own work

Most importantly:

The audience does not want to think about “why” you are telling them something

Which leads me to…

Tip 6

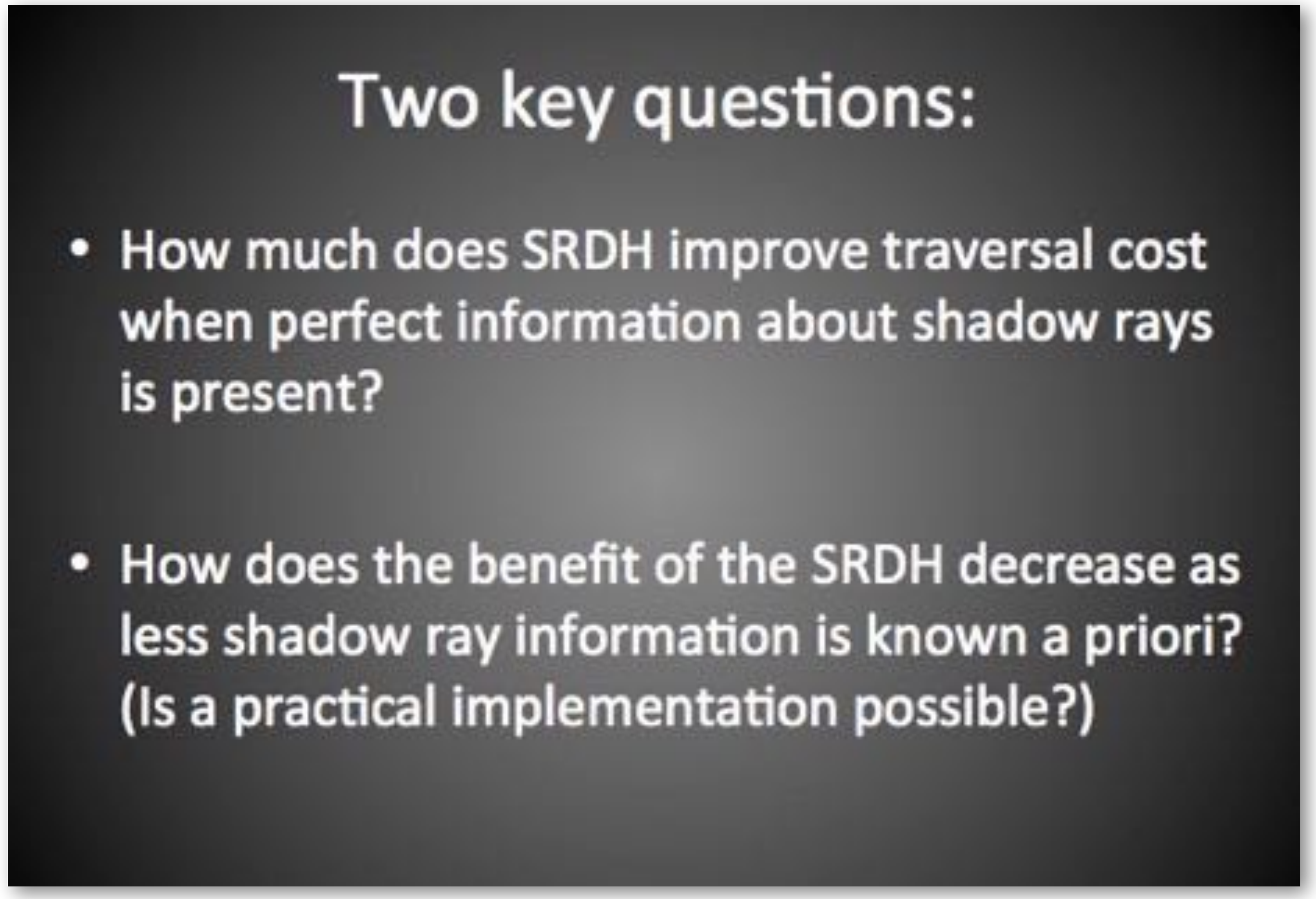
Surprises* are almost always bad:

Always say where you are going and why you must go there before you say what you did.

* I am referring to surprises in talk narrative and/or exposition. A surprising result is great.

Give the <u>why</u> before the <u>what</u>

- **In the algorithm description:**
  - "We need to first establish some terminology"
  - "Even given X, the problem we still haven't solved is..."
  - "Now that we have defined a cost metric we need a method to minimize it..."



Two key questions:

- How much does SRDH improve traversal cost when perfect information about shadow rays is present?
- How does the benefit of the SRDH decrease as less shadow ray information is known a priori? (Is a practical implementation possible?)

- **In the results/evaluation:**
  - Speaker: "Key questions to ask about our approach are..."
  - Audience: "Thanks! I agree, those are good questions. Let's see what the results say!"

- **"The why" provides the listener context for...**
  - Compartmentalizing: assessing how hard they should pay attention (is this a critical idea, or just an implementation detail?). Especially useful if they are getting lost.
  - Understanding how parts of the talk relate ("Why is the speaker now introducing a new optimization framework?")

Big surprises in a narrative are a bad sign

- **Ideally, you want the audience to always be able to <u>anticipate</u>* what you are about to say**
  - **This means: your story is so clear it's obvious!**
  - **It also means the talk is really easy to present without notes or text on slides (it just flows)**

- **If you are practicing your talk, and you keep forgetting what's coming on the next slide (that is, you can't anticipate it)...**
  - **This means: you probably need to restructure your talk because a clear narrative is not there.**
  - **It's not even obvious to you! Ouch!**

* Credit to Abhinav Gupta for suggesting the term anticipation, and for the example on this slide

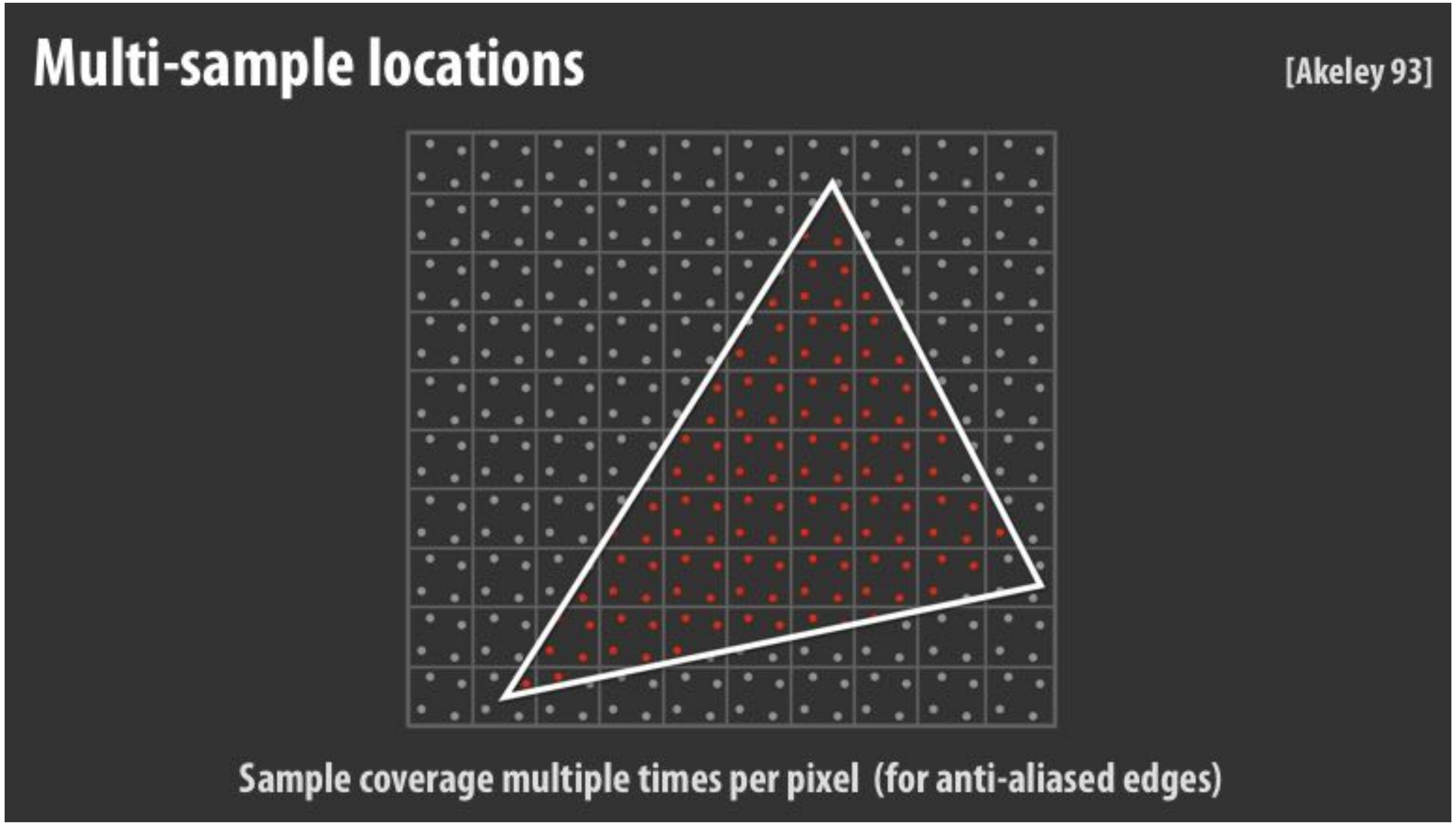
Tip 7

Always, always, always explain any figure or graph

(remember, the audience does not want to think about things you can tell them)

Explain every figure

- Explain every visual element in the figure (never make the audience decode a figure)
- Refer to highlight colors explicitly (explain why the visual element is highlighted)



Multi-sample locations

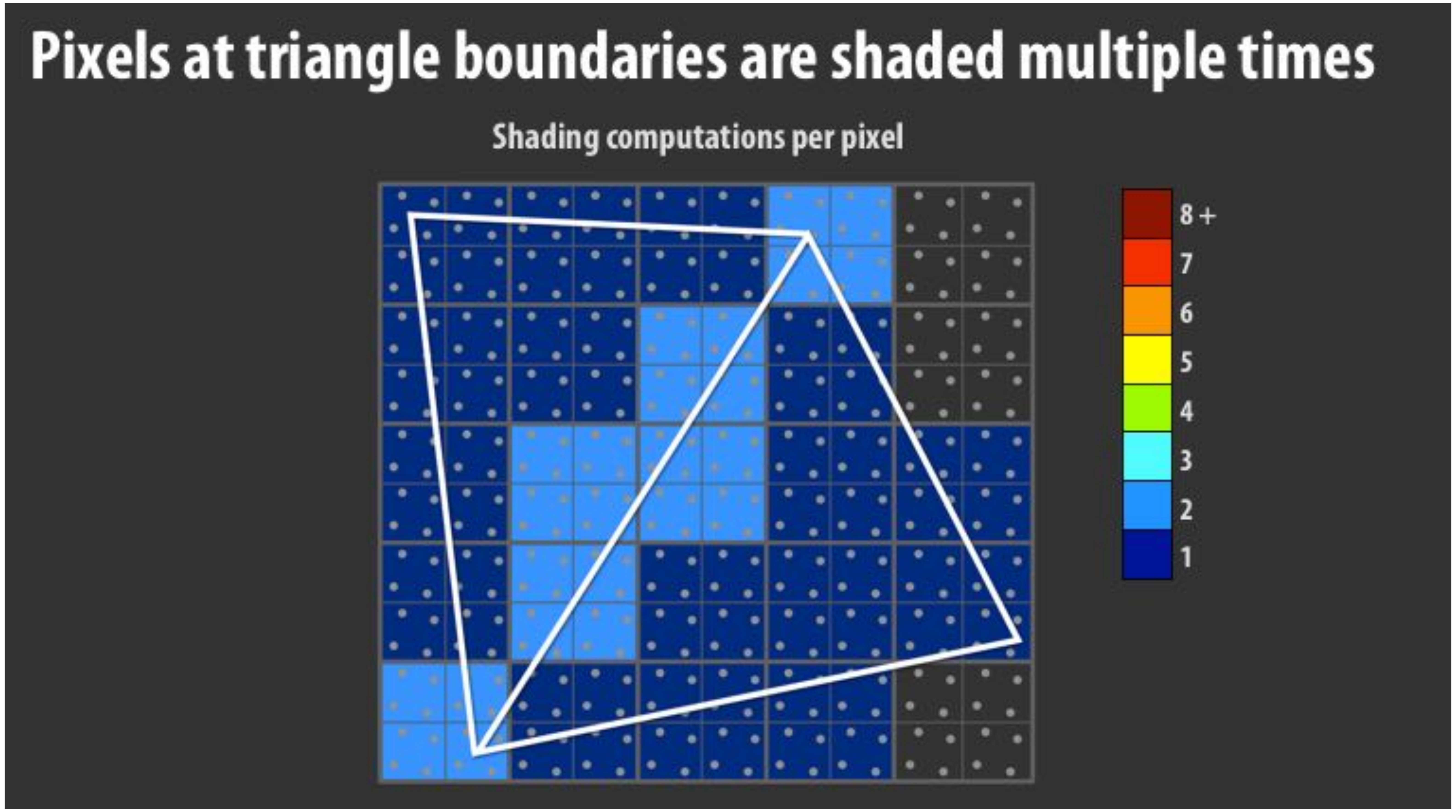
[Akeley 93]

Sample coverage multiple times per pixel (for anti-aliased edges)

Example voice over: "Here I'm showing you a pixel grid, a projected triangle, and the location of four sample points at each pixel. Sample points falling within the triangle are colored red."

Explain every figure

- Lead the listener through the key points of the figure
- A very useful phrase: “As you can see...”
    - It’s like verbal eye contact. It keeps the listener engaged and makes the listener happy... “Oh yeah, I can see that! I am following this talk!”



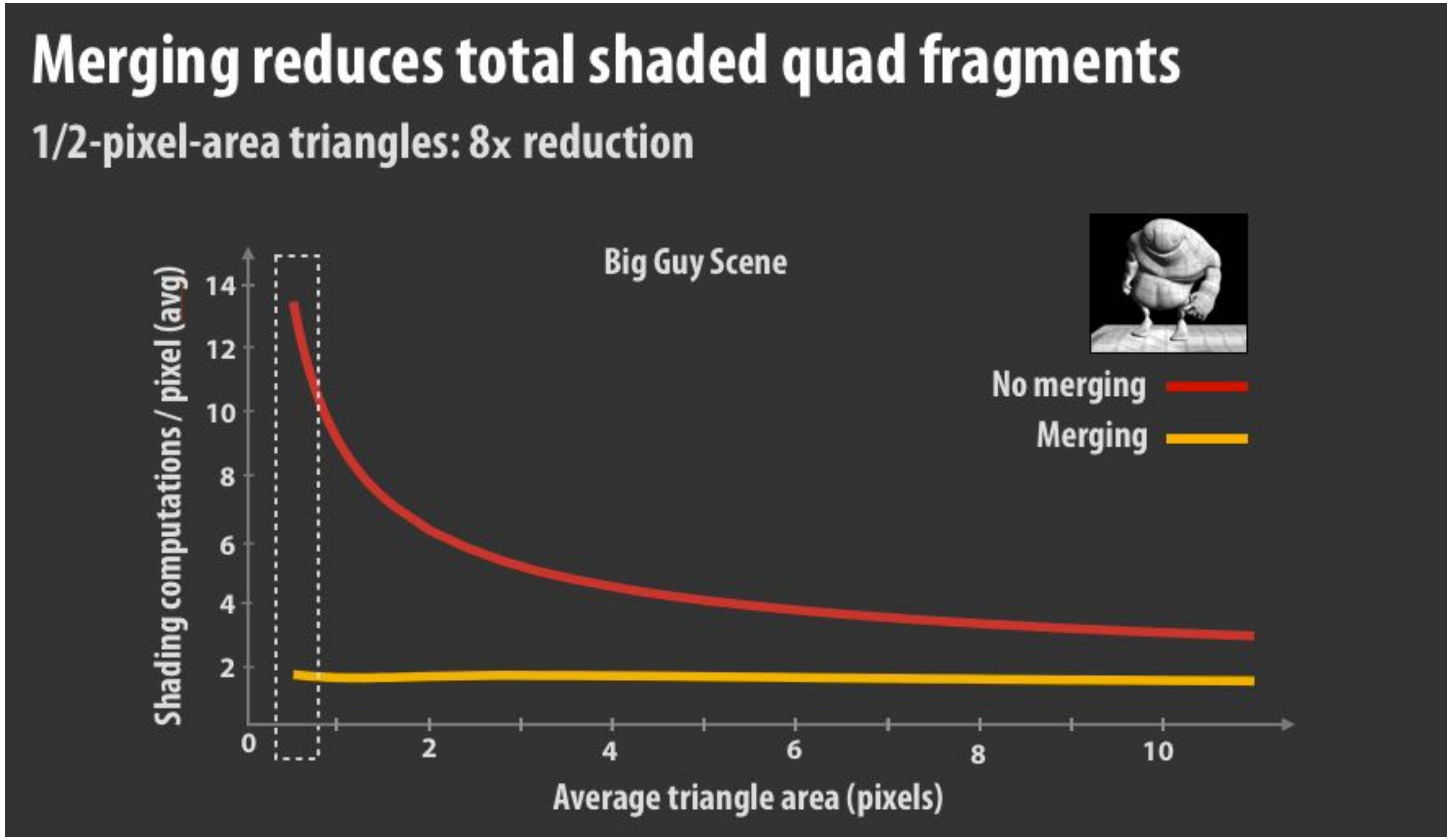
Pixels at triangle boundaries are shaded multiple times

Shading computations per pixel

Example voice over: “Now I’m showing you two adjacent triangles, and I’m coloring pixels according to the number of shading computations that occur at each pixel as a result of rendering these two triangles. As you can see from the light blue region, pixels near the boundary of the two triangles get shaded twice.

Explain every results graph

- May start with a general intro of what the graph will address ("anticipate" the result)
- Then describe the axes (and your axes better have labels!)
- Then describe the <u>one point</u> that you wish to make with this results slide (more on this later!)



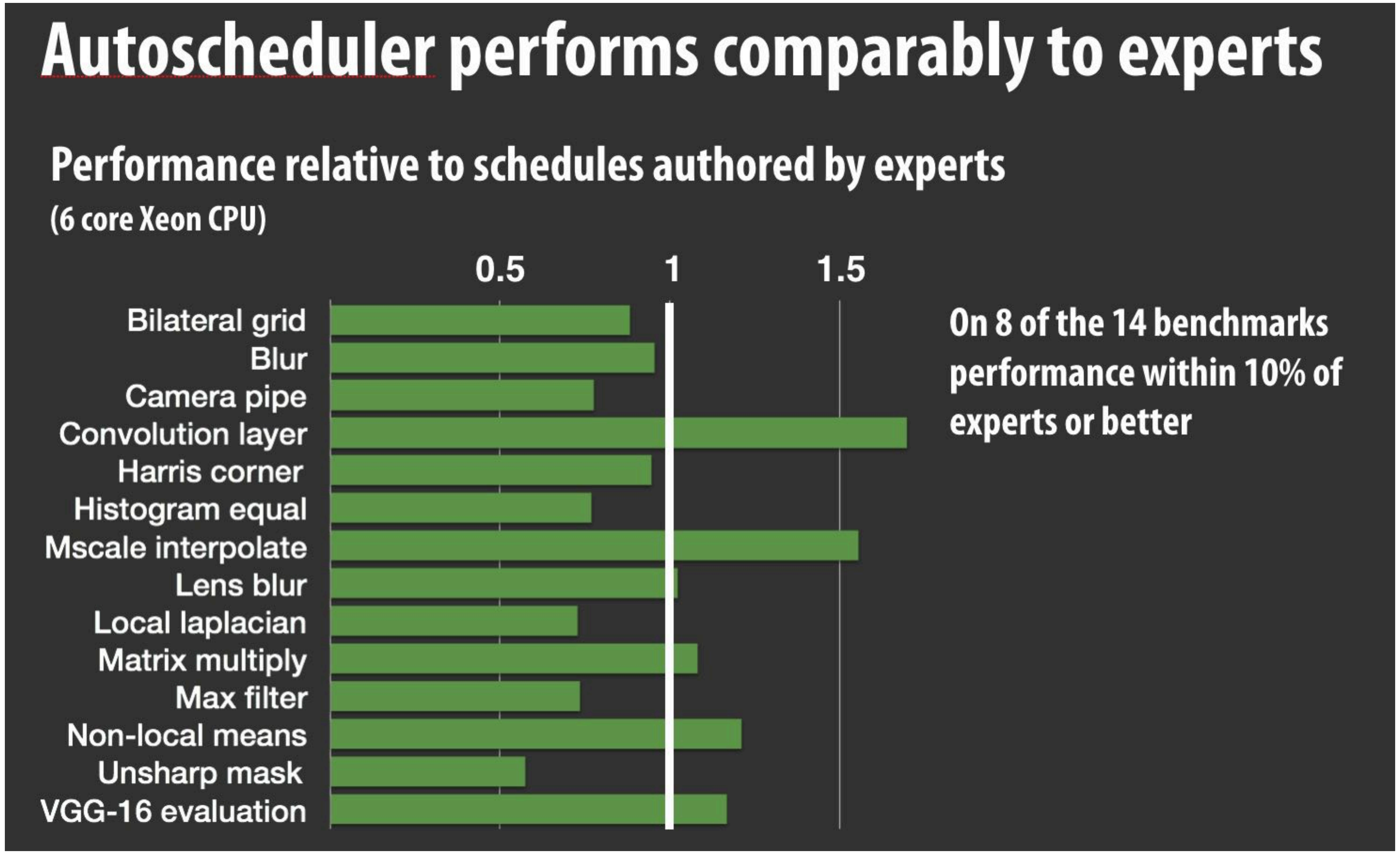
Merging reduces total shaded quad fragments

1/2-pixel-area triangles: 8x reduction

Example voice over: "Our first questions were about performance: how much did the algorithm reduce the number of the shading computations? And we found out that the answer is a lot. This figure plots the number of shading computations per pixel when rendering different tessellations of the big guy scene. X-axis gives triangle size. If you look at the left side of the graph, which corresponds to a high-resolution micropolygon mesh, <u>you can see</u> that merging, shown by yellow line, shades over eight times less than the convention pipeline.

Explain every results graph

- May start with a general intro of what the graph will address.
- Then describe the axes (your axes better have labels!)
- Then describe the <u>one point</u> that you wish to make with this results slide (more on this later!)



Autoscheduler performs comparably to experts

Performance relative to schedules authored by experts
(6 core Xeon CPU)

On 8 of the 14 benchmarks performance within 10% of experts or better

Example voice over: "Our first question was about performance: how fast is the auto scheduler compared to experts? And we found out that it's quite good. This figure plots the performance of the autoscheduler compared to that of expert code. So expert code is 1. Faster code is to the right. As you can see, the auto scheduler is within 10% of the performance of the experts in many cases, and always within a factor of 2.

Tip 8

In the results section:

One point per slide!

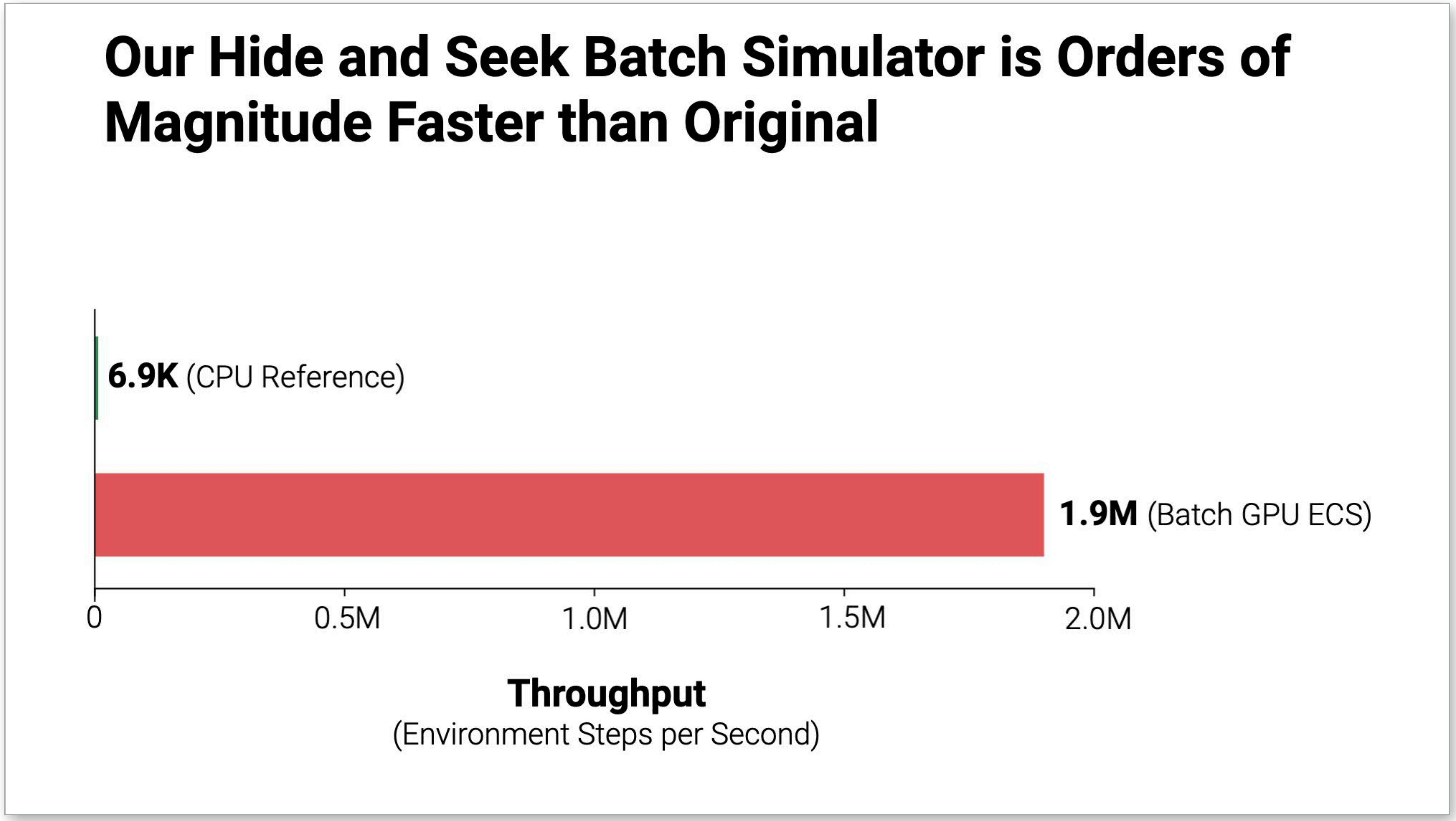
One point per slide!

One point per slide!

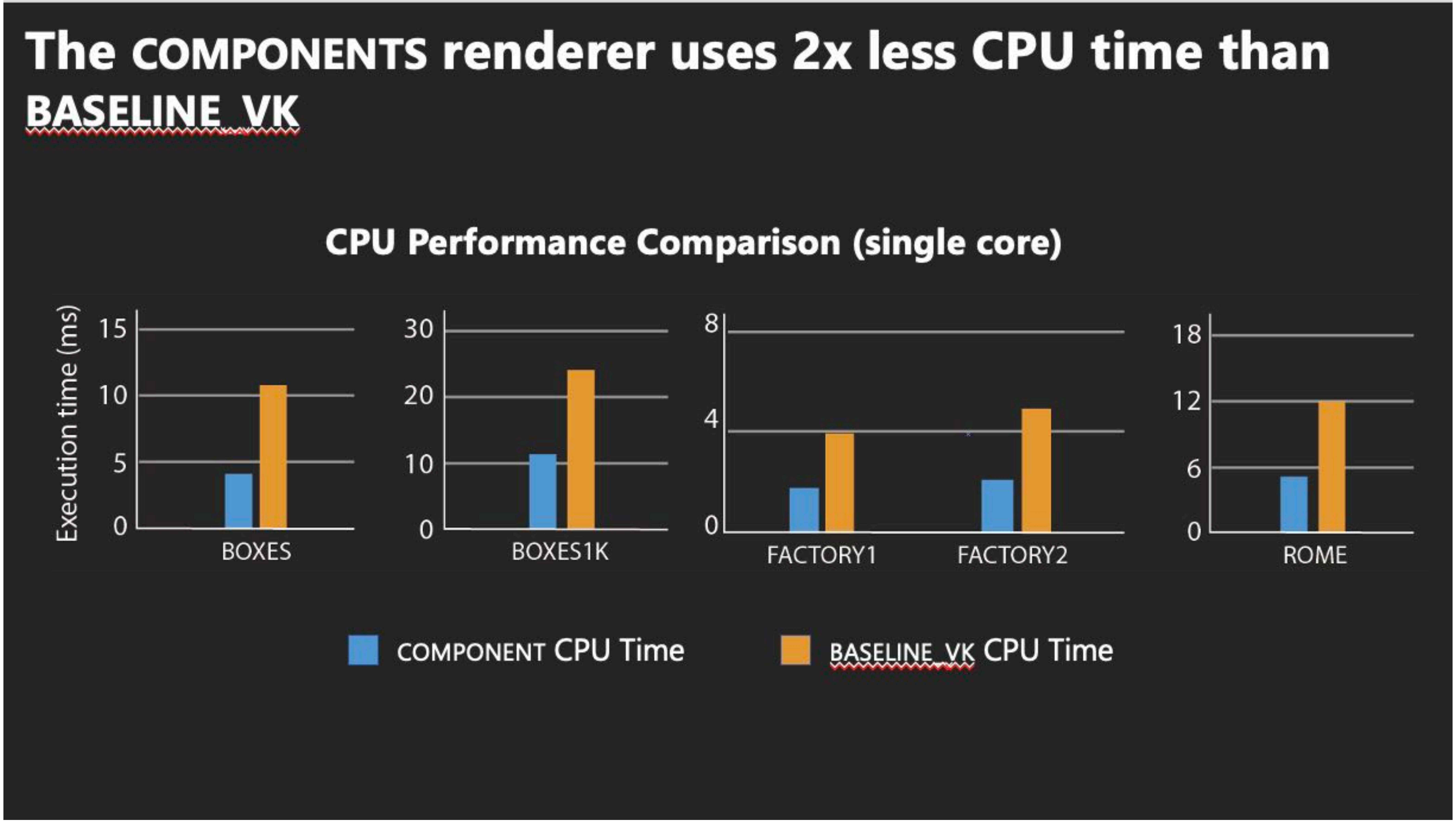
(and the point is the title of the slide!!!)

Make the point of the graph the slide's title:

- It provides audience context for interpreting the graph
  - Audience reaction: "Let me see if I can verify that point in the graph to check my understanding"
- This is another example of the "audience prefers not to think" principle



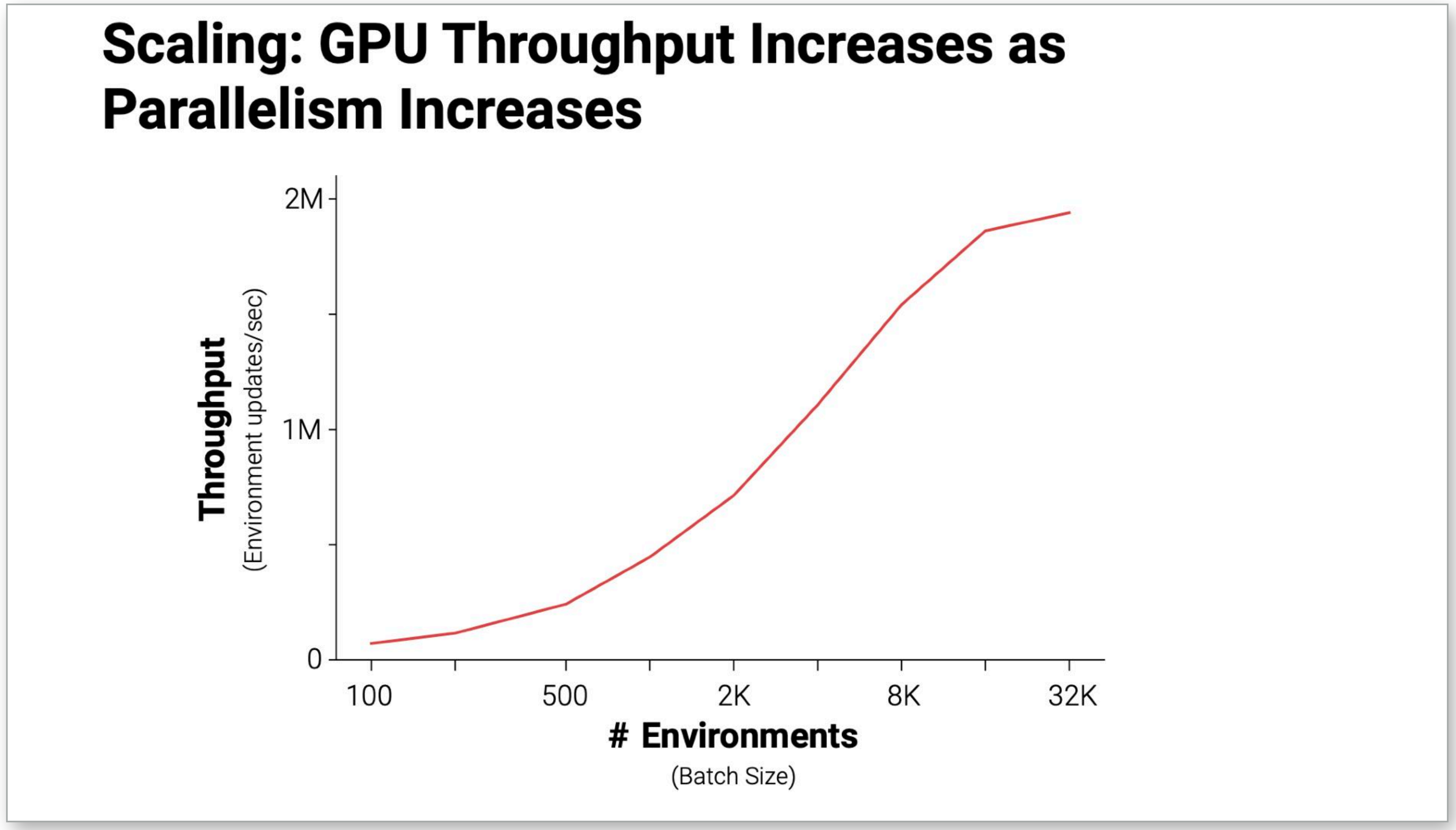
Another example: point is the title of the slide



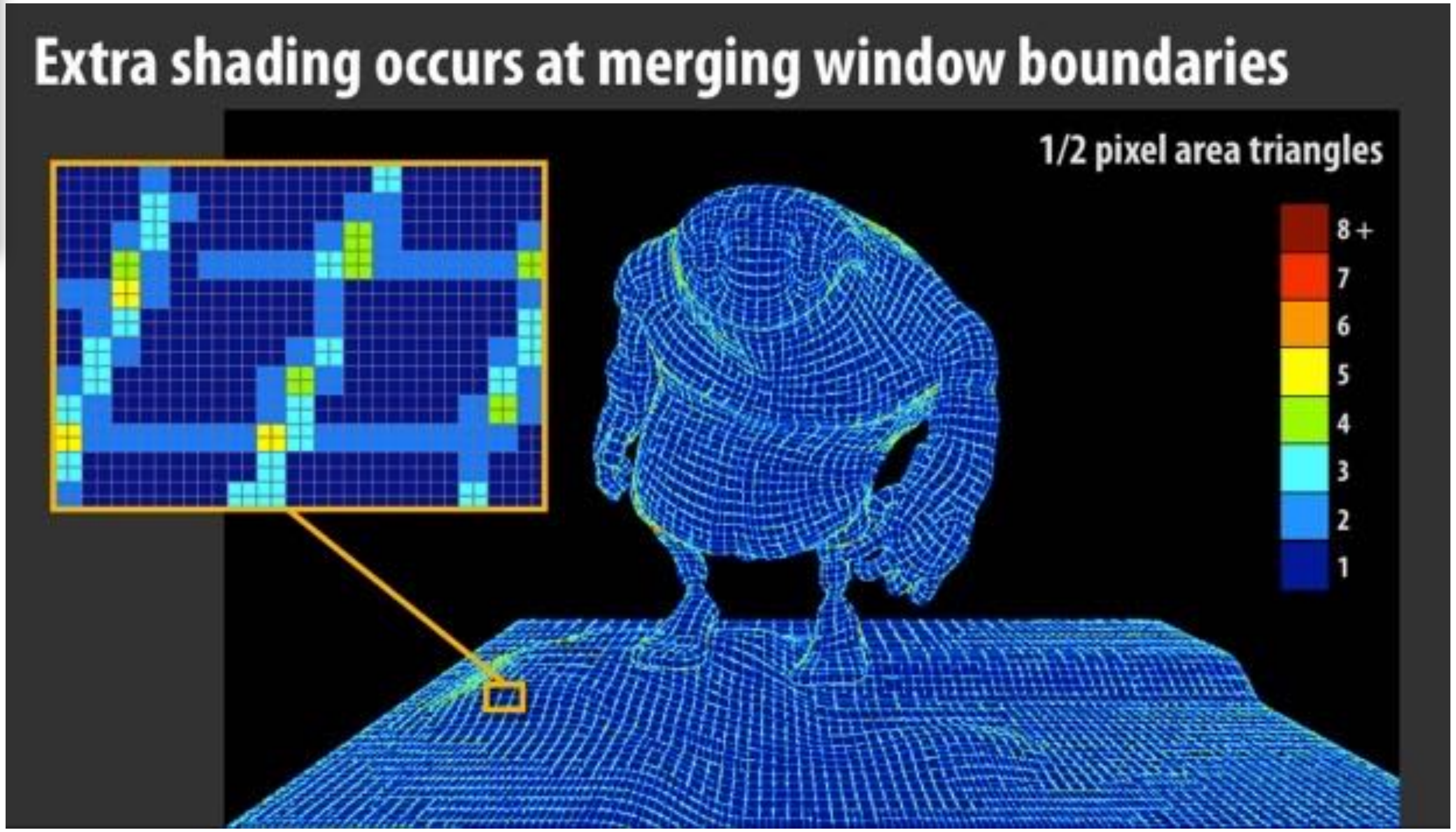
The COMPONENTS renderer uses 2x less CPU time than BASELINE_VK

CPU Performance Comparison (single core)

More examples: point is the title of the slide



Scaling: GPU Throughput Increases as Parallelism Increases



Extra shading occurs at merging window boundaries

Corollary to the one-point-per-slide rule

- In general, you don’t want to show data on a results slide that is unrelated to the point of the slide

- This usually means you need to remake the results graphs from your paper for the purpose of the talk (it’s a pain, but sorry, it’s important) *

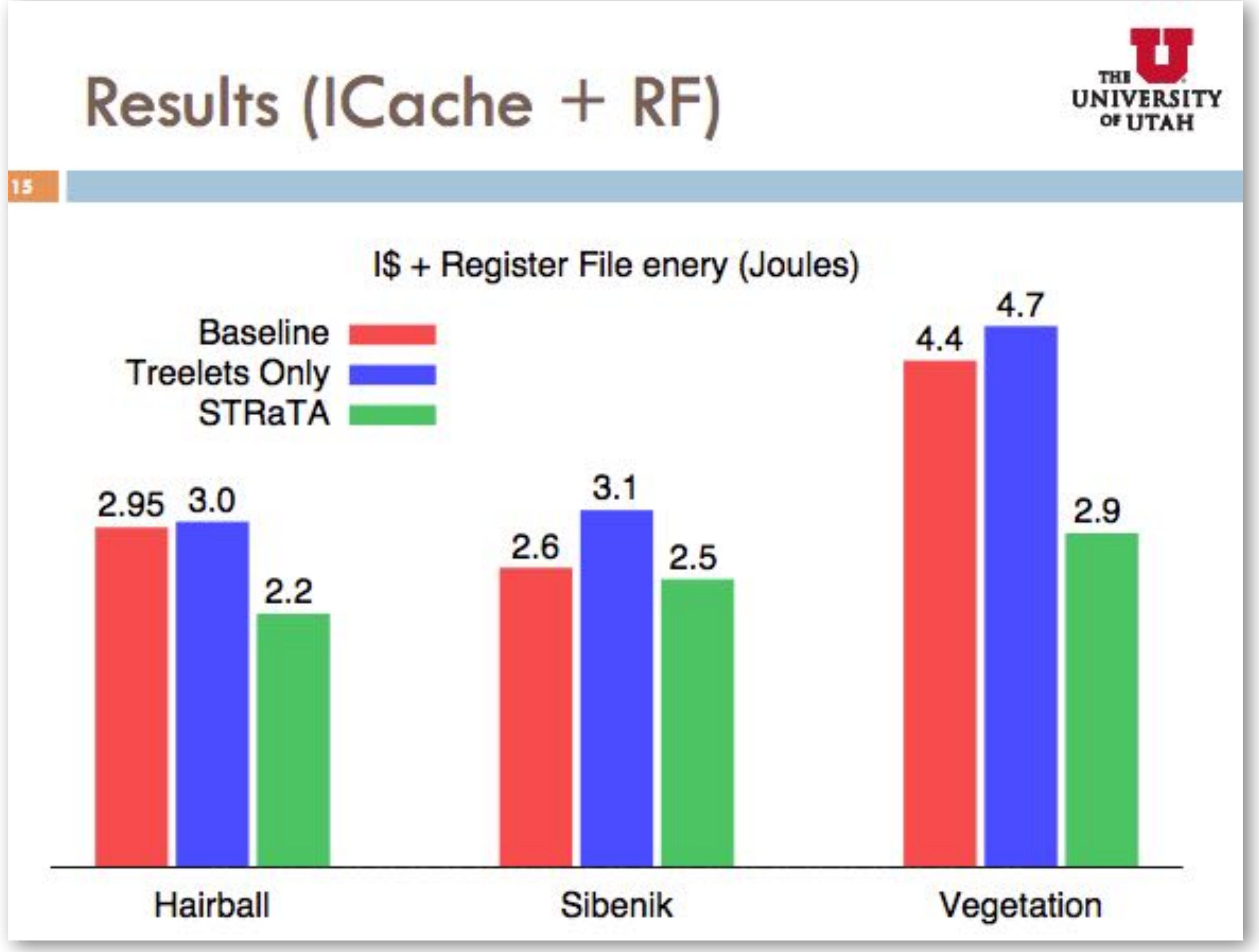
- This is the “every sentence matters” principle applied to visual details on a slide

* This is an example of a tip for conference talk polish: not for informal talks

Suboptimal examples of results slides

- Notice how you (as an audience member) are working hard to interpret the trends in these graphs
  - You are asking: what do these results say?
  - What am I supposed to be concluding?

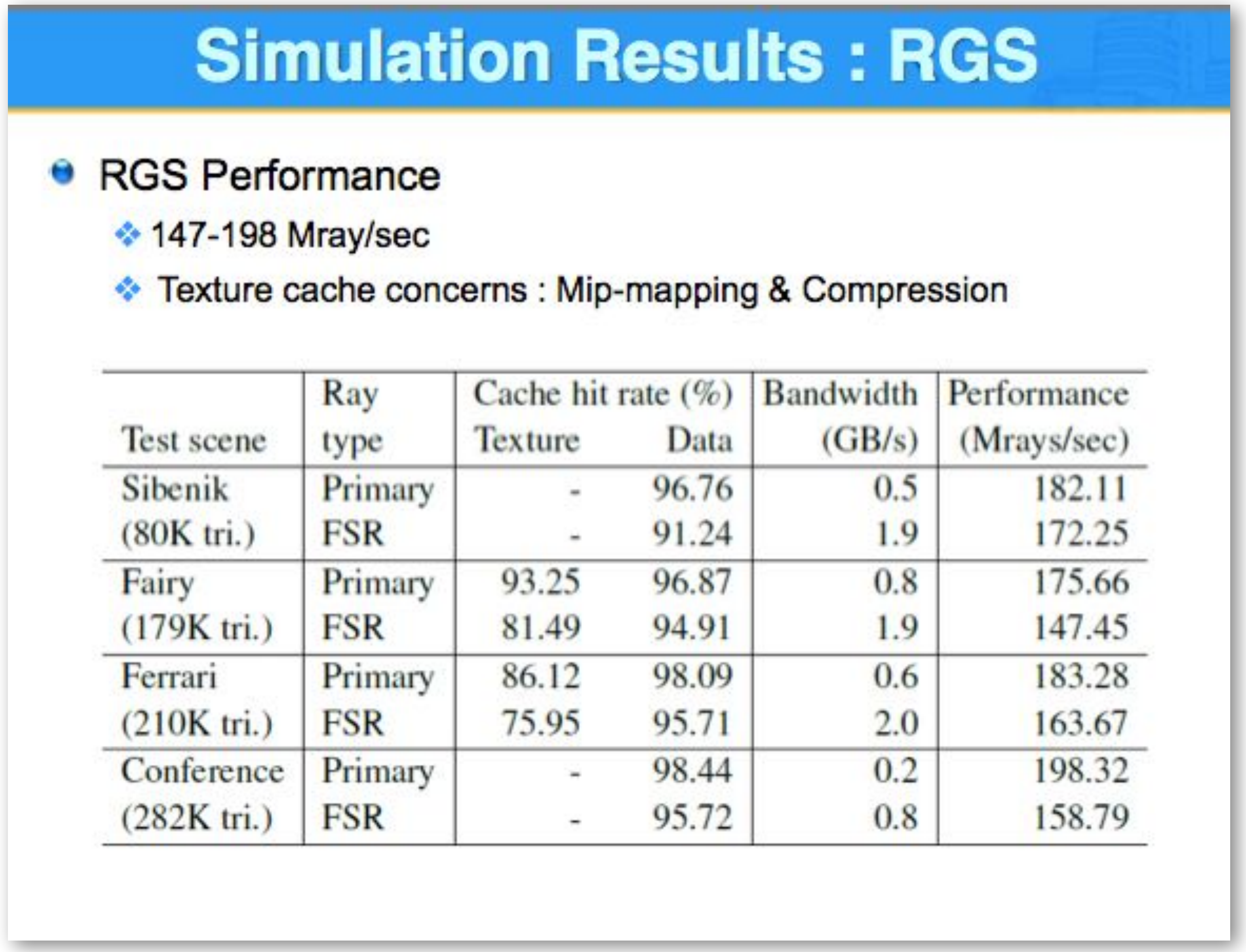
- The audience just wants to be told what to look for!
  - They are reading the graphs to verify the main point, not determine the main point.



Results (ICache + RF)

THE UNIVERSITY OF UTAH

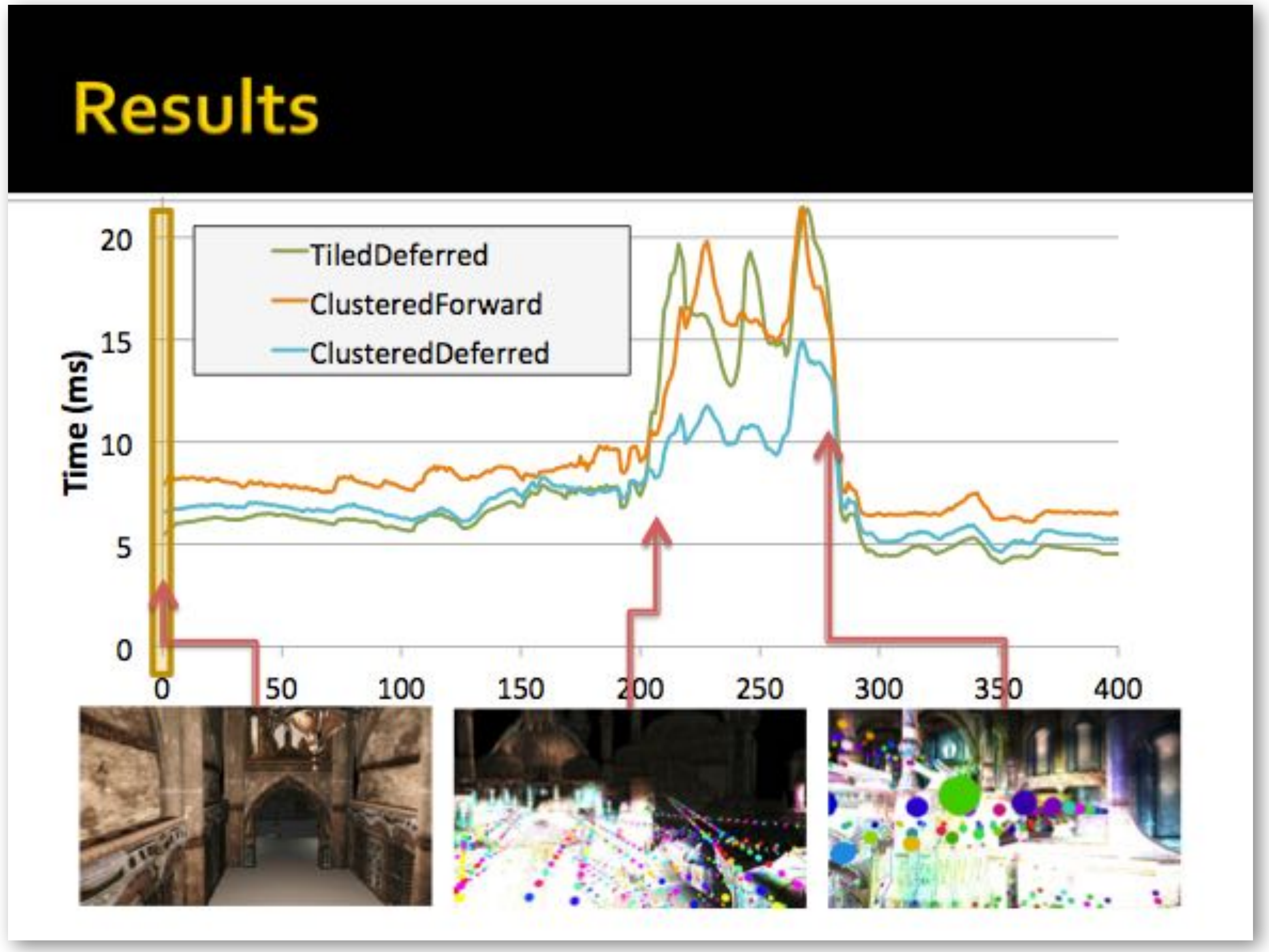
15



Simulation Results : RGS

- RGS Performance
  - 147-198 Mray/sec
  - Texture cache concerns : Mip-mapping & Compression

Test scene	Ray type	Cache hit rate (%) Texture	Data	Bandwidth (GB/s)	Performance (Mrays/sec)
Sibenik (80K tri.)	Primary	-	96.76	0.5	182.11
	FSR	-	91.24	1.9	172.25
Fairy (179K tri.)	Primary	93.25	96.87	0.8	175.66
	FSR	81.49	94.91	1.9	147.45
Ferrari (210K tri.)	Primary	86.12	98.09	0.6	183.28
	FSR	75.95	95.71	2.0	163.67
Conference (282K tri.)	Primary	-	98.44	0.2	198.32
	FSR	-	95.72	0.8	158.79



Results

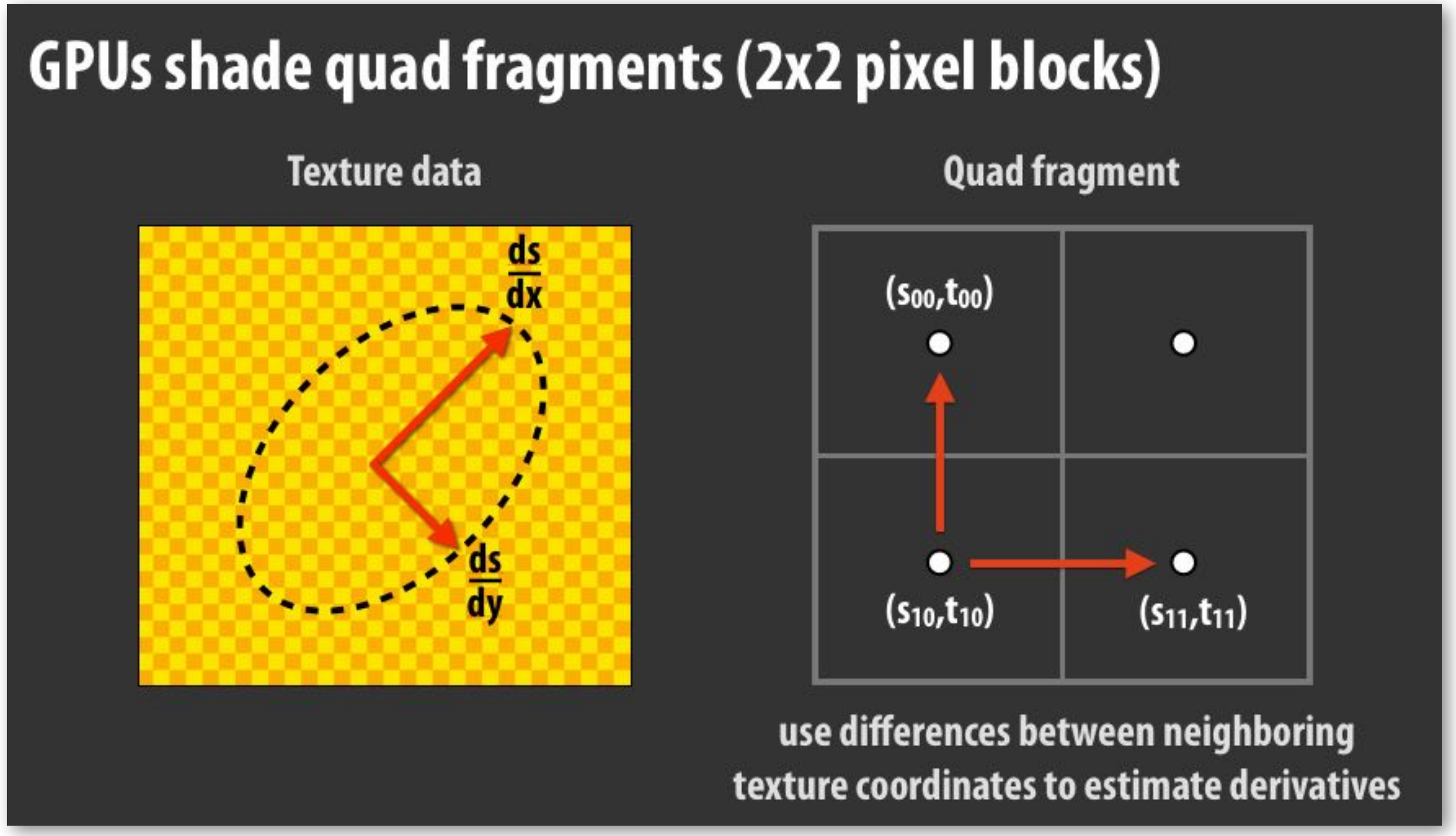
Tip 9

Side titles matter

If you read the titles of your talk all the way through, it should be a great summary of the talk.

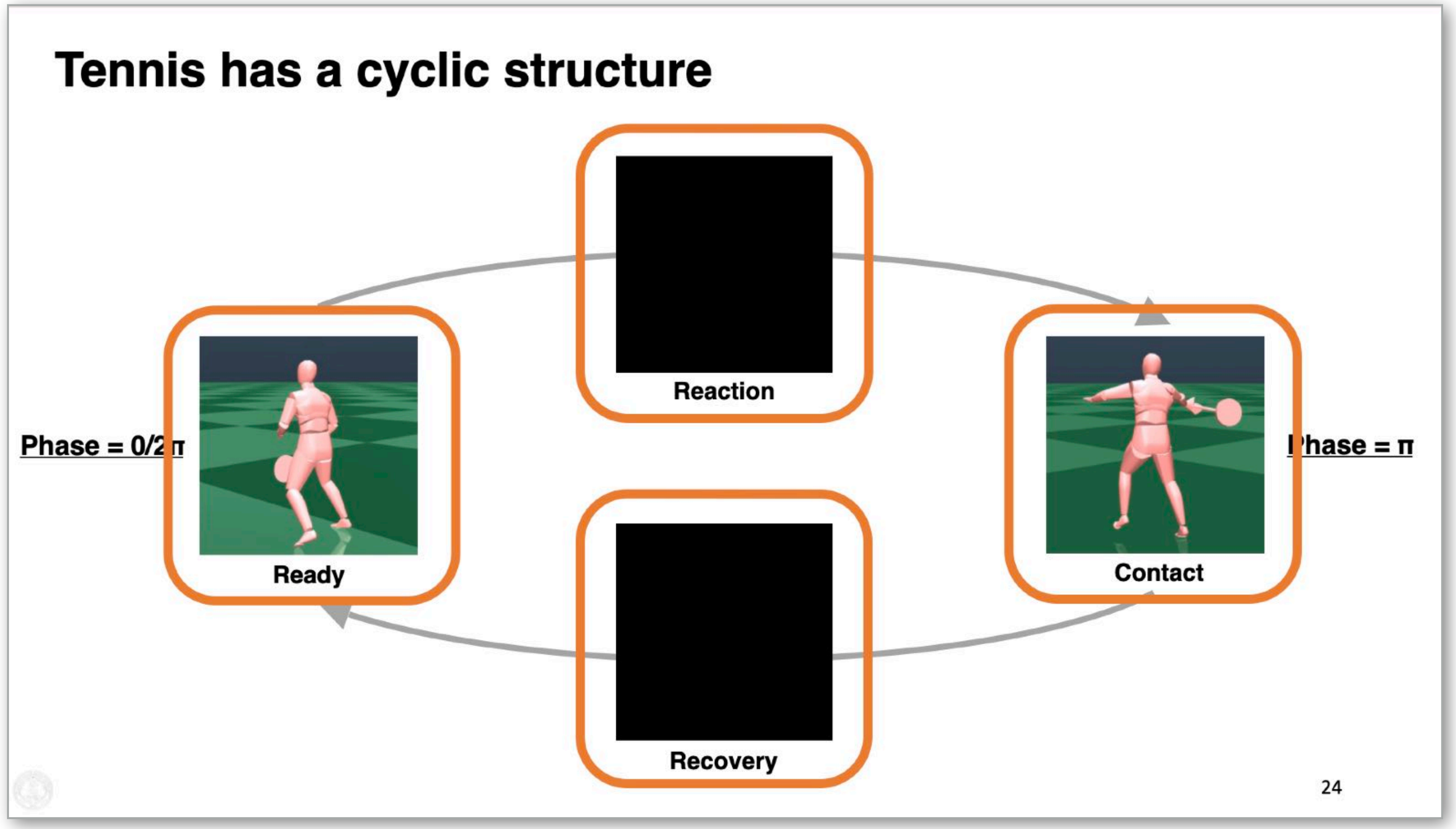
(basically, this is “one-point-per-slide” for the whole talk)

Examples of good slide titles

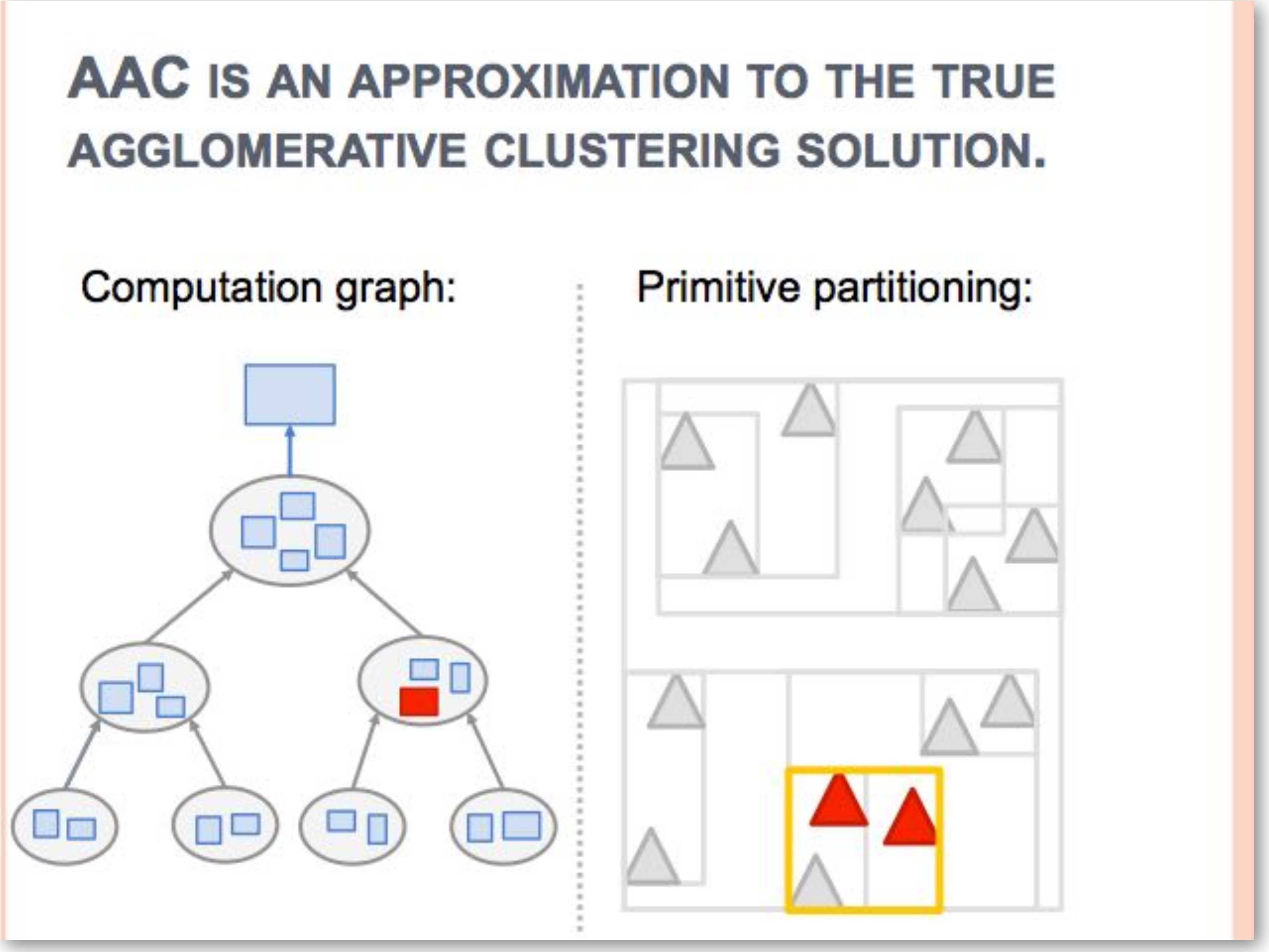


GPUs shade quad fragments (2x2 pixel blocks)

use differences between neighboring texture coordinates to estimate derivatives



Tennis has a cyclic structure



AAC IS AN APPROXIMATION TO THE TRUE AGGLOMERATIVE CLUSTERING SOLUTION.

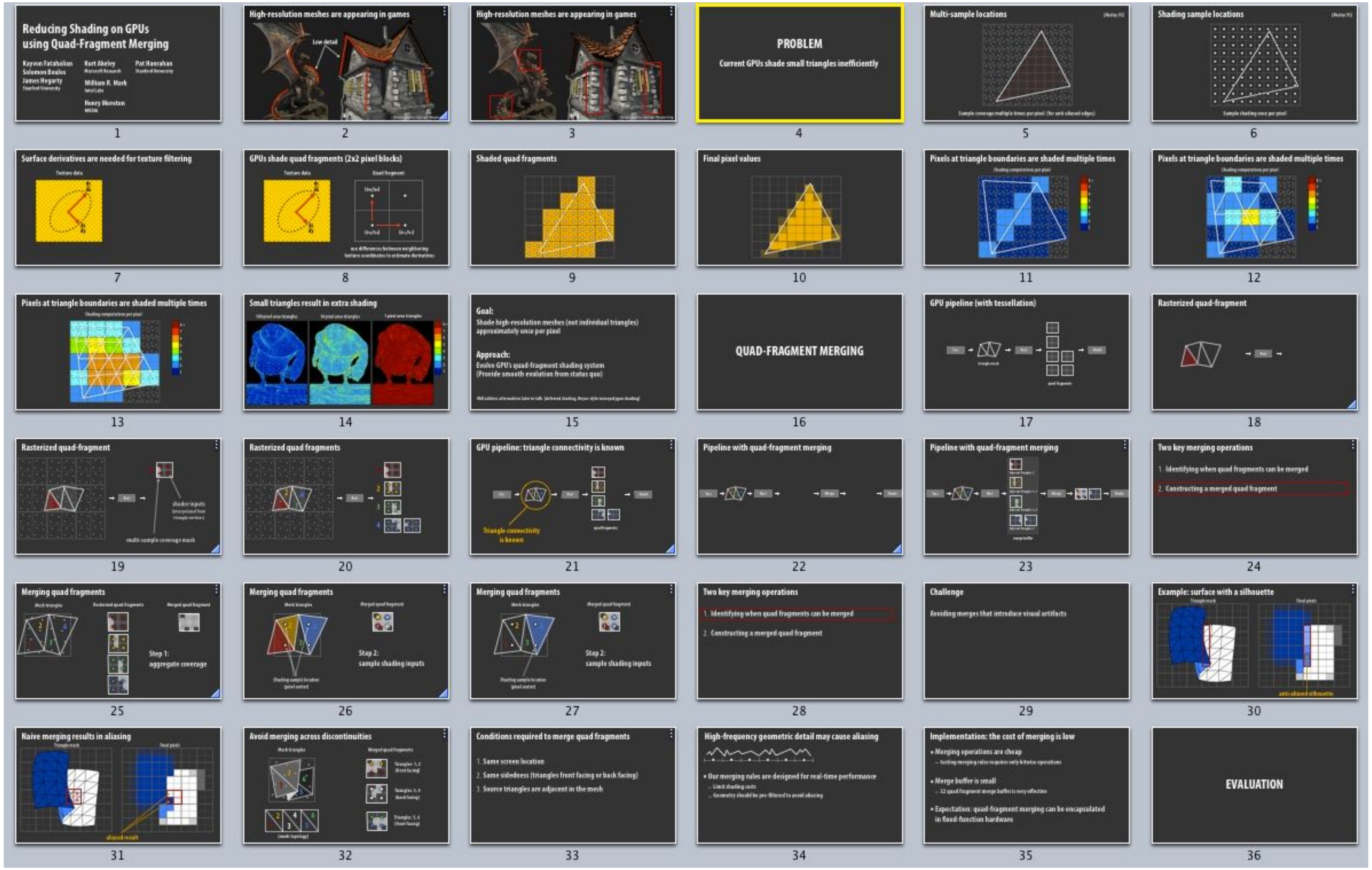
The reason for meaningful slide titles is convenience and clarity for the audience

“Why is the speaker telling me this again?”

(Recall “why before what”)

Read your slide titles in thumbnail view

Do they make all the points of the story you are trying to tell?



1 Reducing Shading on GPUs using Quad-Fragment Merging
Kayvon Fatahalian, Solomon Boulos, James Hegarty — Stanford University; Kurt Akeley — Microsoft Research; William R. Mark — Intel Labs; Henry Moreton — NVIDIA; Pat Hanrahan — Stanford University

2 High-resolution meshes are appearing in games

3 High-resolution meshes are appearing in games

4 PROBLEM
Current GPUs shade small triangles inefficiently

5 Multi-sample locations

6 Shading sample locations

7 Surface derivatives are needed for texture filtering

8 GPUs shade quad fragments (2x2 pixel blocks)

9 Shaded quad fragments

10 Final pixel values

11 Pixels at triangle boundaries are shaded multiple times

12 Pixels at triangle boundaries are shaded multiple times

13 Pixels at triangle boundaries are shaded multiple times

14 Small triangles result in extra shading

15 Goal:
Shade high-resolution meshes (not individual triangles) approximately once per pixel
Approach:
Evolve GPU's quad-fragment shading system
(Provide smooth evolution from status quo)

16 QUAD-FRAGMENT MERGING

17 GPU pipeline (with tessellation)

18 Rasterized quad-fragment

19 Rasterized quad-fragment

20 Rasterized quad fragments

21 GPU pipeline: triangle connectivity is known

22 Pipeline with quad-fragment merging

23 Pipeline with quad-fragment merging

24 Two key merging operations
1. Identifying when quad fragments can be merged
2. Constructing a merged quad fragment

25 Merging quad fragments
Step 1: aggregate coverage

26 Merging quad fragments
Step 2: sample shading inputs

27 Merging quad fragments
Step 2: sample shading inputs

28 Two key merging operations
1. Identifying when quad fragments can be merged
2. Constructing a merged quad fragment

29 Challenge
Avoiding merges that introduce visual artifacts

30 Example: surface with a silhouette

31 Naive merging results in aliasing

32 Avoid merging across discontinuities

33 Conditions required to merge quad fragments
1. Same screen location
2. Same sidedness (triangles front facing or back facing)
3. Source triangles are adjacent in the mesh

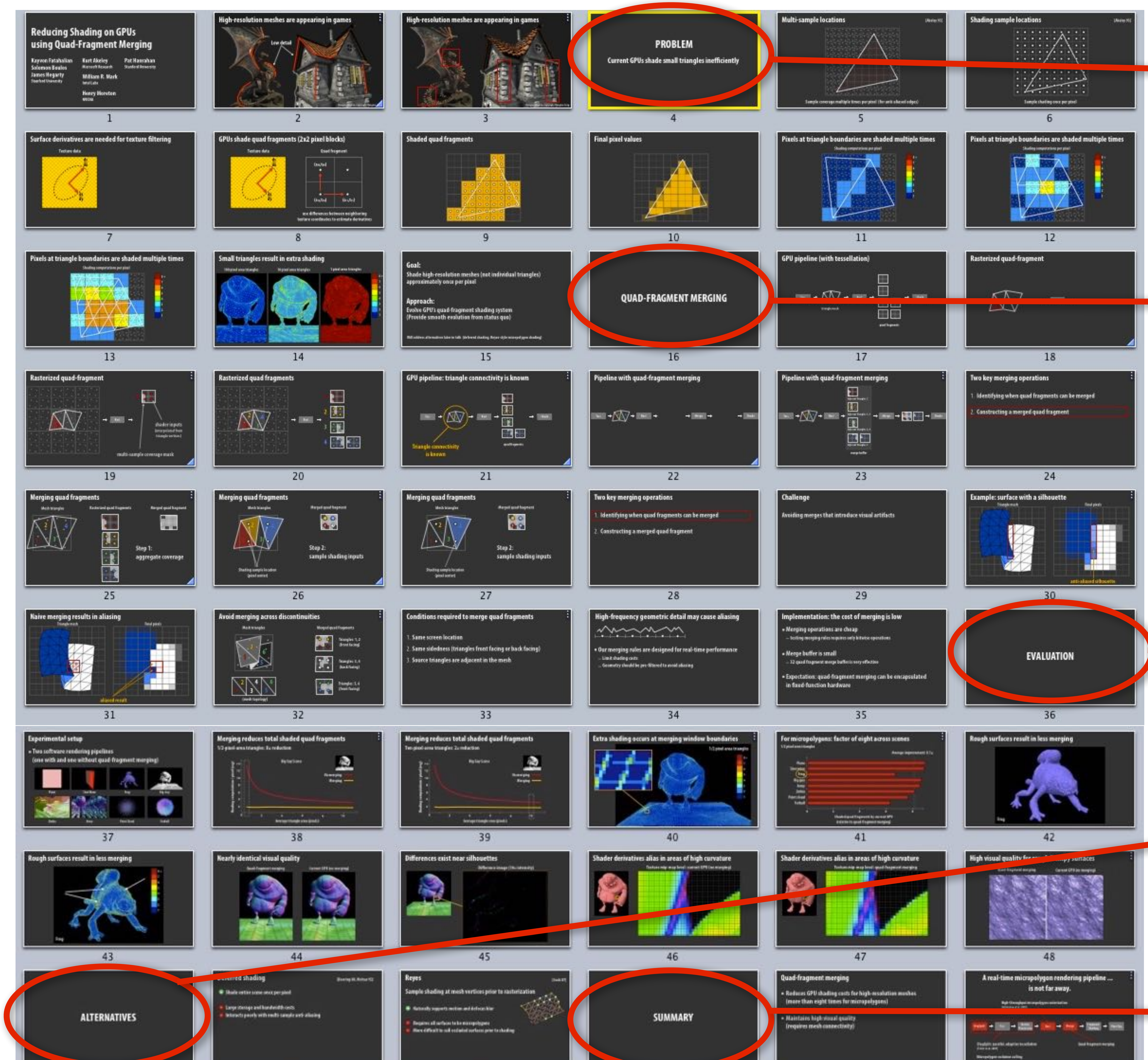
34 High-frequency geometric detail may cause aliasing
- Our merging rules are designed for real-time performance

35 Implementation: the cost of merging is low
- Merging operations are cheap
- Merge buffer is small
- Expectation: quad-fragment merging can be encapsulated in fixed-function hardware

36 EVALUATION

Tip 10

Organize your talk with section slides


Reducing Shading on GPUs using Quad-Fragment Merging
PROBLEM
Current GPUs shade small triangles inefficiently
QUAD-FRAGMENT MERGING
EVALUATION
ALTERNATIVES
SUMMARY

I'm about to frame the problem in my terms

Ok, let's talk about our solution

Let's take a look at how well it works

Of course there are some potential alternatives that we didn't go with

Let me wrap up

Stage your talk with section slides

- **Useful for your audience**
  - It provides guidance for <u>what</u> you hope to achieve next (no surprises!)
  - Compartmentalization: it's absolutely clear where the shifts are
  - If a listener got lost, it's a good place for them to re-engage
  - It's a place for the audience to take a breath
  - It gives the talk a more colloquial tone

- **Useful for you**
  - It's a chance for you to pause and take a breath
  - It's a great breakdown of the talk for practicing subsections

Tip 11

End on a positive note!

End on a positive note!

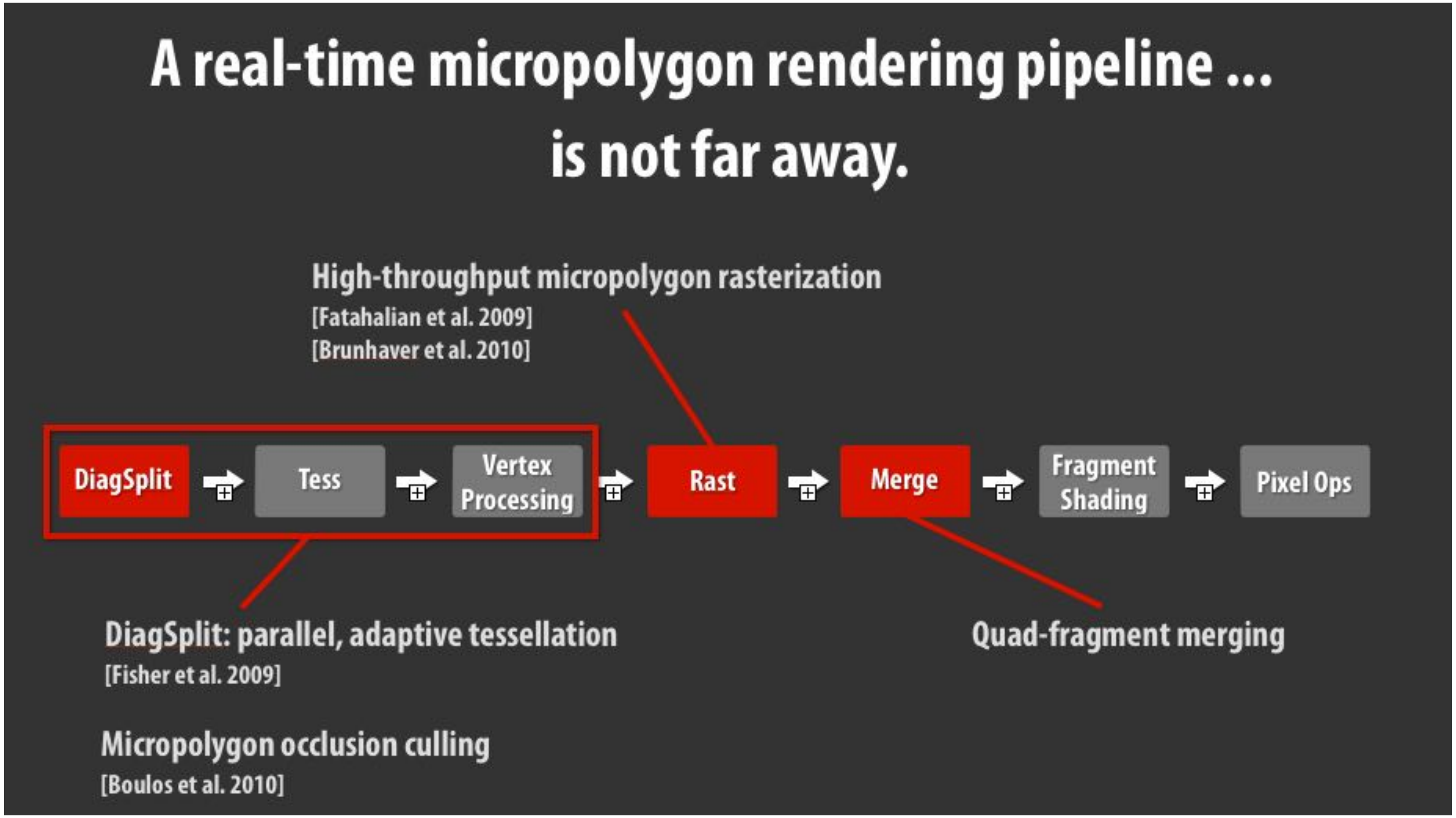
The future is bright!
Lots of new work to do, here are some ideas!
This work is one part of something much bigger!

Many talks end on future work in a manner that stresses problems with the current work or enumerates obvious next steps ("we will apply this work to X", "generalize method to Y", etc.)

- It's boring, and the audience wants to think about the bigger picture
  (Audience: "Is that the speaker's away from the whole project?" "What am I supposed to conclude here?")
- It's a lost opportunity to impart critical intellectual thought to the field *
  - Recall: introduction was where you contributed critical intellectual thought in how to think about the problem being solved today.
  - Conclusion is where you contribute intellectual thought that reflects on what you have done, or what is means for the future

* This is "every sentence matters" on the conclusion

Final SIGGRAPH slide as a Ph.D. student



A real-time micropolygon rendering pipeline ... is not far away.

High-throughput micropolygon rasterization
[Fatahalian et al. 2009]
[Brunhaver et al. 2010]

DiagSplit → Tess → Vertex Processing → Rast → Merge → Fragment Shading → Pixel Ops

DiagSplit: parallel, adaptive tessellation
[Fisher et al. 2009]

Quad-fragment merging

Micropolygon occlusion culling
[Boulos et al. 2010]

An earlier draft had a very simple future work slide ("we could do X, Y, Z"). I was told by my advisor that it was a let down and to think about how to end on a broader note. This one slide took at least half a day to come up with on its own.

Result: Tony DeRose at PIXAR got it! He realized the point wasn't just the one particular optimization that was the contribution of the SIGGRAPH paper, but a broader line of work on rethinking the graphics pipeline for high-quality rendering: his comment was of the effect, "I'm glad someone's finally figured the big pieces of this out."

Tip 12

The audience is always right:
When receiving feedback on a practice talk,
do not be defensive!

The audience is always right

- Your tendency will be to be defensive when someone claims an idea in your talk was not explained well or was not clear

- You will find yourself turning to the relevant slide in your talk and saying “I mentioned that here”.

- The customer (the audience) is always right in this situation. Sure, you might have mentioned it, but if it wasn’t understood, it’s your fault not theirs.
  - Find a way to make it more clear!

- The correct response is to turn to the appropriate slide and say:
  - “I tried to explain that idea here. And this is what I was trying to say. What could I have said to make that point more clear?”

- The complainer should then work with you to explain what they interpreted instead, and offer suggestions on what information they would require to have better understood.

Wrap Up

General principles to keep in mind

Identify your audience, and strive for perfect clarity for them.

“Every sentence matters.”

“Show, don’t tell.”

“The audience prefers not to think” (about things you can just tell them)

“Surprises are bad”: say <u>why</u> before what
(indicate why you are saying something before you say it)

Always explain every figure, graph, or equation

One point per slide, and the point is the title of the slide

When improving the talk, the audience is always right

Summary

The good news…

Some simple principles enable much more efficient, and more rigorous, technical communication

Take pride in your craft. All students should follow these principles until they become second nature. Then you can start violating them.