

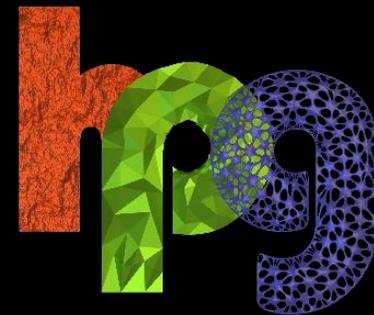
PATCH TEXTURES:

HARDWARE IMPLEMENTATION OF MESH COLORS

Ian Mallett Larry Seiler Cem Yuksel

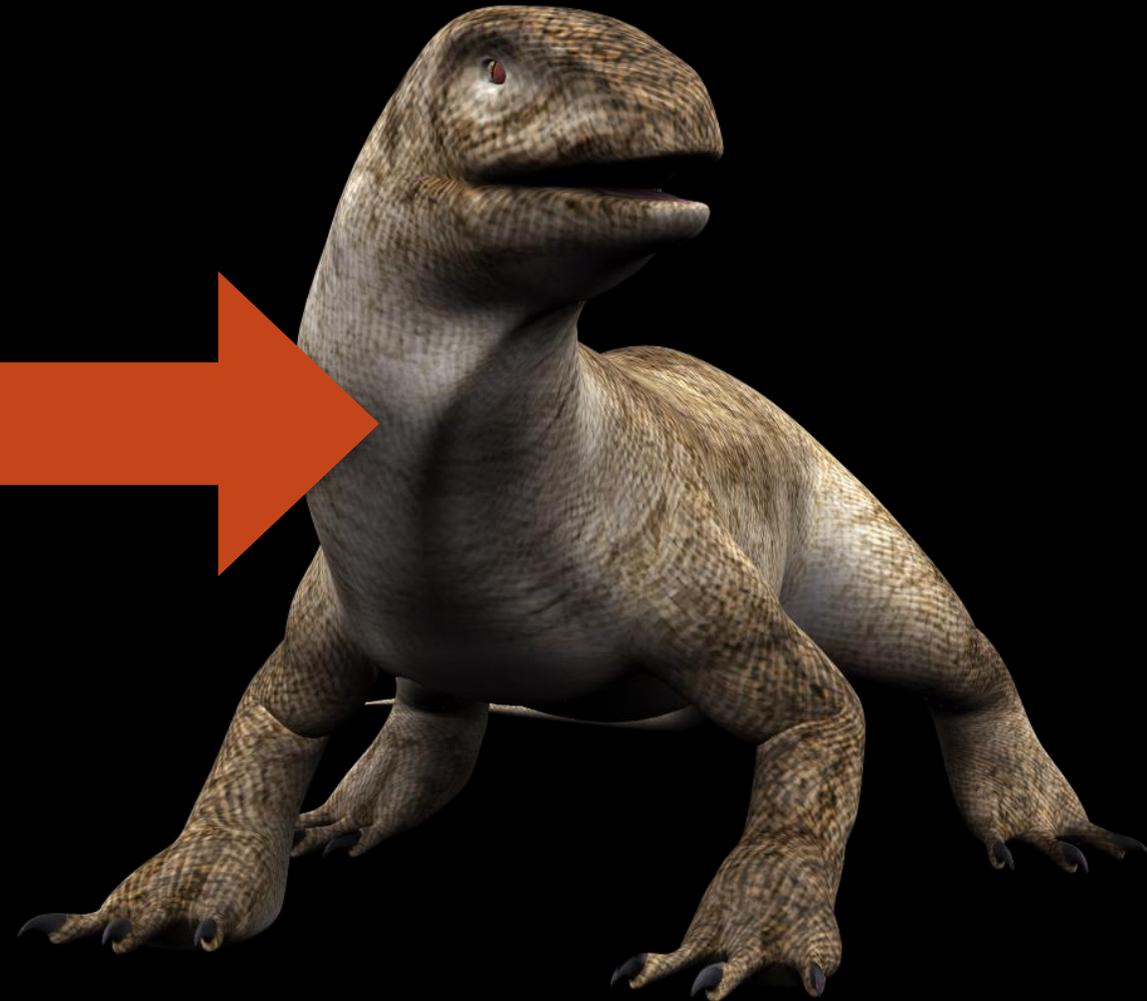
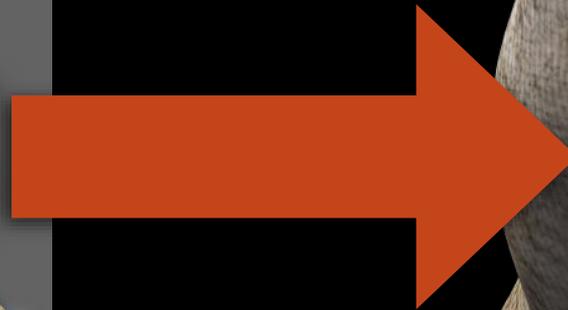


Utah Graphics

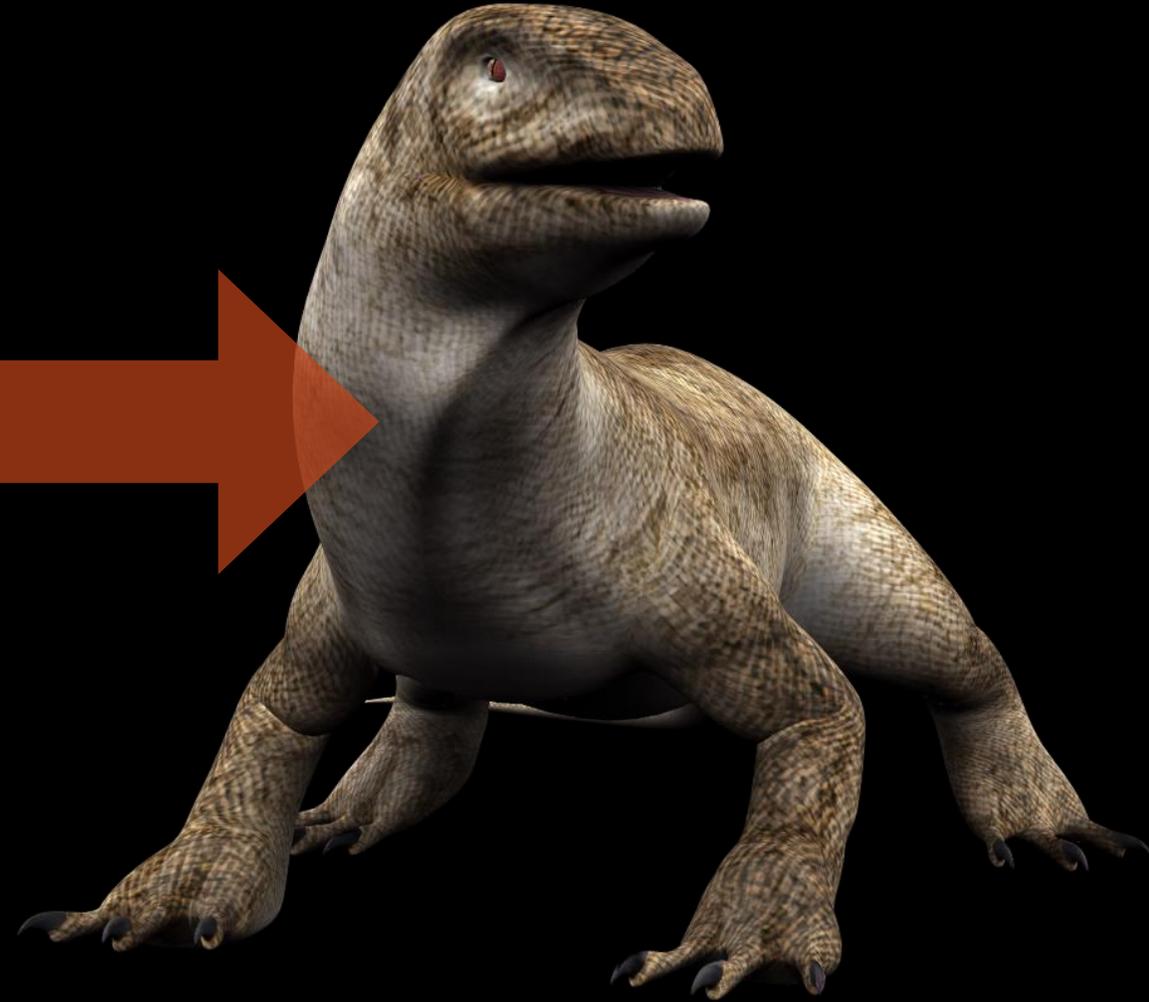
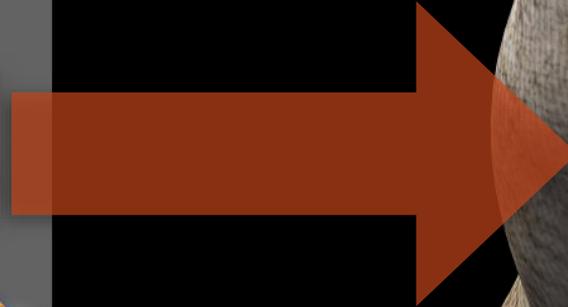
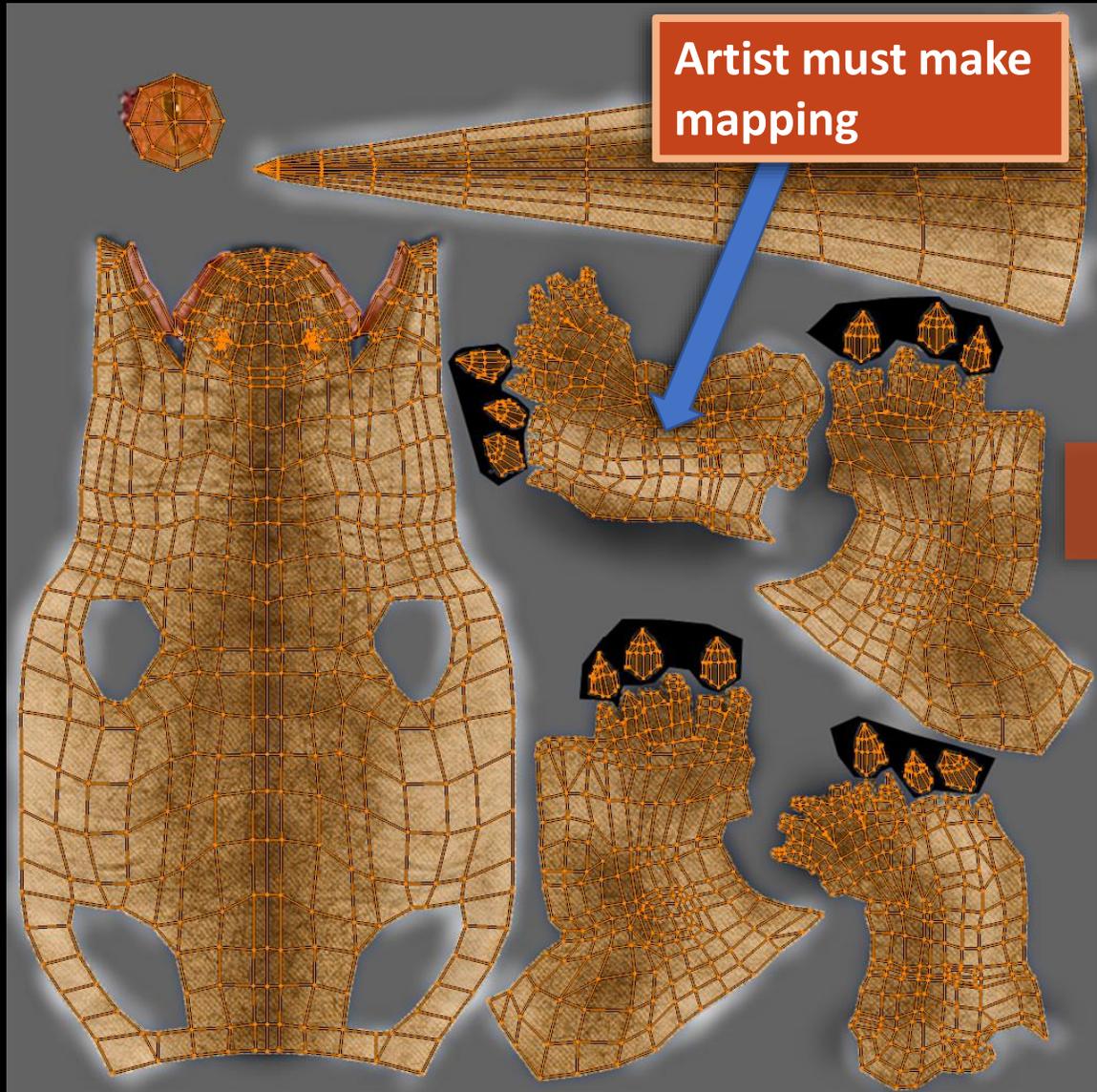


2019

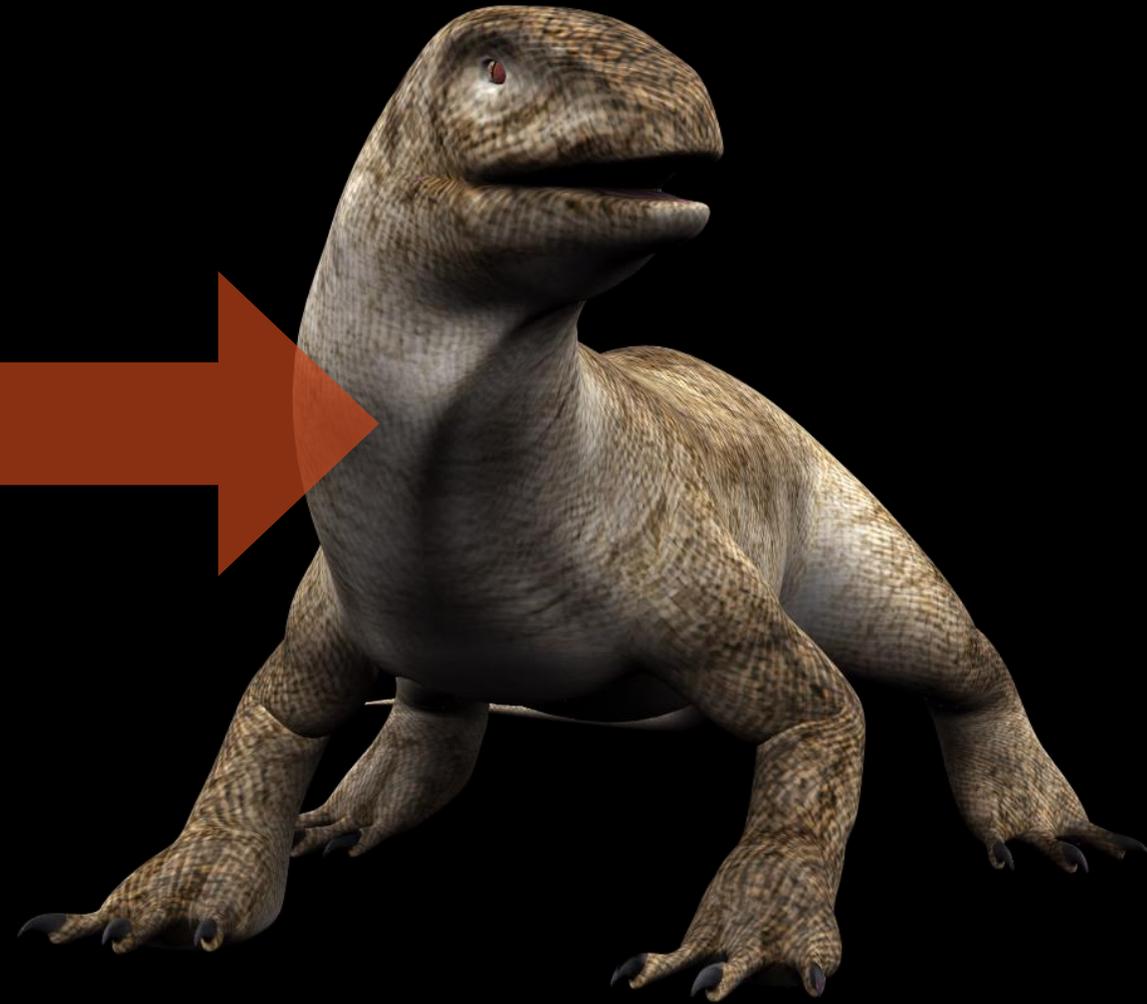
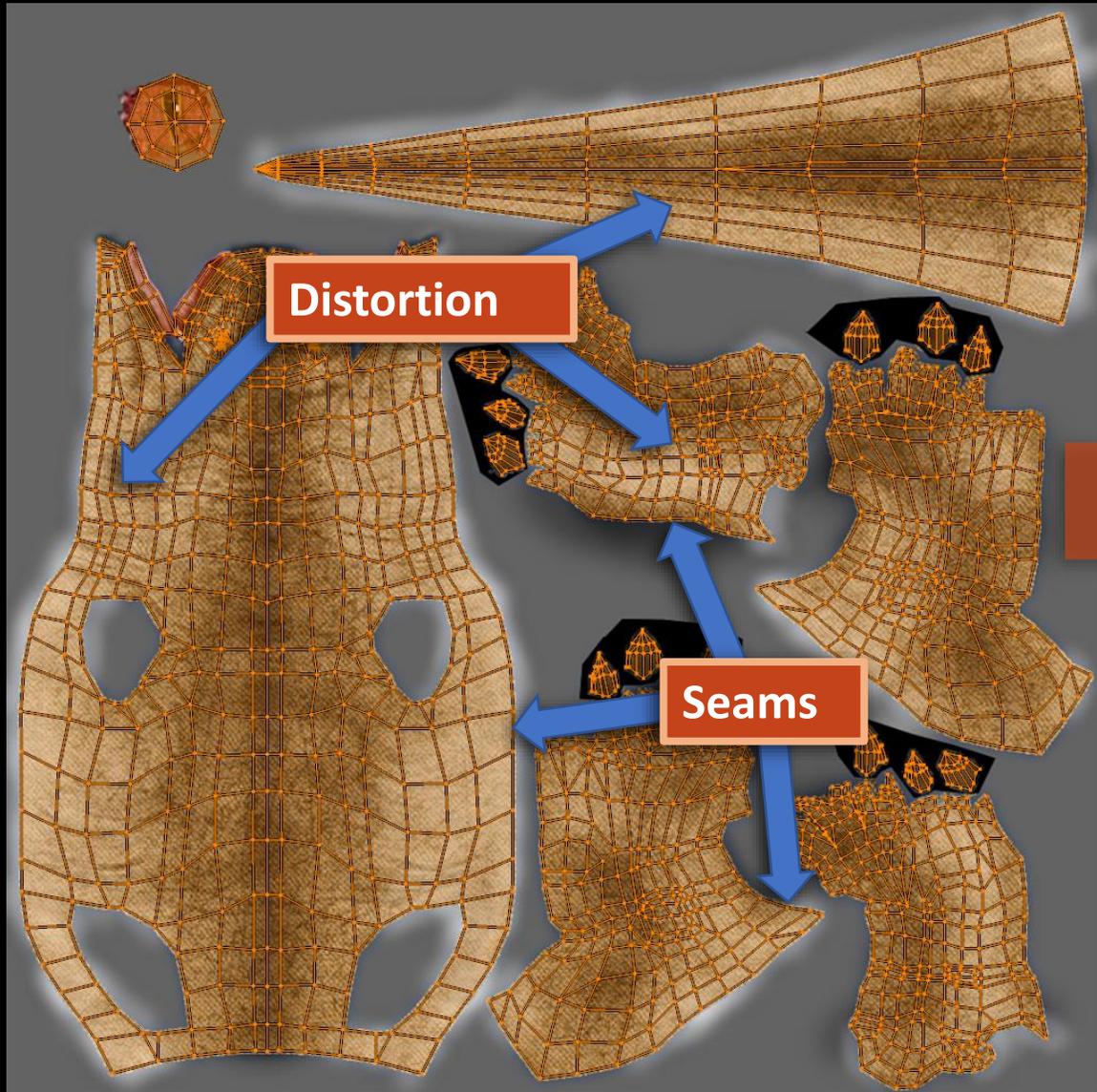
TEXTURES . . . HAVE PROBLEMS



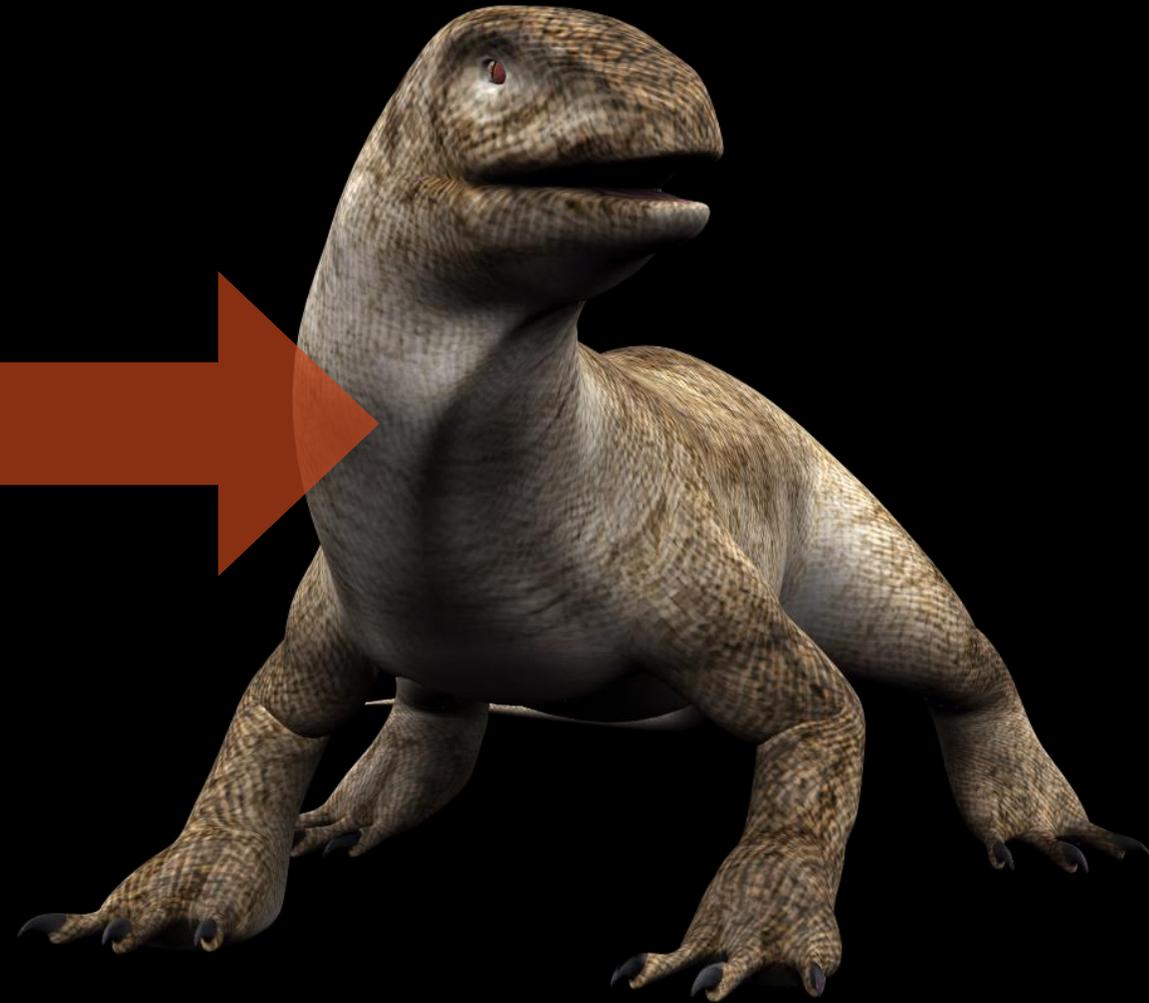
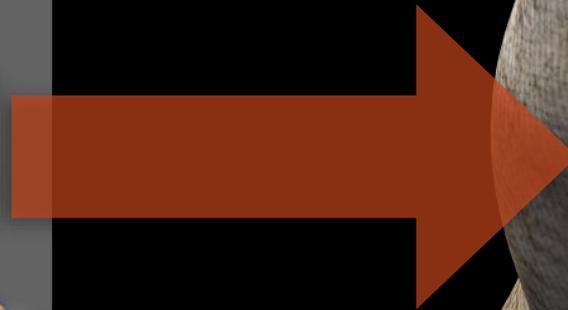
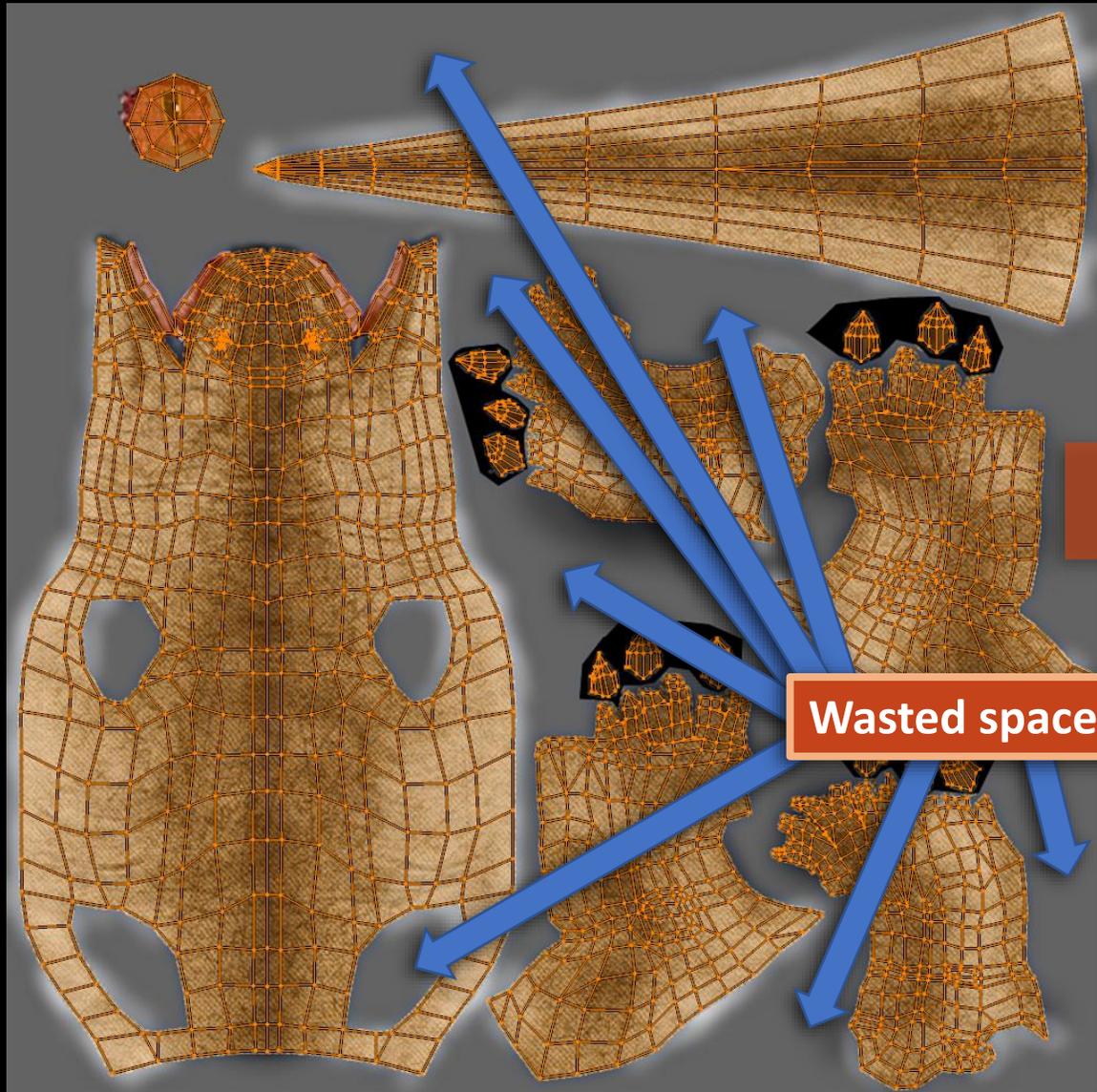
TEXTURES . . . HAVE PROBLEMS



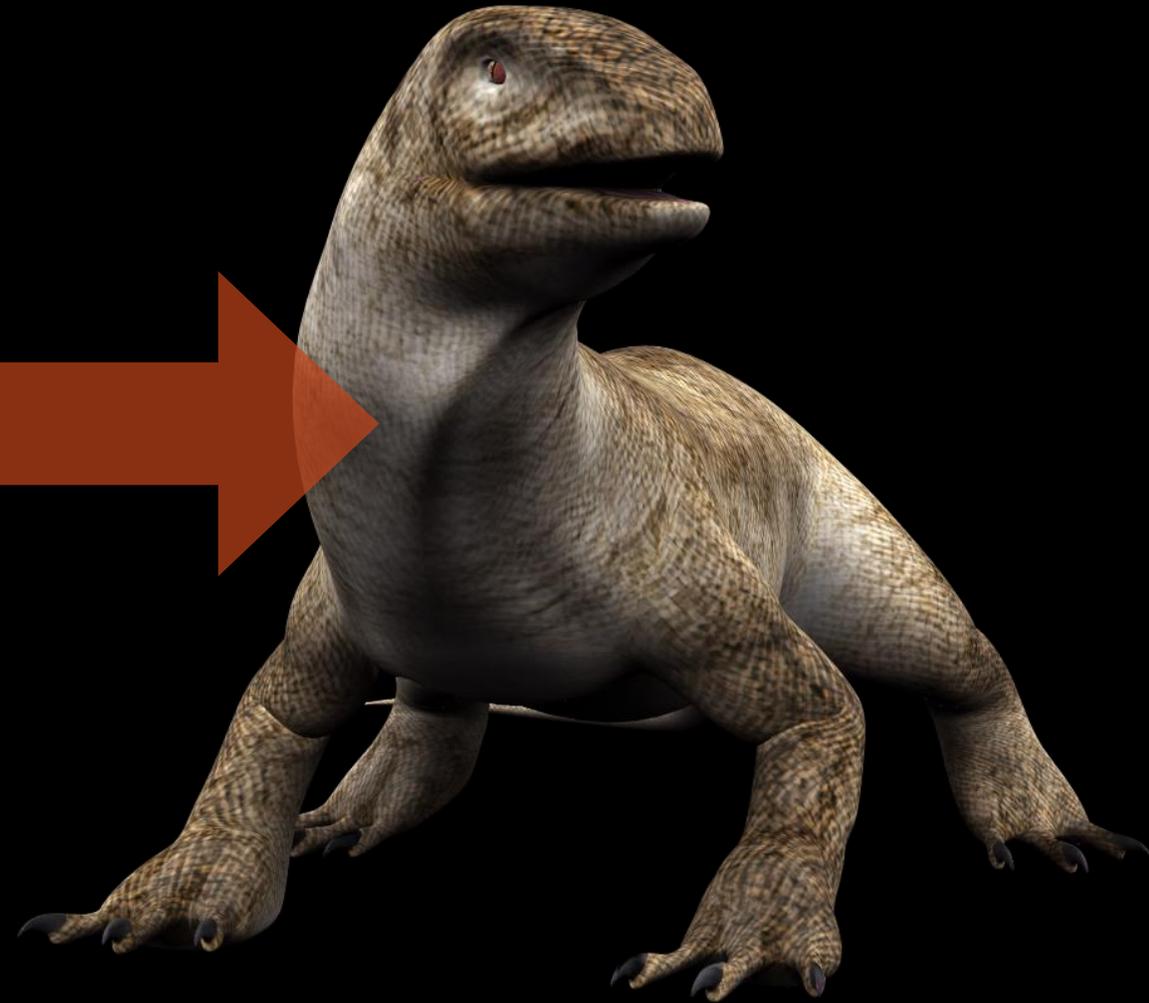
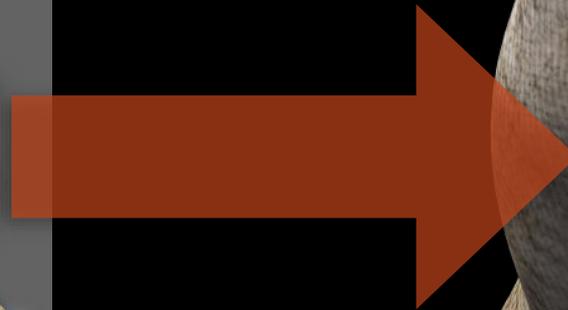
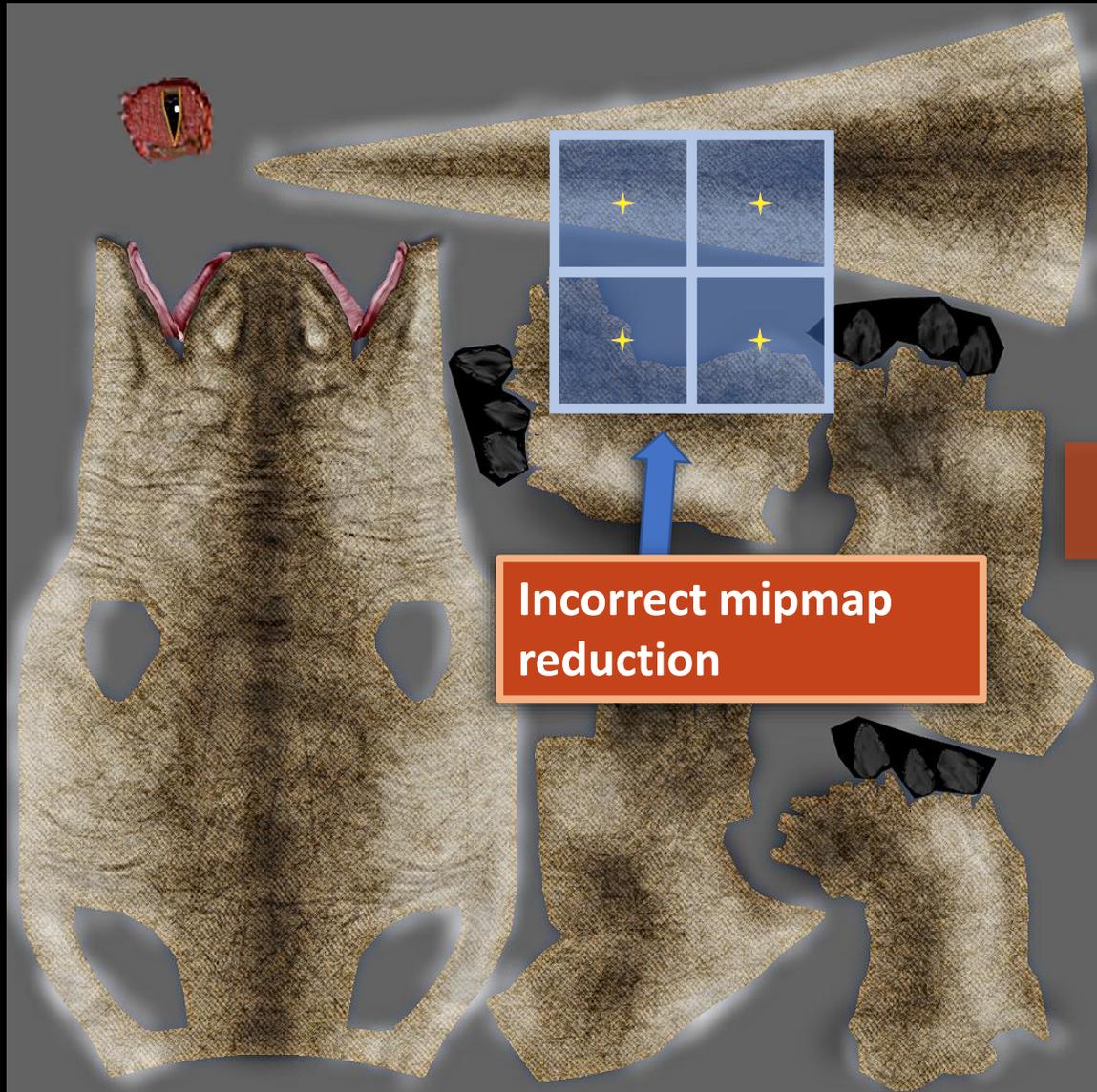
TEXTURES . . . HAVE PROBLEMS



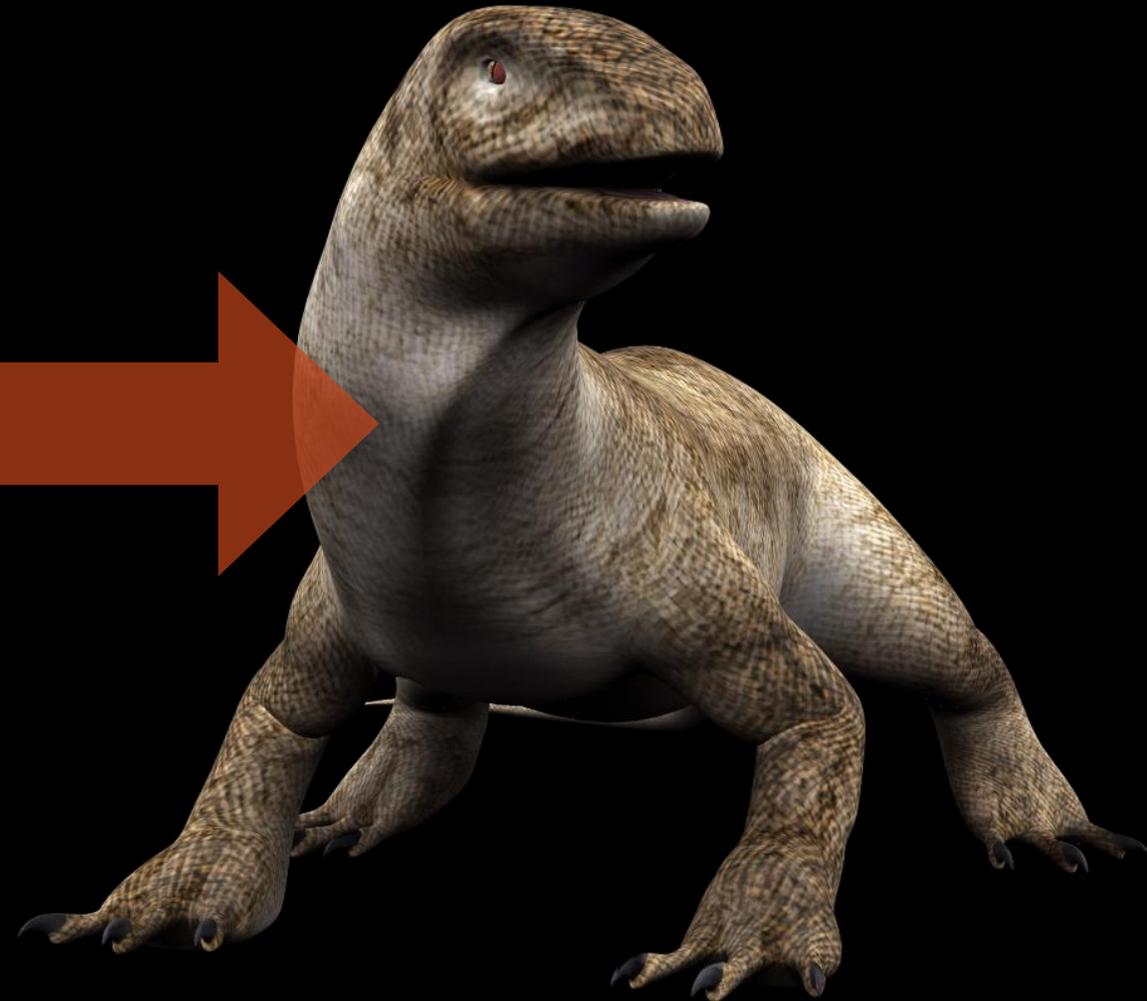
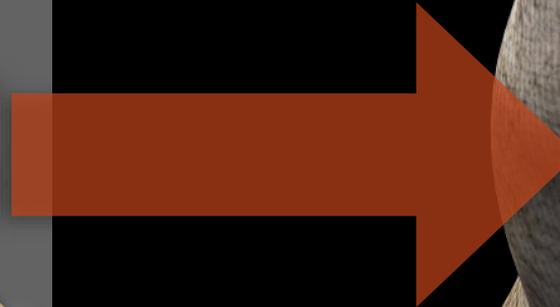
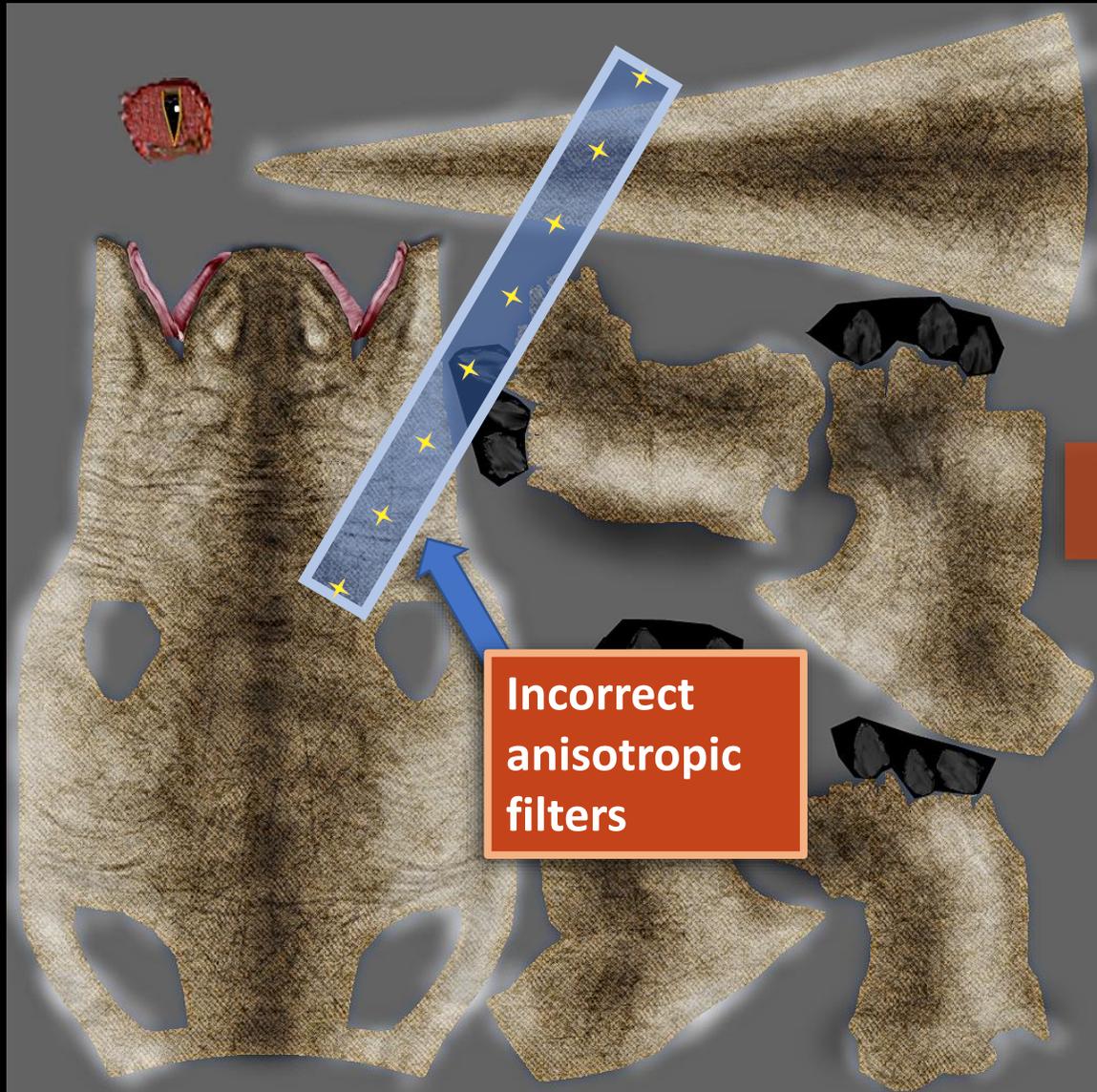
TEXTURES . . . HAVE PROBLEMS



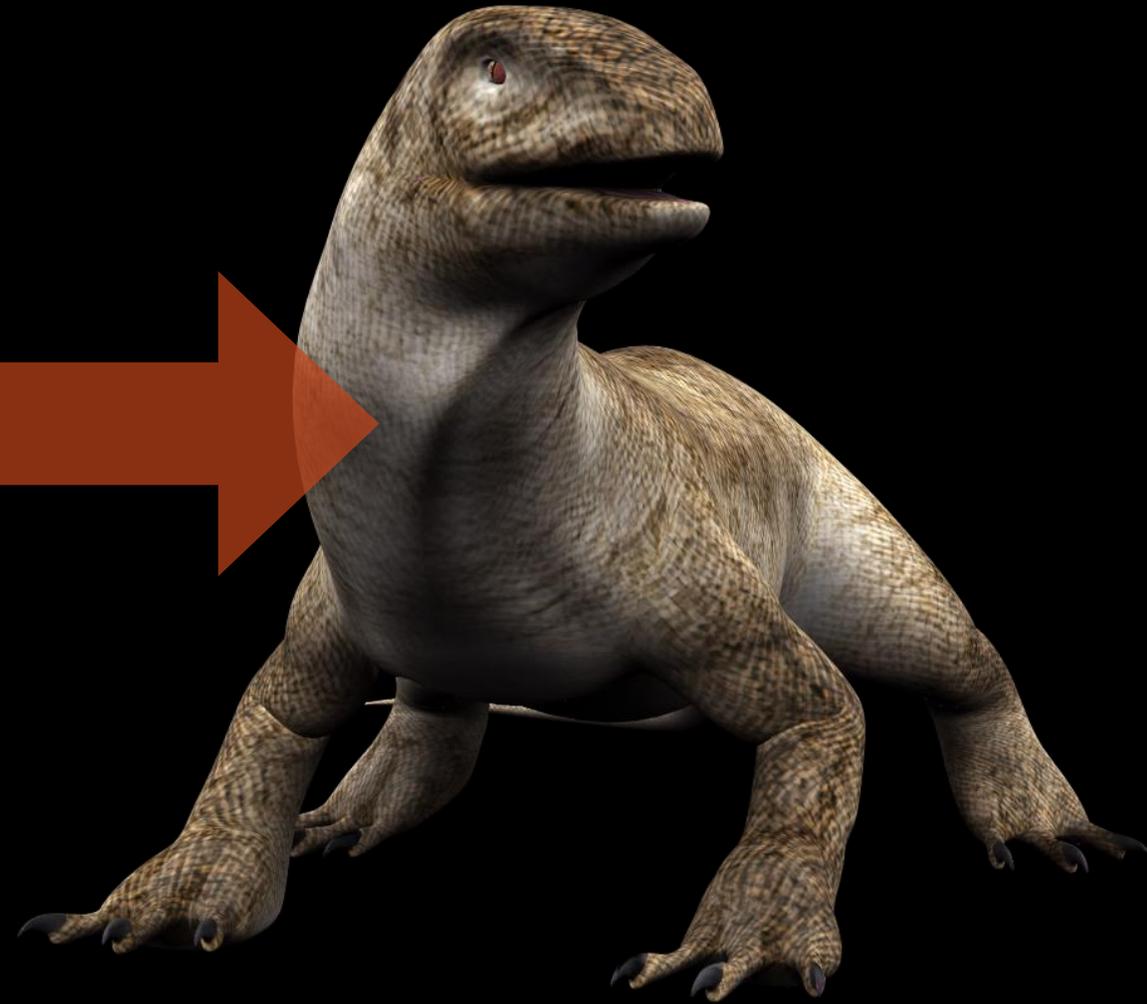
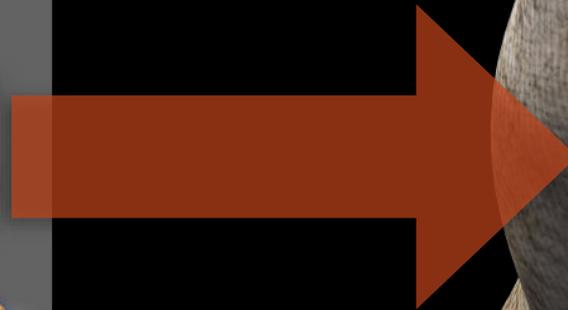
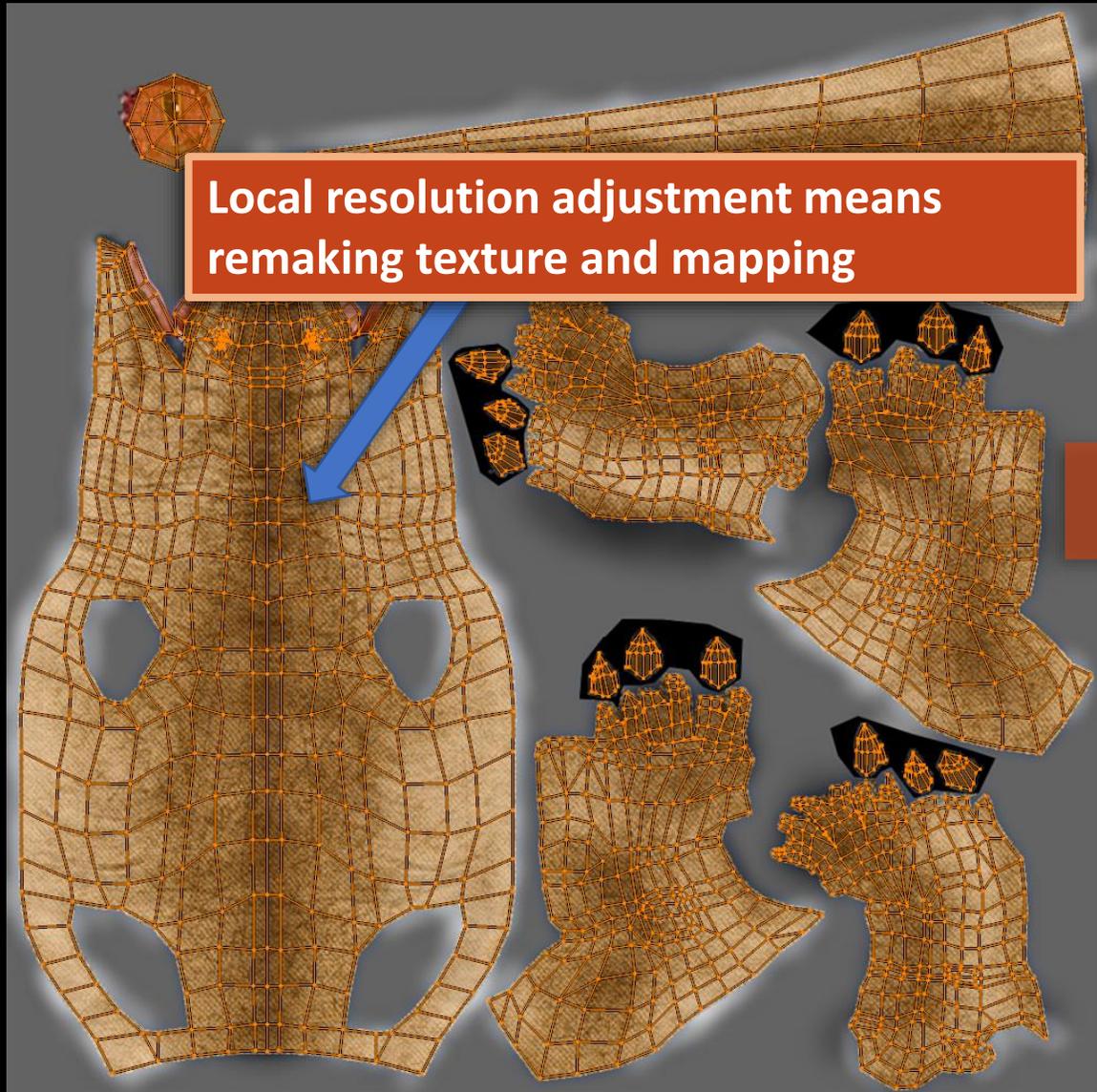
TEXTURES . . . HAVE PROBLEMS



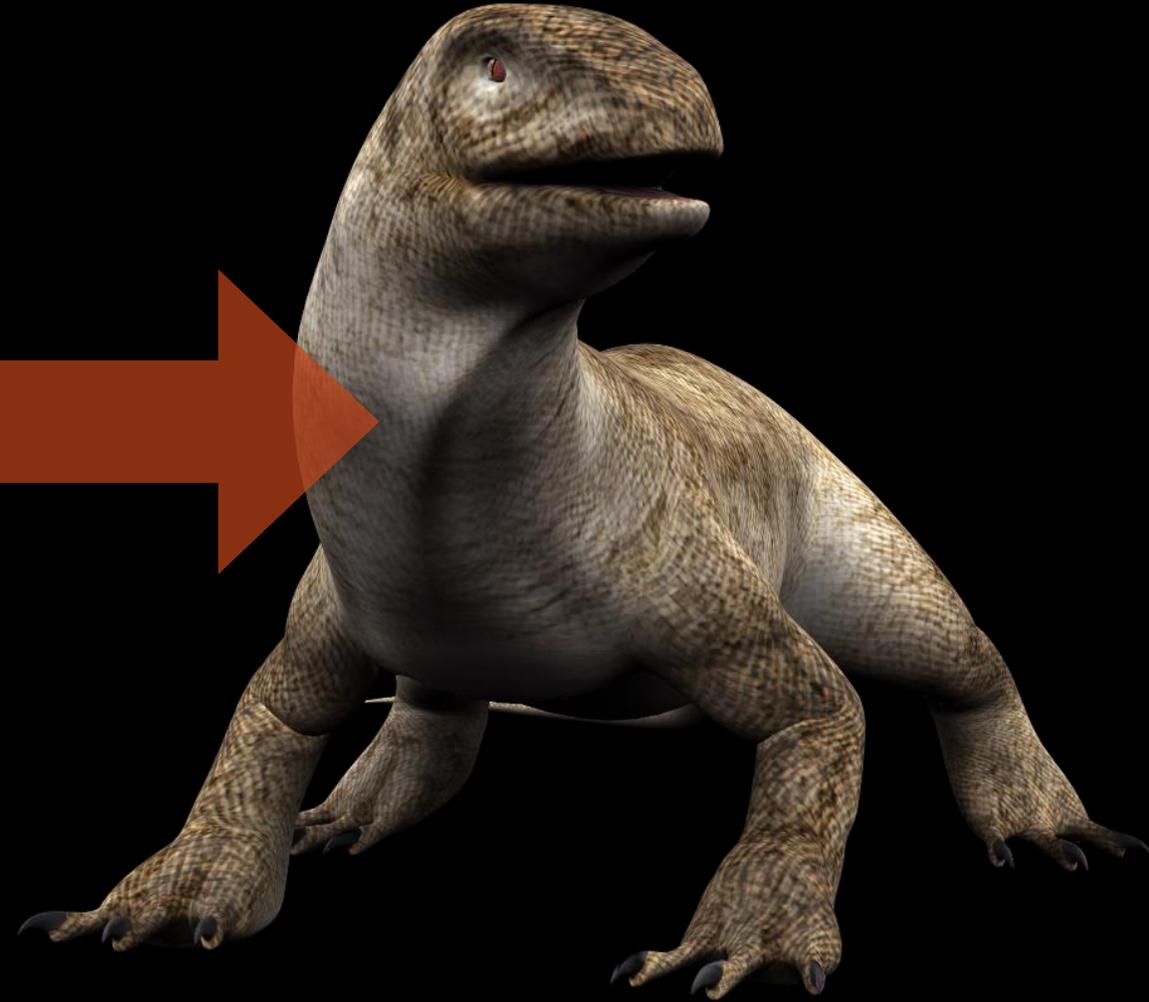
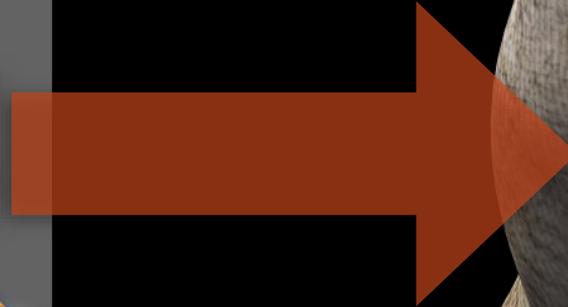
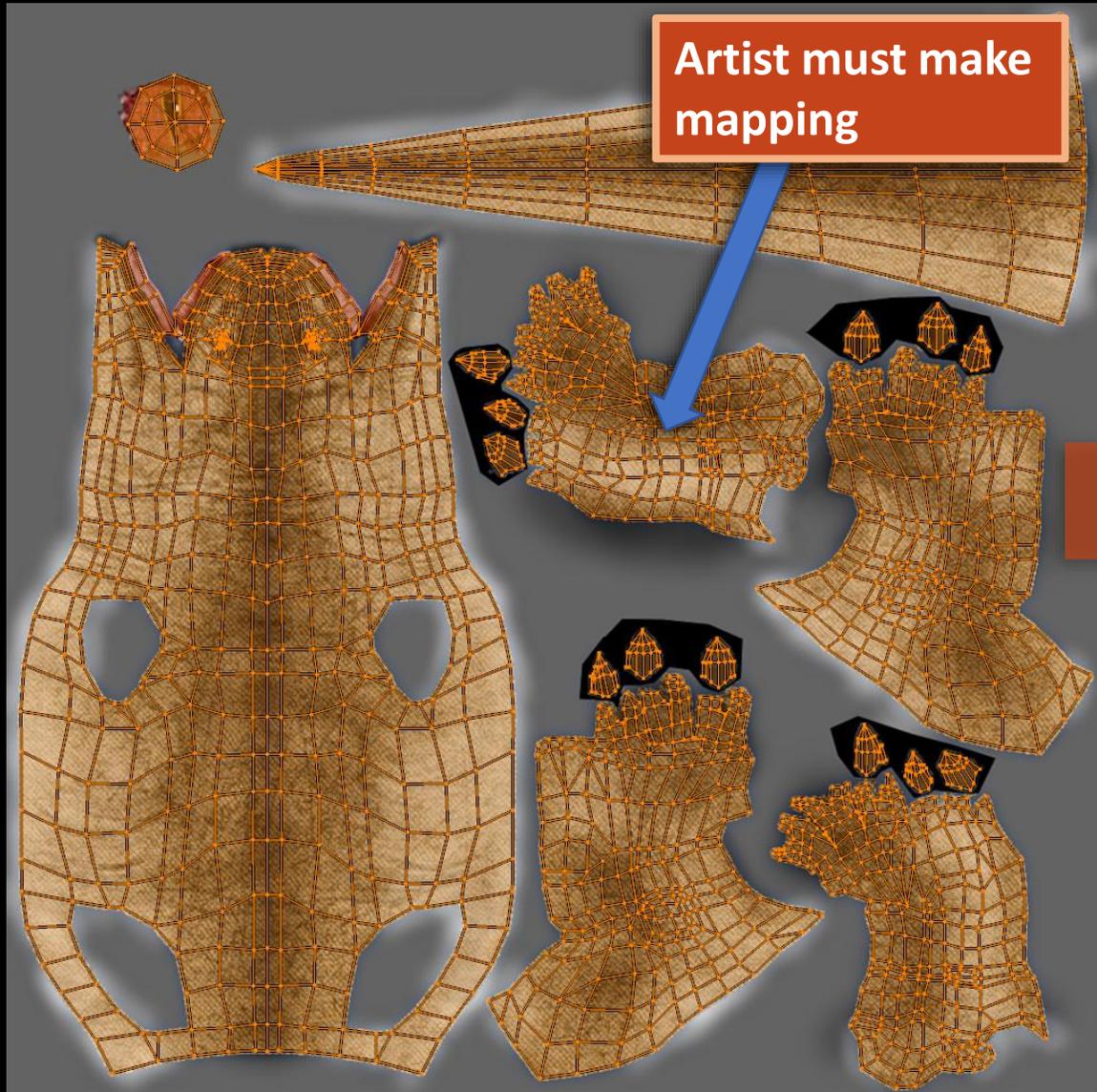
TEXTURES . . . HAVE PROBLEMS



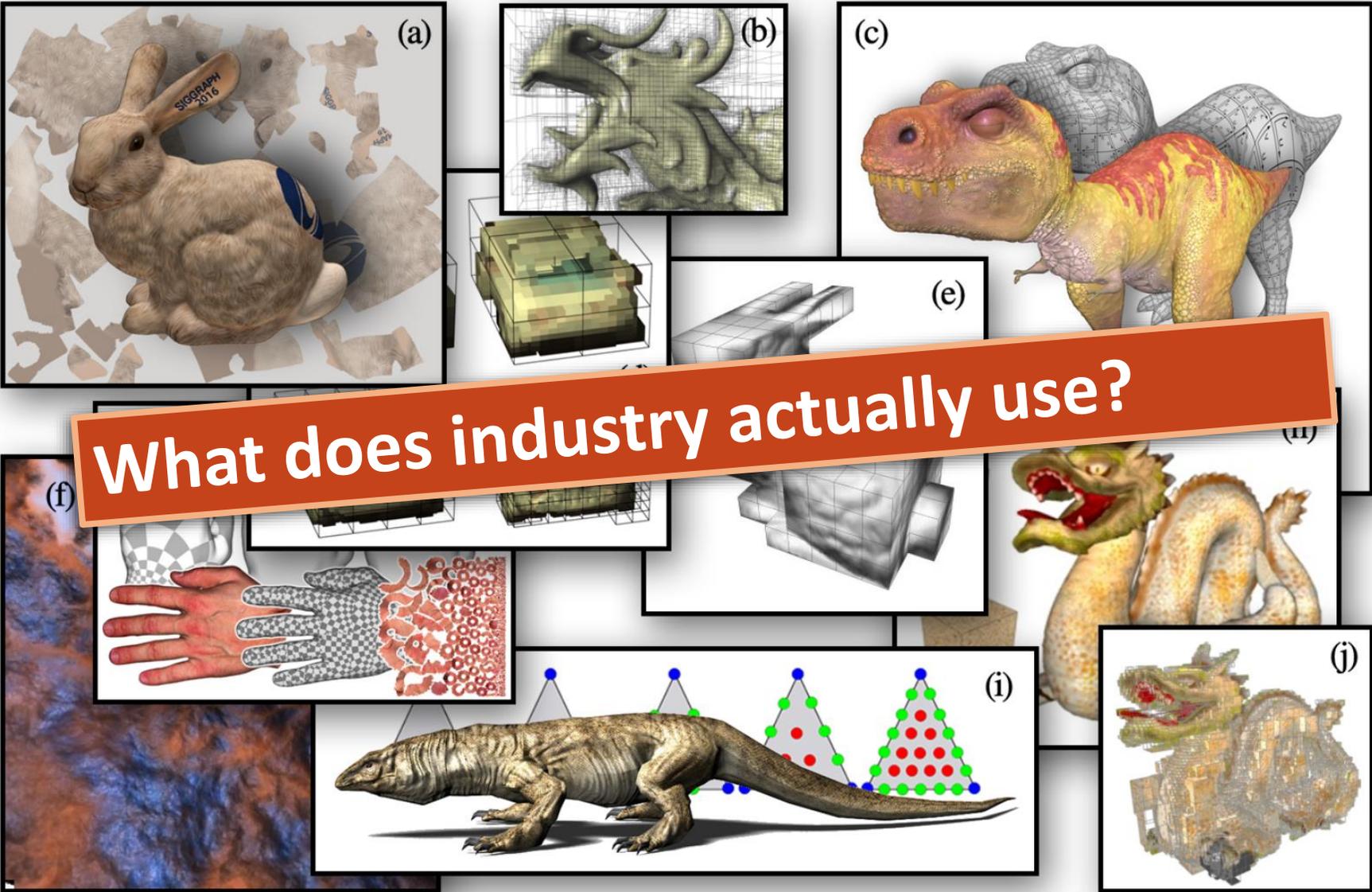
TEXTURES . . . HAVE PROBLEMS



TEXTURES . . . HAVE PROBLEMS



LOTS OF DIFFERENT APPROACHES . . .



What does industry actually use?

(Figure [Yuksel et al. 2019])

TEXTURE DATA LIVING ON SURFACES!

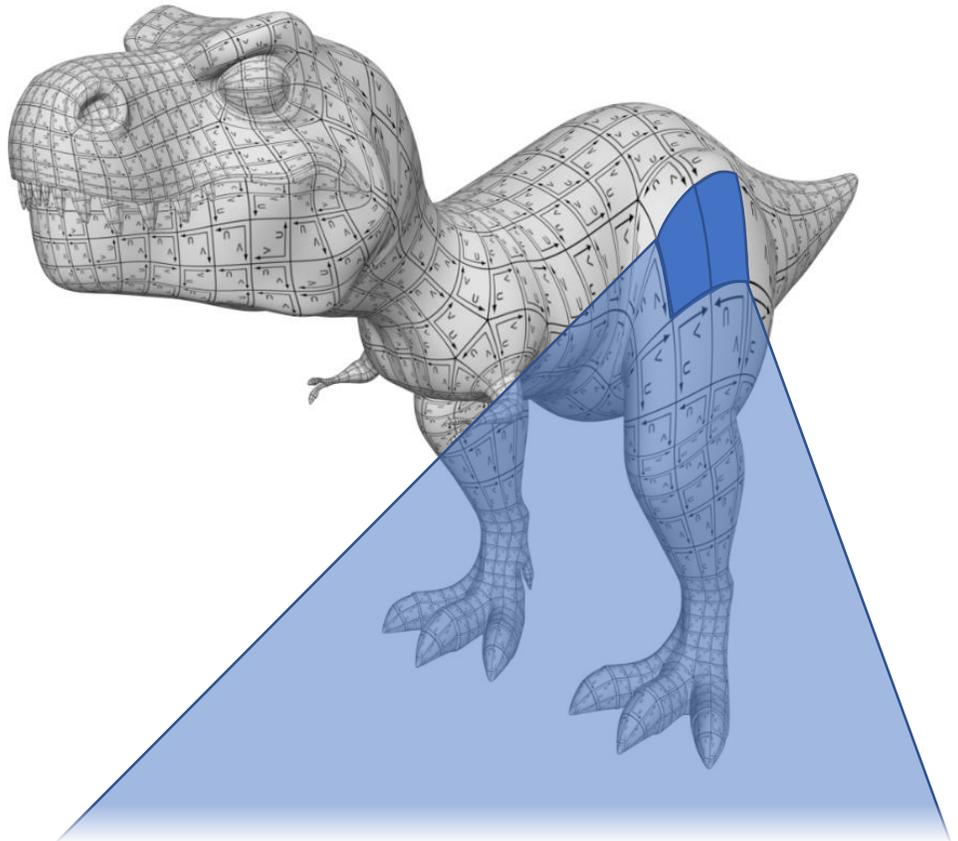
- Ptex and Mesh Colors



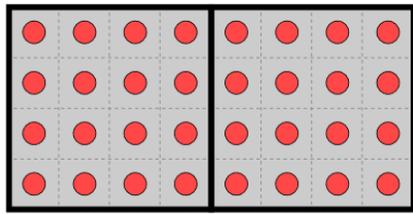
TEXTURE DATA LIVING ON SURFACES!

- Film uses texture data directly on surfaces
 - Much easier to model!
 - Get all the rendering benefits too
- Ptex widely (sometimes exclusively) used
- Mesh Colors has also been used in production

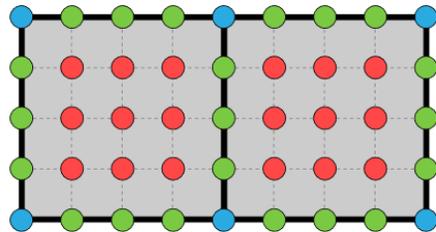
MESH COLORS AND PTEX



Every model has a list of patches, each with its own separate texture.

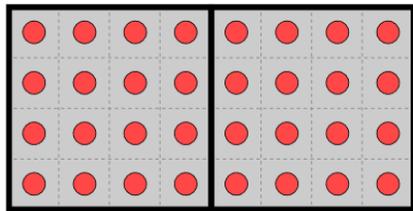
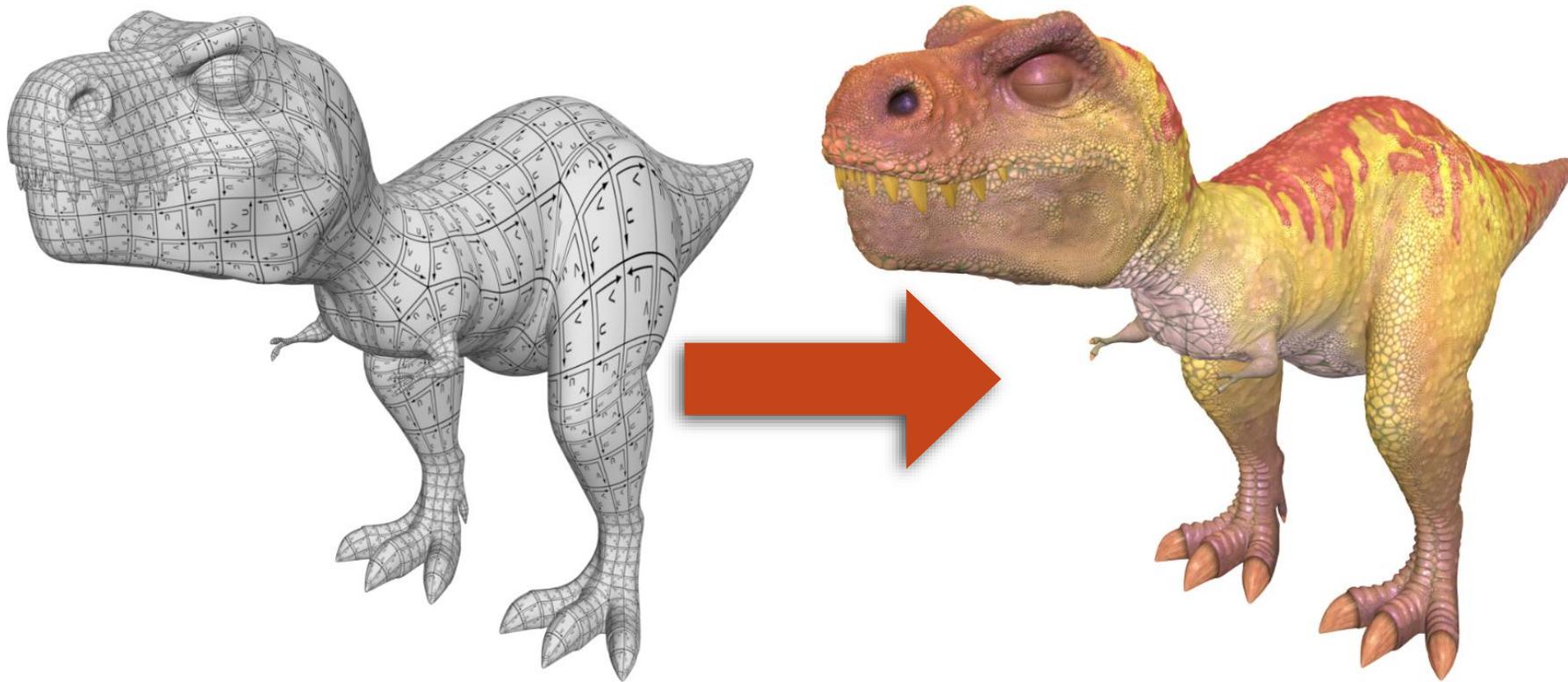


Ptex

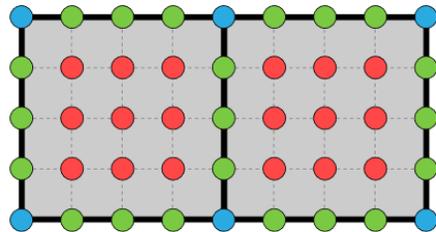


Mesh Colors

MESH COLORS AND PTEX



Ptex



Mesh Colors

MESH COLORS OR PTEX . . . IN REALTIME?

- Possible [McDonald 2013], [Yuksel et al. 2010], ...
- Still less-practical than 2D textures due to SW (or SW+HW) implementation:
 - Software is much-slower than hardware!
 - Complicated to emulate filtering functionality
 - Implementation limitations

MESH COLOR TEXTURES [YUKSEL 2017]

- Implements Mesh Colors using standard 2D textures
 - Almost as fast as 2D textures for simple filtering
 - Added shader complexity
 - Complicated implementation
 - Anisotropic filtering problematic
- What if the required HW changes to implement directly were small?

OUR CONTRIBUTION

- We show that implementing Mesh Colors on existing GPUs would require minimal HW changes
- Introduce “patch texture” representation—this is what makes the required HW changes minor!

Can leverage existing storage!

Can leverage existing filtering HW!

Edge-crossing unnecessary

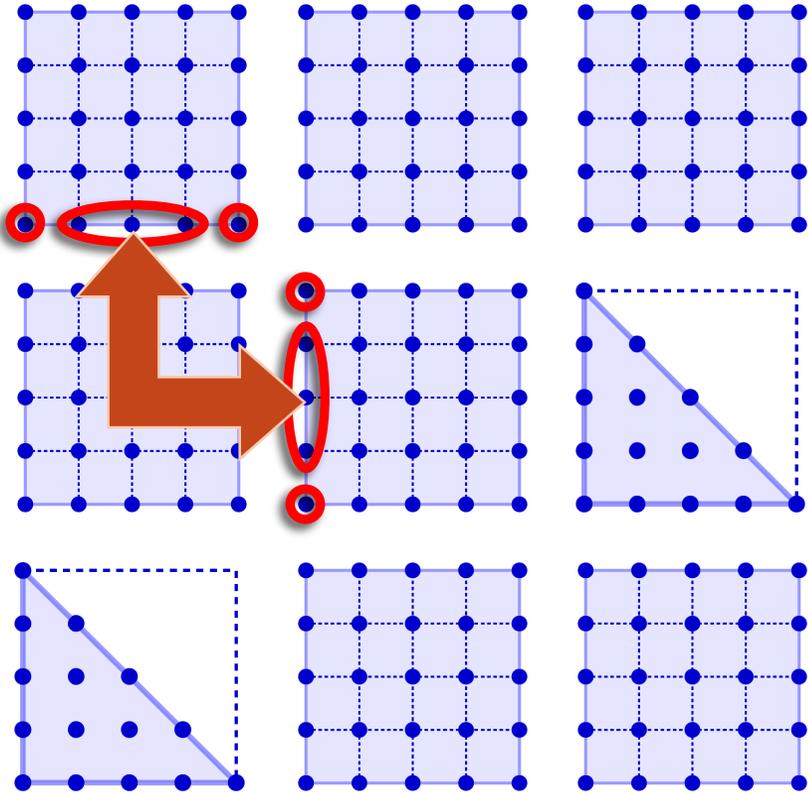
Similar performance expected

PATCH TEXTURES



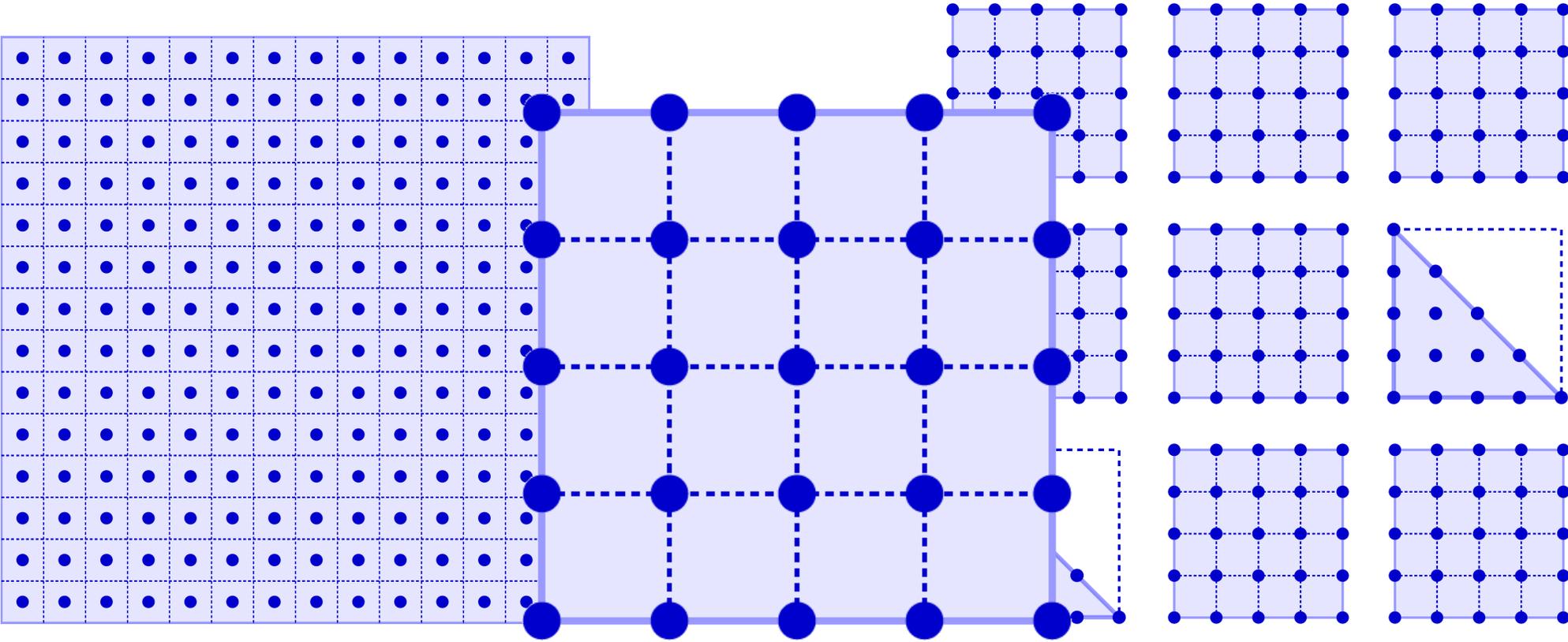
Standard 2D Texture

Unlike in Mesh Colors, in a Patch Texture, edge and corner data is duplicated, not shared with neighbors.



Set of Patch Textures

PATCH TEXTURES



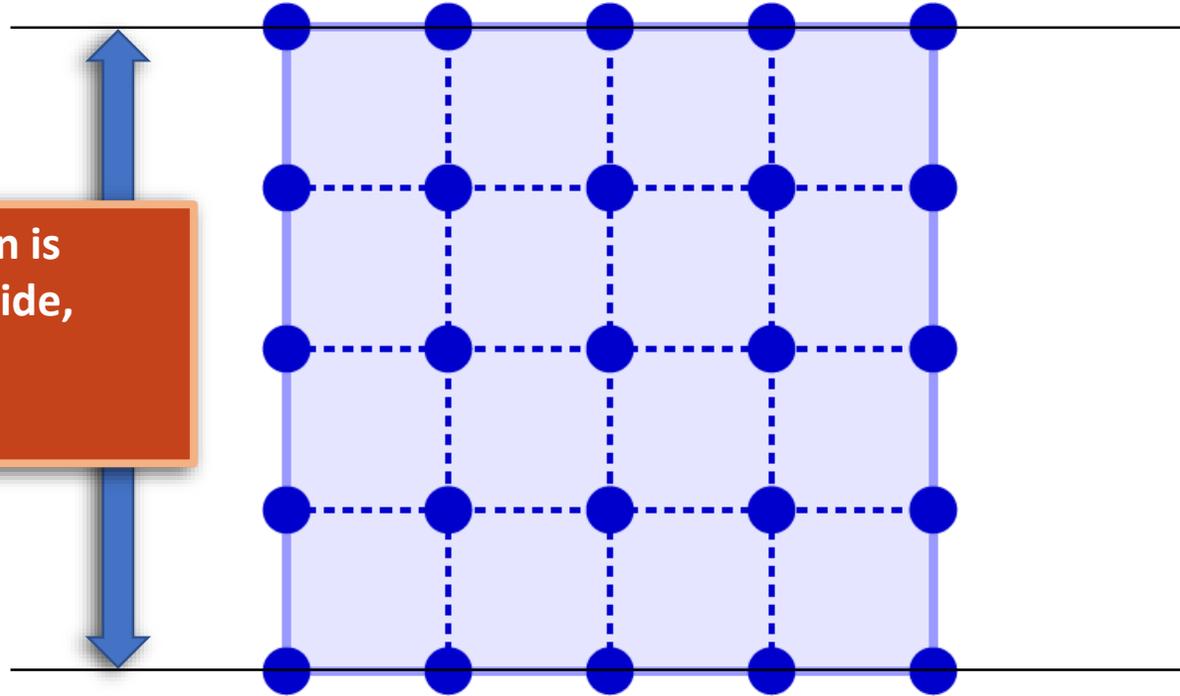
Standard 2D Texture

Set of Patch Textures

Individual Patch Texture

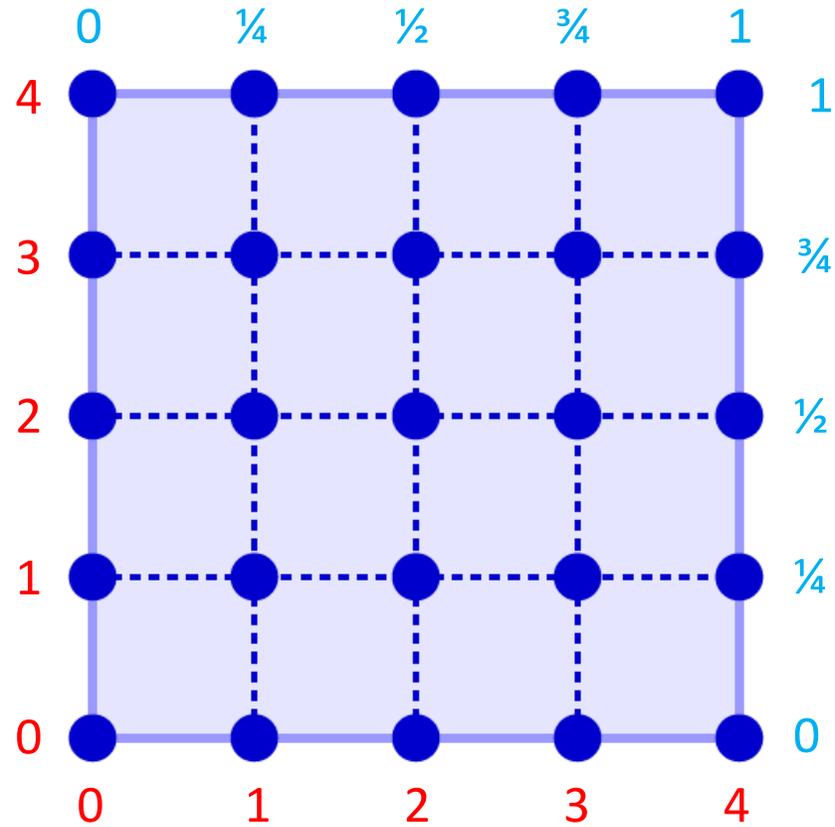
PATCH TEXTURES

No restriction on resolution is imposed, but if 2^n+1 on a side, Smooth filtering between neighbors is possible.



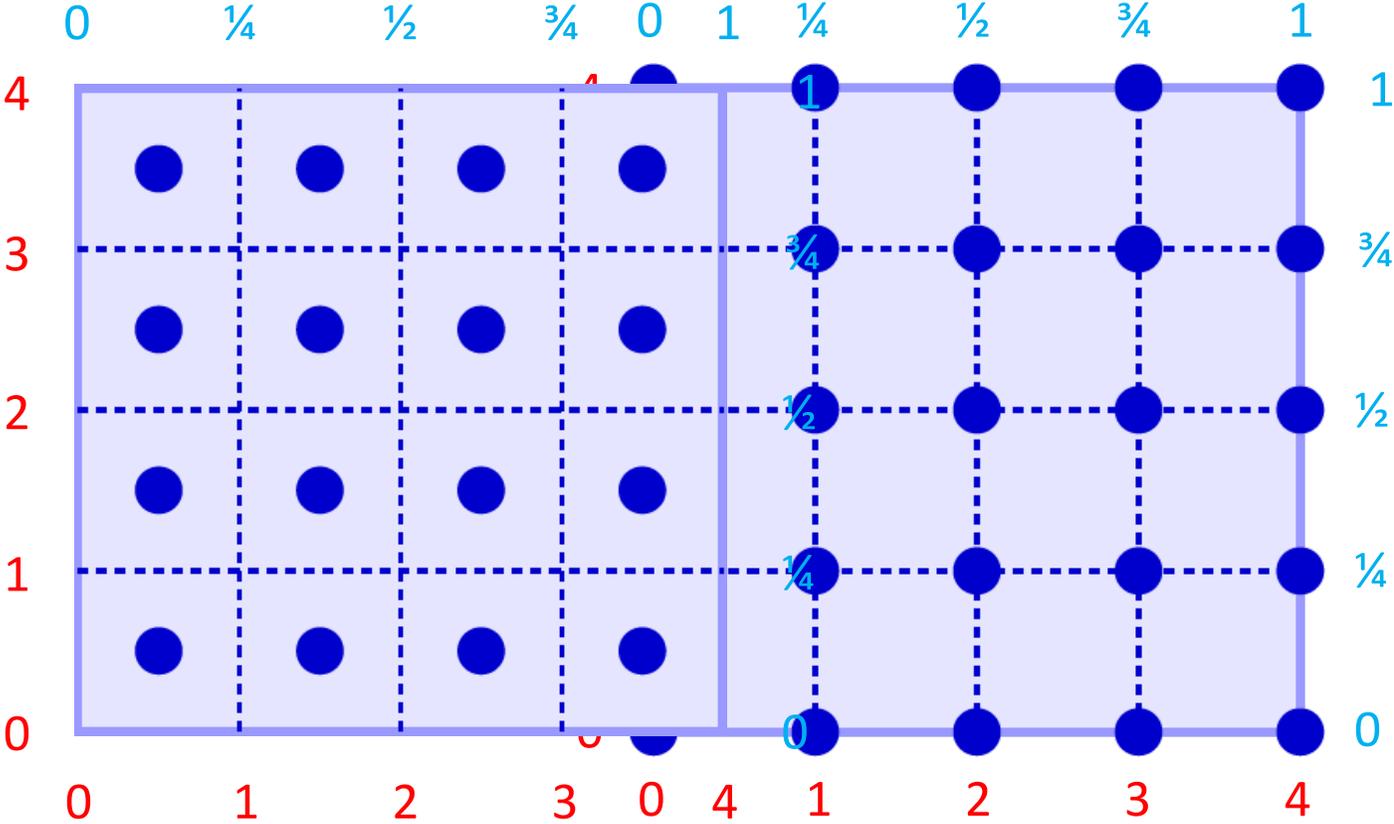
Individual Patch Texture

PATCH TEXTURES



(u,v) and (s,t) coordinates in a patch texture

PATCH TEXTURES



2D Texture (u,v) and (s,t) coordinates in a patch texture (u,v) and (s,t)

PATCH TEXTURE STORAGE

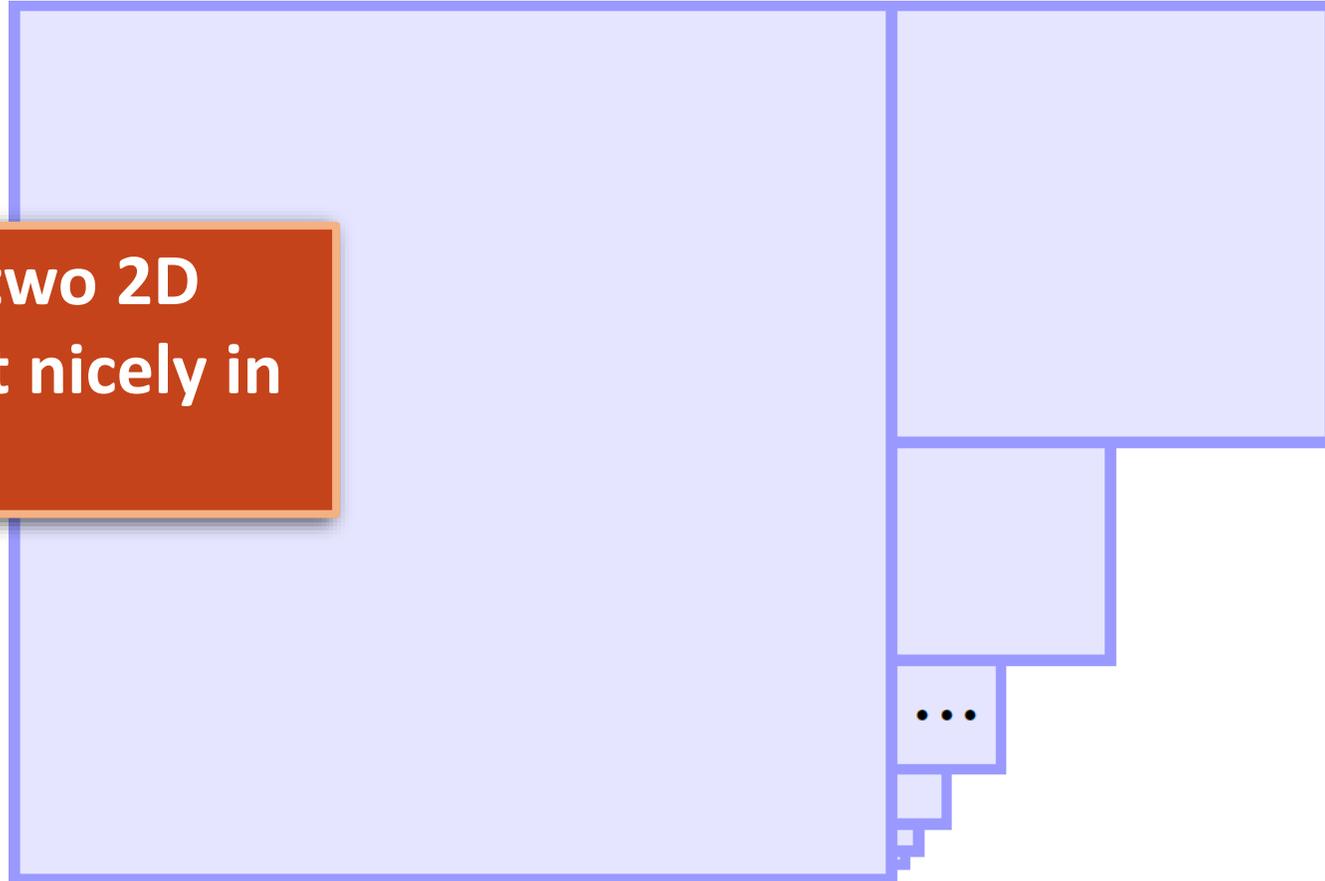
- Very similar to standard 2D textures

PATCH TEXTURE STORAGE (MIPMAPS)

- Very similar to standard 2D textures
- Most GPUs store in e.g. 4×4 tiles
 - If mesh textures are 2^n+1 on a side, requires padding

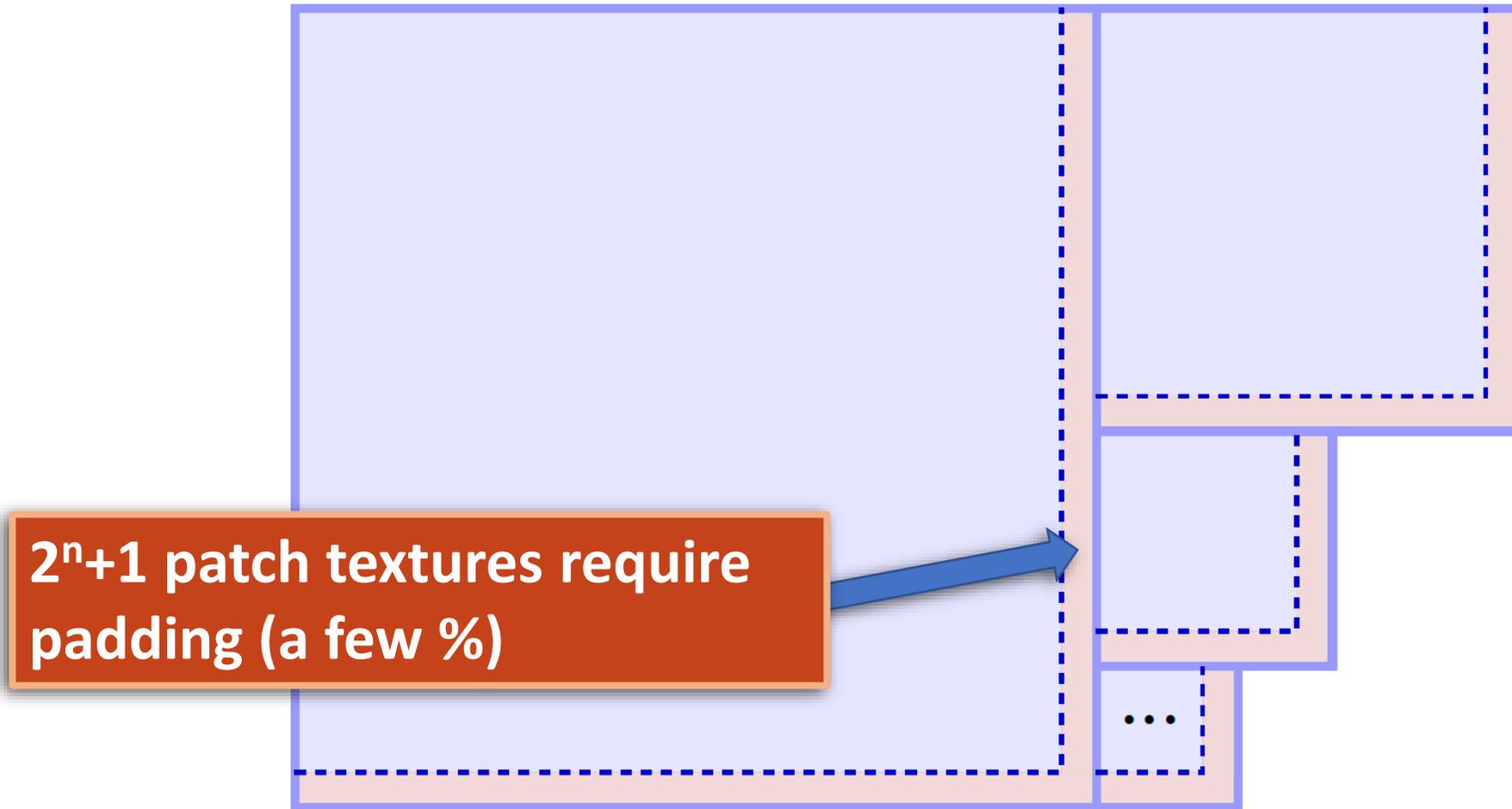
PATCH TEXTURE STORAGE (MIPMAPS)

Power-of-two 2D textures fit nicely in memory



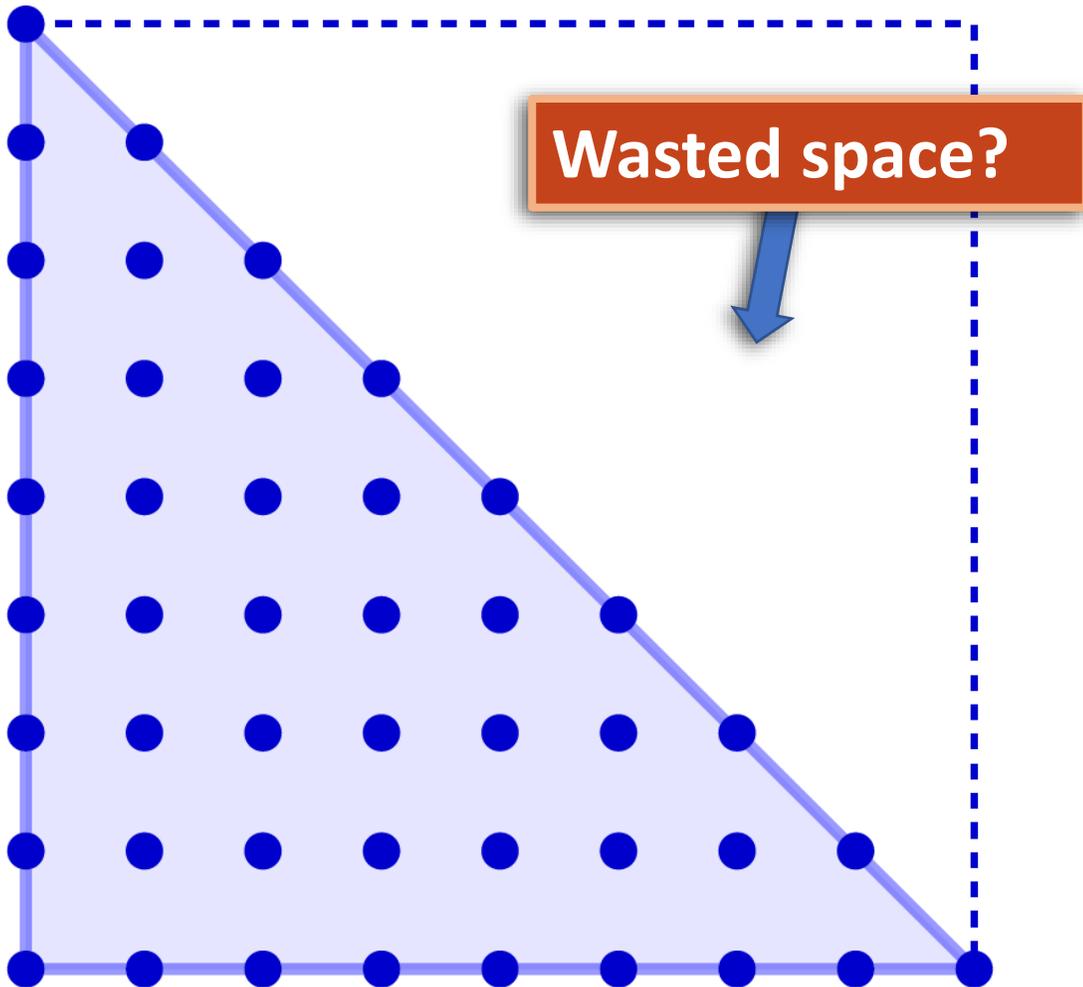
Standard 2D texture

PATCH TEXTURE STORAGE (MIPMAPS)



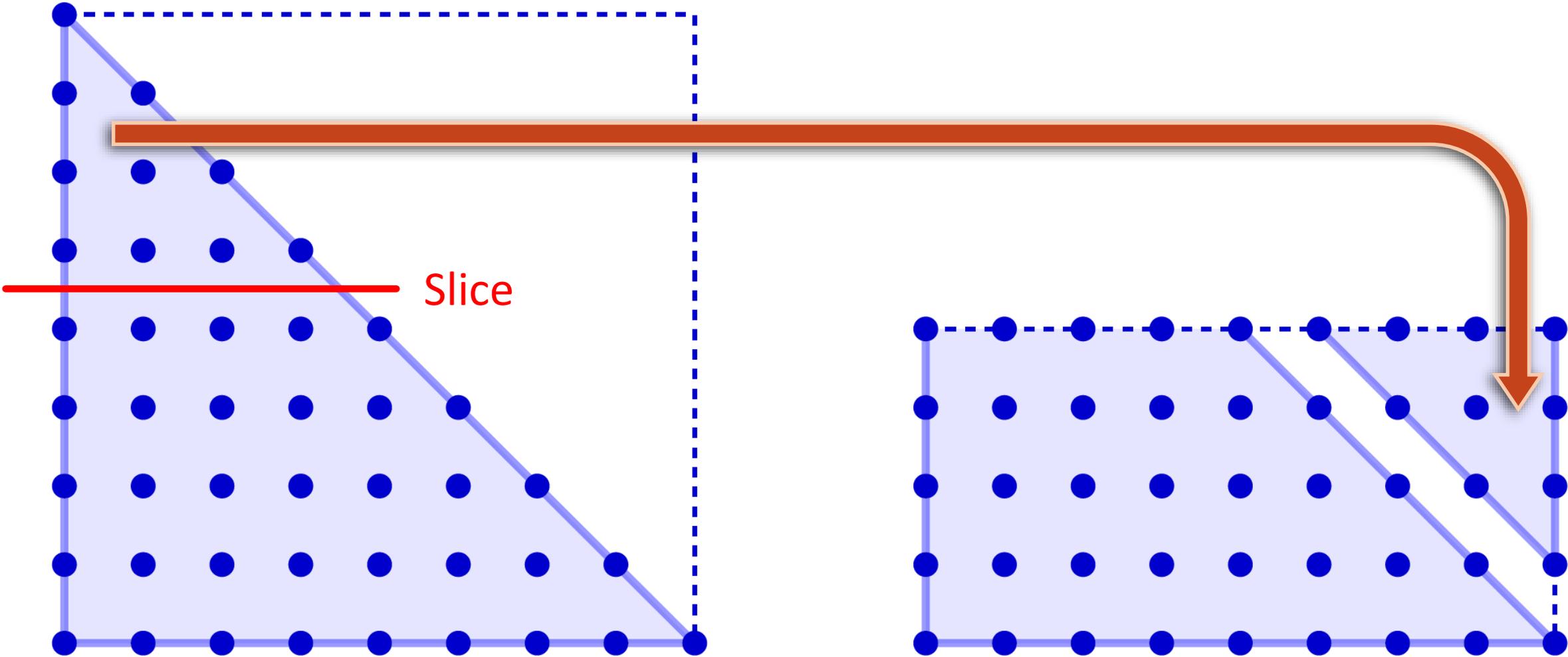
Patch texture

PATCH TEXTURE STORAGE (TRIANGULAR PATCHES)

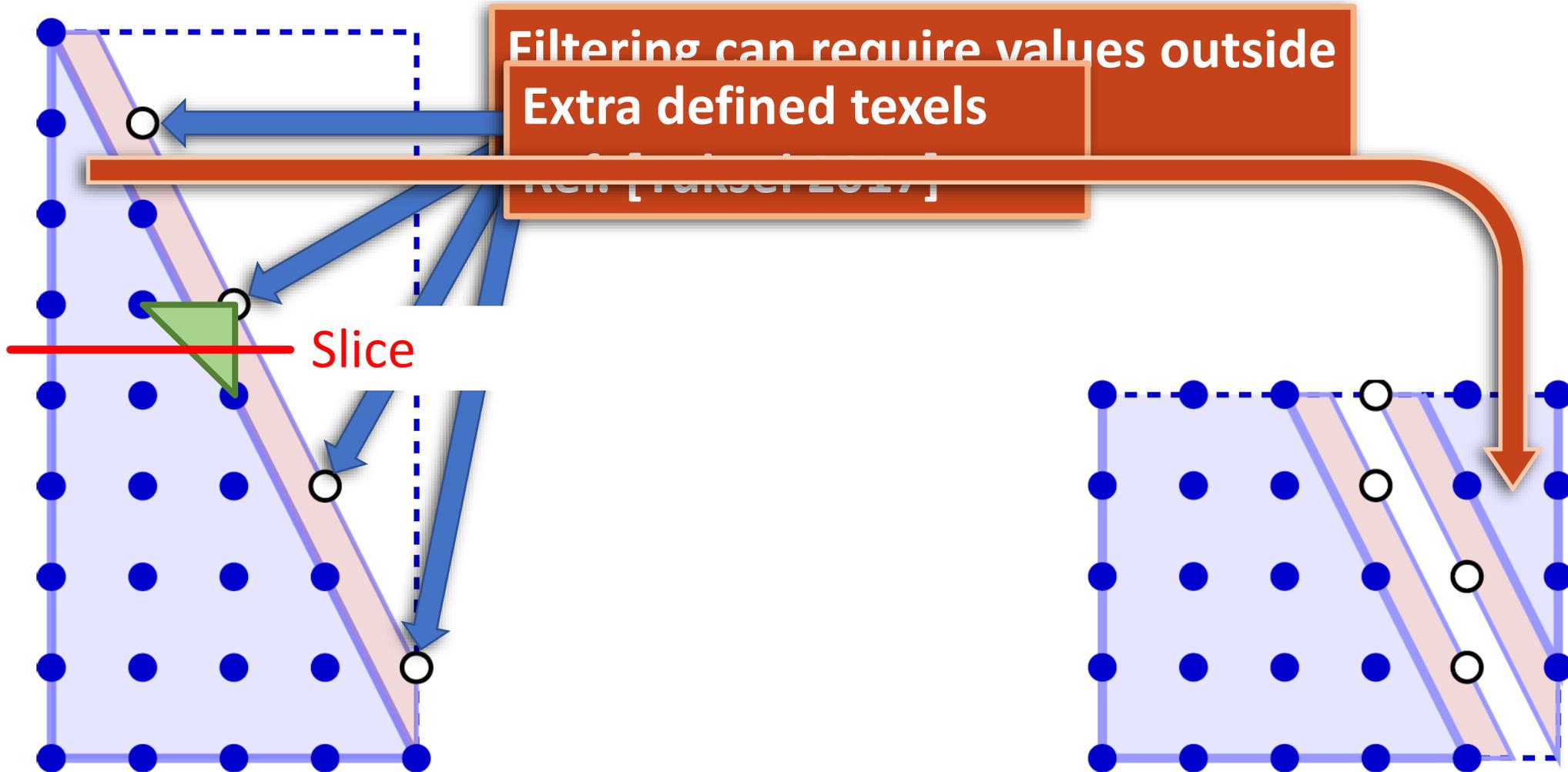


- Triangular patch textures do not map nicely to 2D storage
- Not necessarily a problem
 - Quad-dominant meshes standard
 - Patch textures typically small
- Clever workarounds exist (though require more HW changes)

PATCH TEXTURE STORAGE (TRIANGULAR PATCHES)

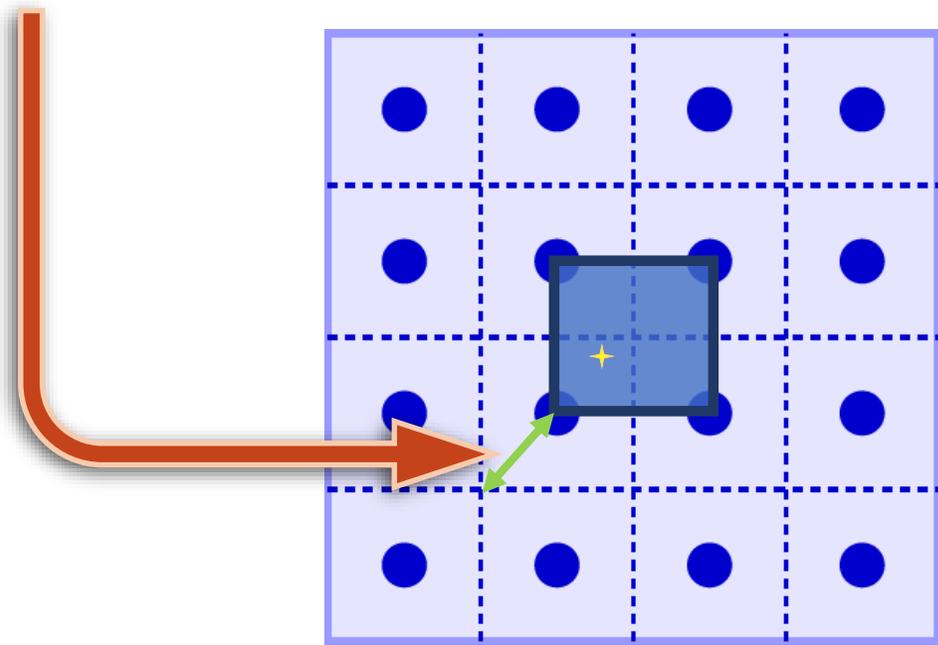


PATCH TEXTURE STORAGE (TRIANGULAR PATCHES)

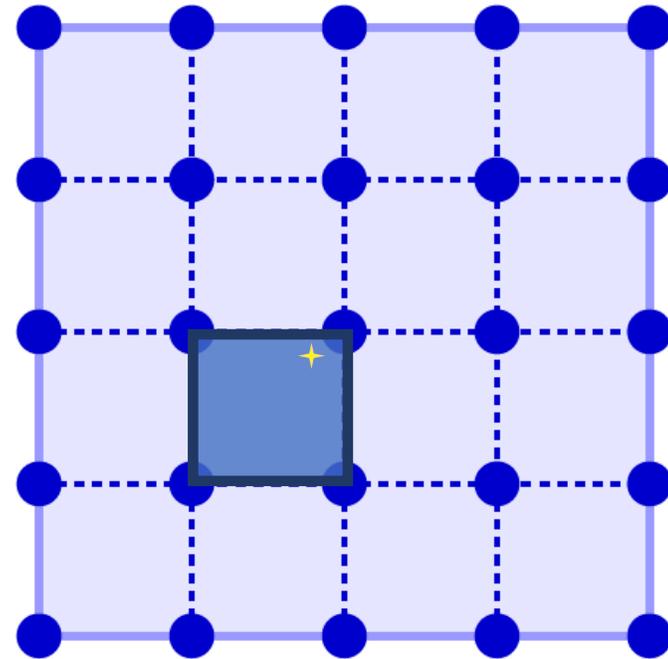


FILTERING (QUADRILATERAL PATCHES)

- Exactly the same as for 2D textures!
 - (Except we don't need the half-texel shift of 2D textures for $(s,t) \mapsto (u,v)$ conversion.)



2D Texture



Quad Patch Texture

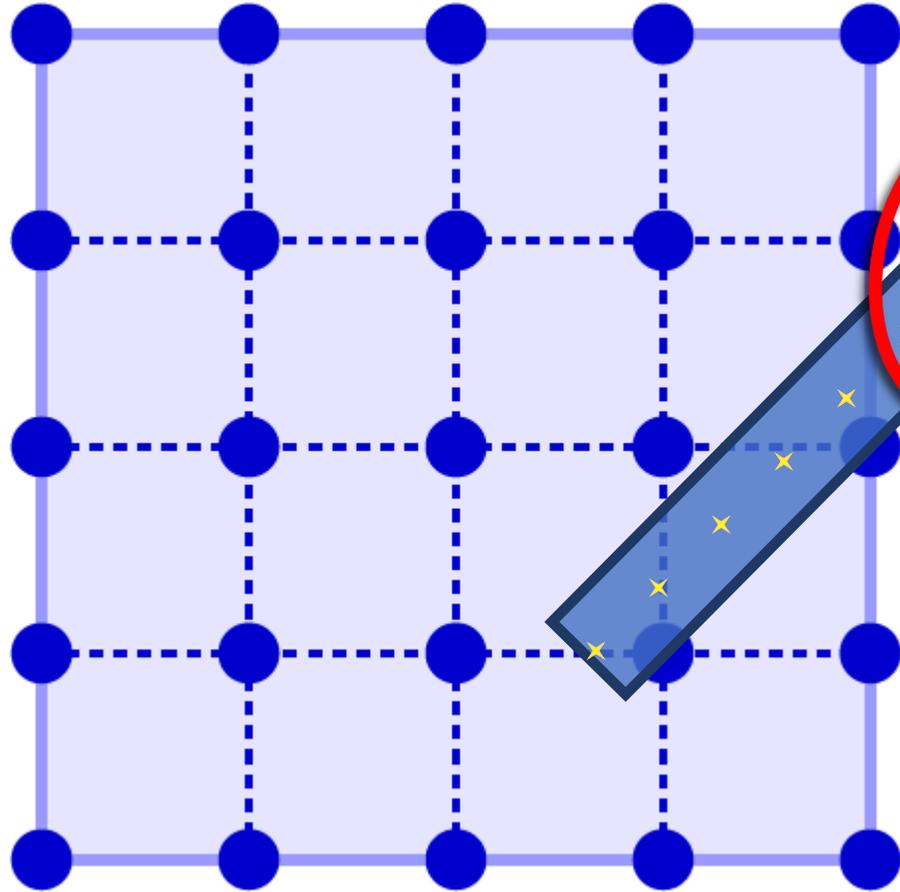
FILTERING (TRIANGULAR PATCHES)

- Triangular patches use barycentric filtering
- *Many* possible ways to tweak existing logic so that it can implement this
- Mostly, just pass 0 in some places (see paper)

ANISOTROPIC FILTERING

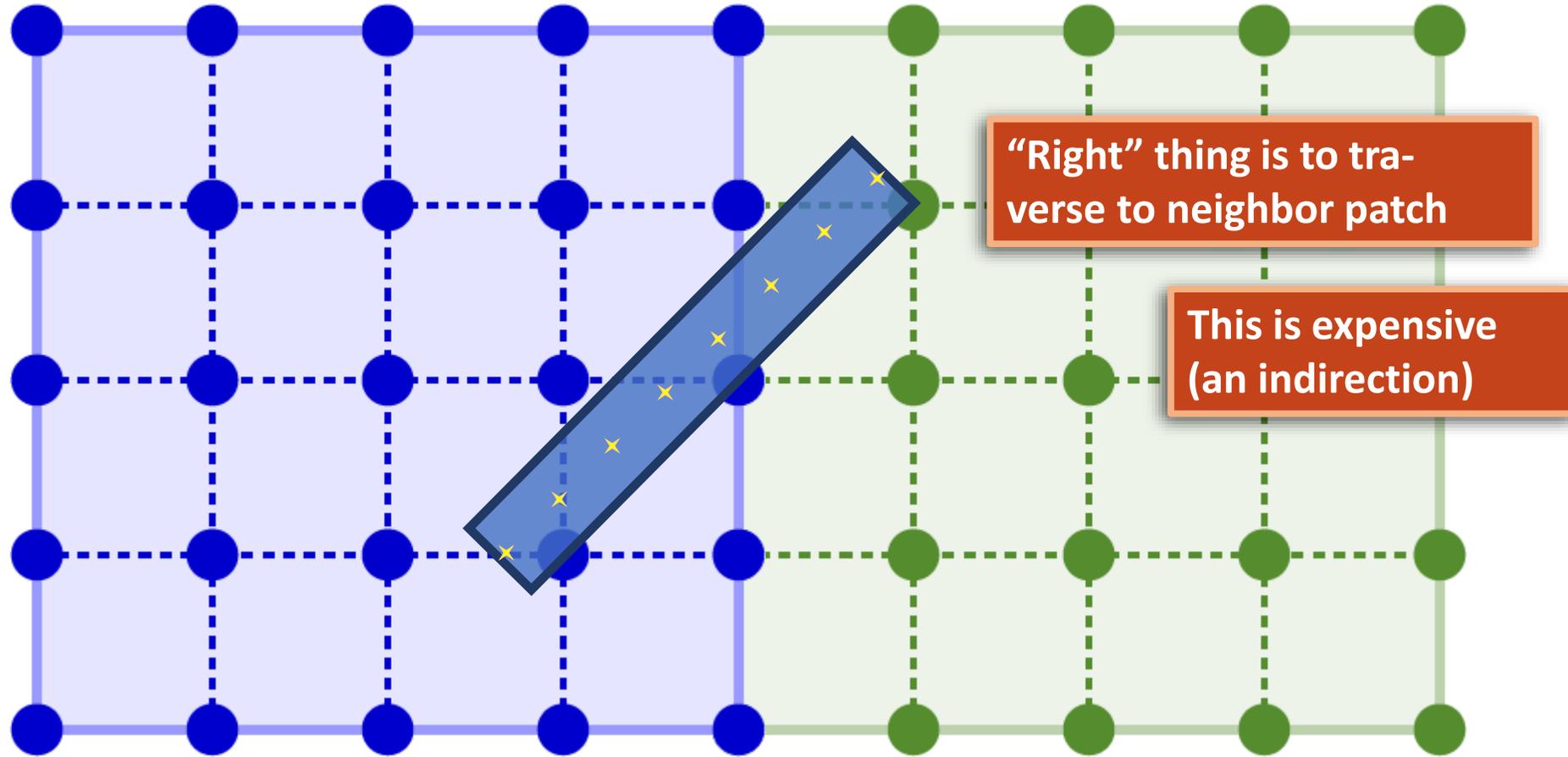
- Same process as for 2D textures
- However, we now have the chance to detect patch boundaries!

ANISOTROPIC FILTERING (PATCH BOUNDARIES)

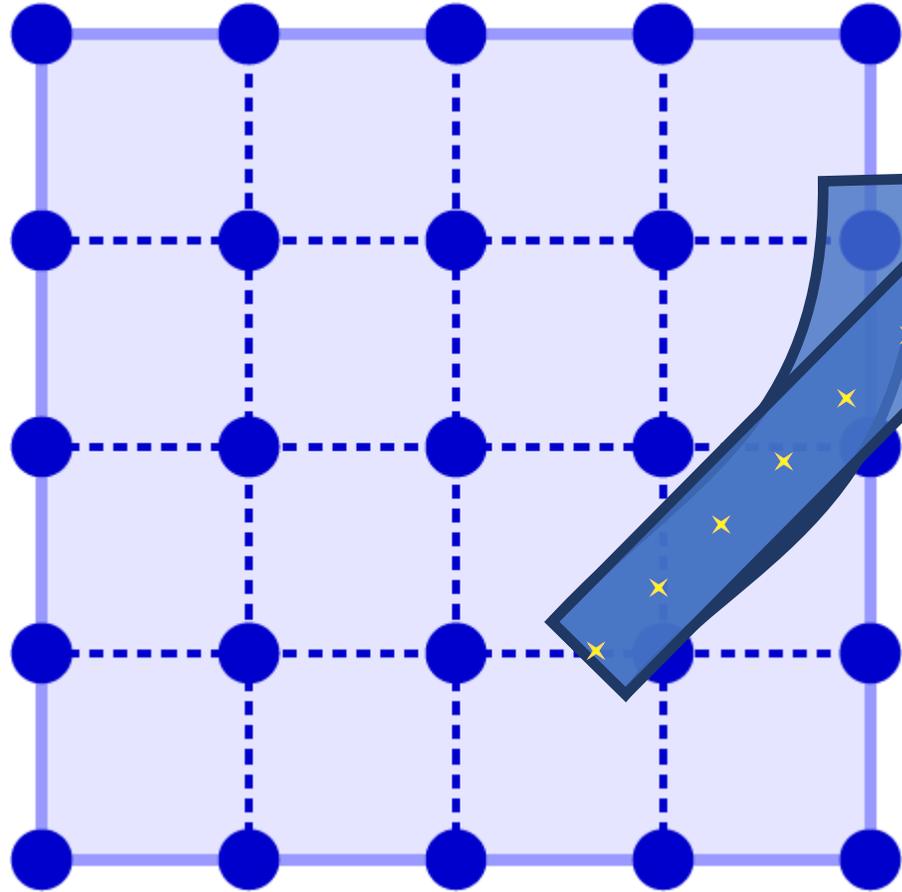


What to do about out-of-patch samples?

ANISOTROPIC FILTERING (PATCH BOUNDARIES)

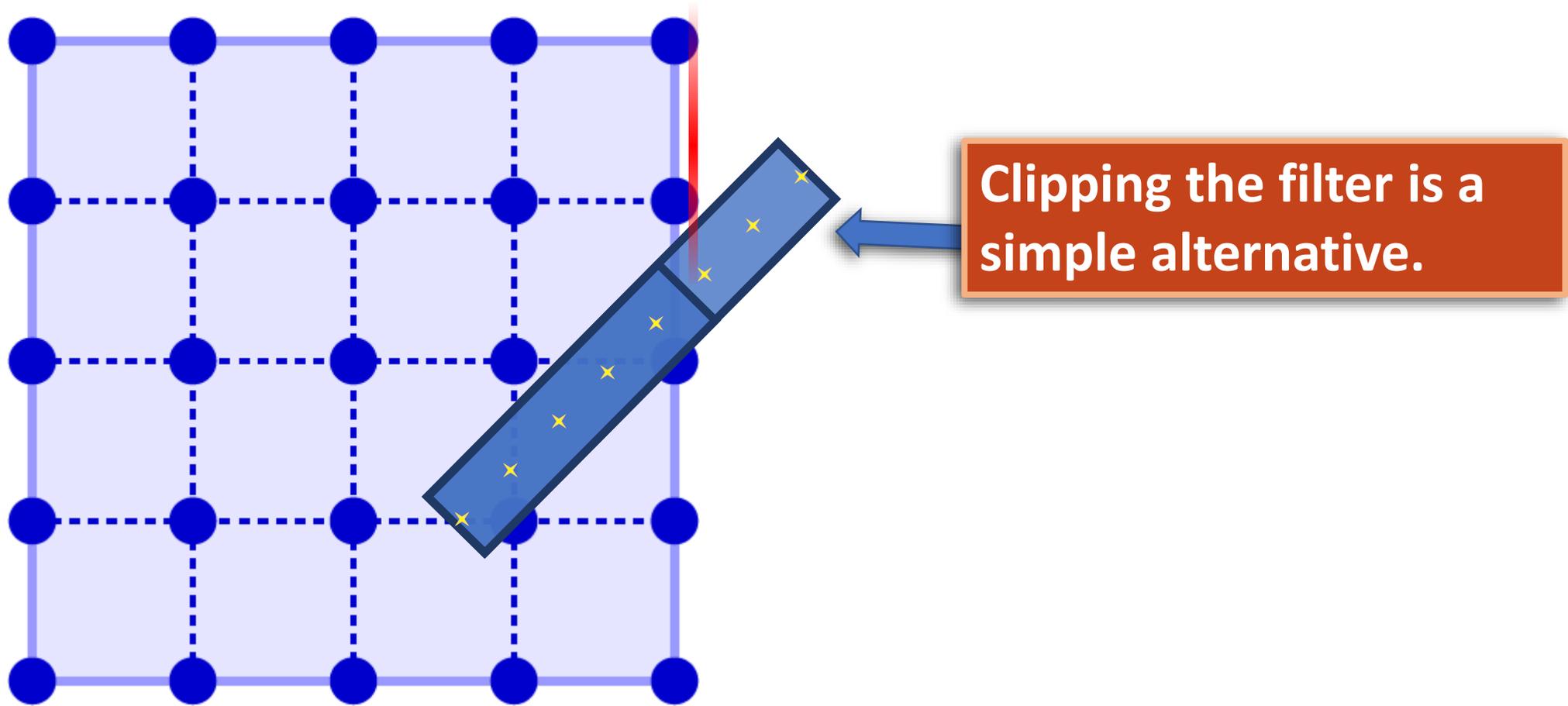


ANISOTROPIC FILTERING (PATCH BOUNDARIES)

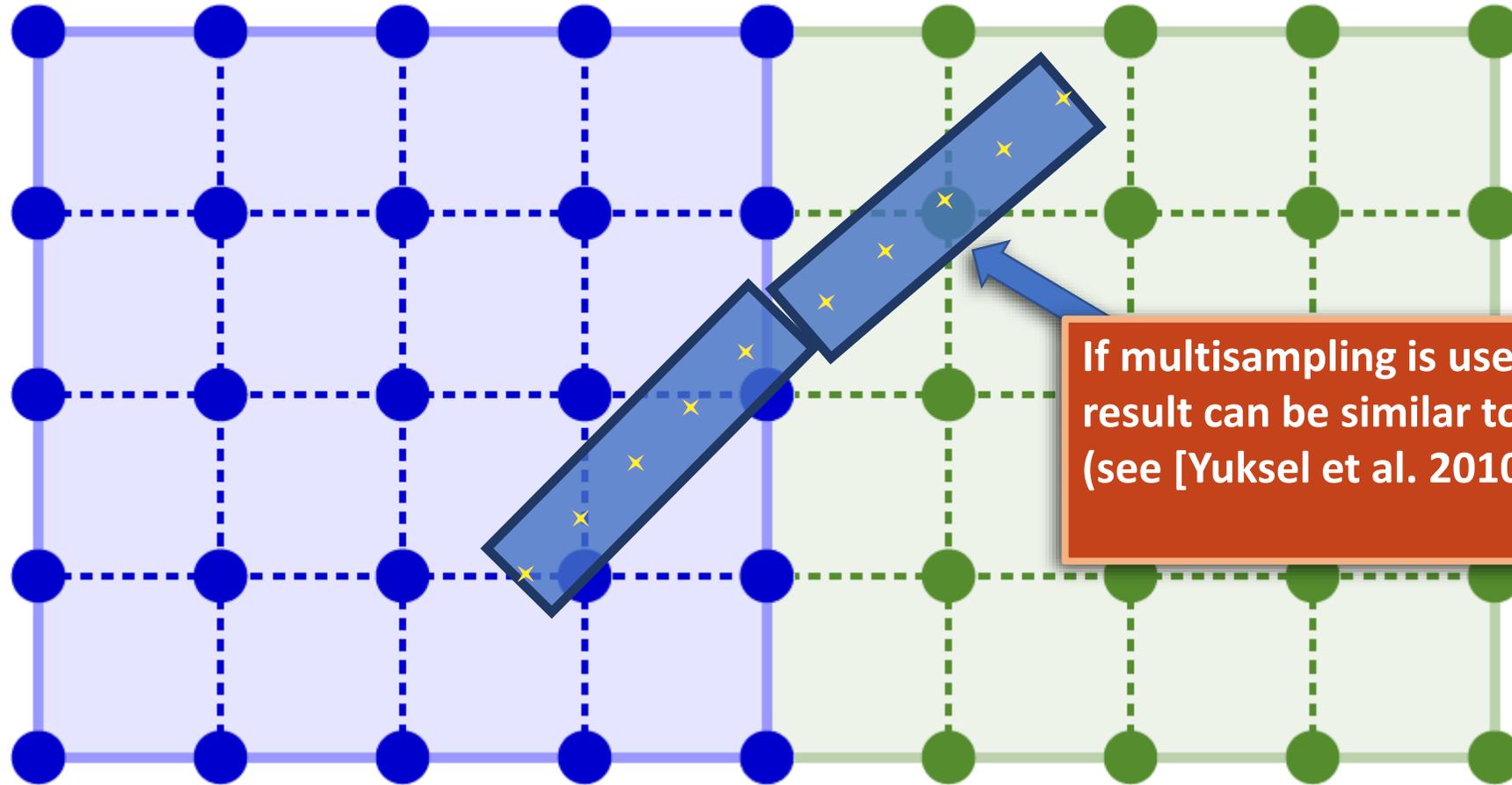


Clamping the points to the edge changes the filter shape.

ANISOTROPIC FILTERING (PATCH BOUNDARIES)



ANISOTROPIC FILTERING (PATCH BOUNDARIES)



If multisampling is used, the on-screen result can be similar to the correct filter (see [Yuksel et al. 2010], [Toth 2013])

ANISOTROPIC FILTERING COMPARISON

- Any approach is acceptable
 - None of the methods reveal the edges
 - Ground truth not expected without edge-crossing
- This is because not filtering across edges, like in 2D textures

ANISOTROPIC FILTERING COMPARISON



Patch Edges



Ground Truth

ANISOTROPIC FILTERING COMPARISON



Patch Edges



Clamped

ANISOTROPIC FILTERING COMPARISON



Patch Edges



Clipped

ANISOTROPIC FILTERING COMPARISON



Patch Edges



Ground Truth

ANISOTROPIC FILTERING COMPARISON



Patch Edges



Clamped+MSAA

ANISOTROPIC FILTERING COMPARISON



Patch Edges



Clipped+MSAA

PROOF OF CONCEPT

- Implemented all algorithms in GPU renderer

PROOF OF CONCEPT



LIZARD
(1751 patches)



NYRA
(15124 patches)



ALIEN
(5488 patches)



HEAD
(9094 patches)

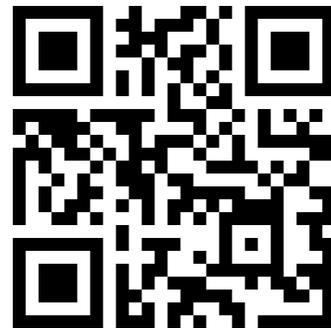
CONCLUSION

- Implementing Mesh Colors requires only minimal changes to existing GPU hardware
- Takeaway for vendors: go implement it already! 😊

QUESTIONS



Utah Graphics



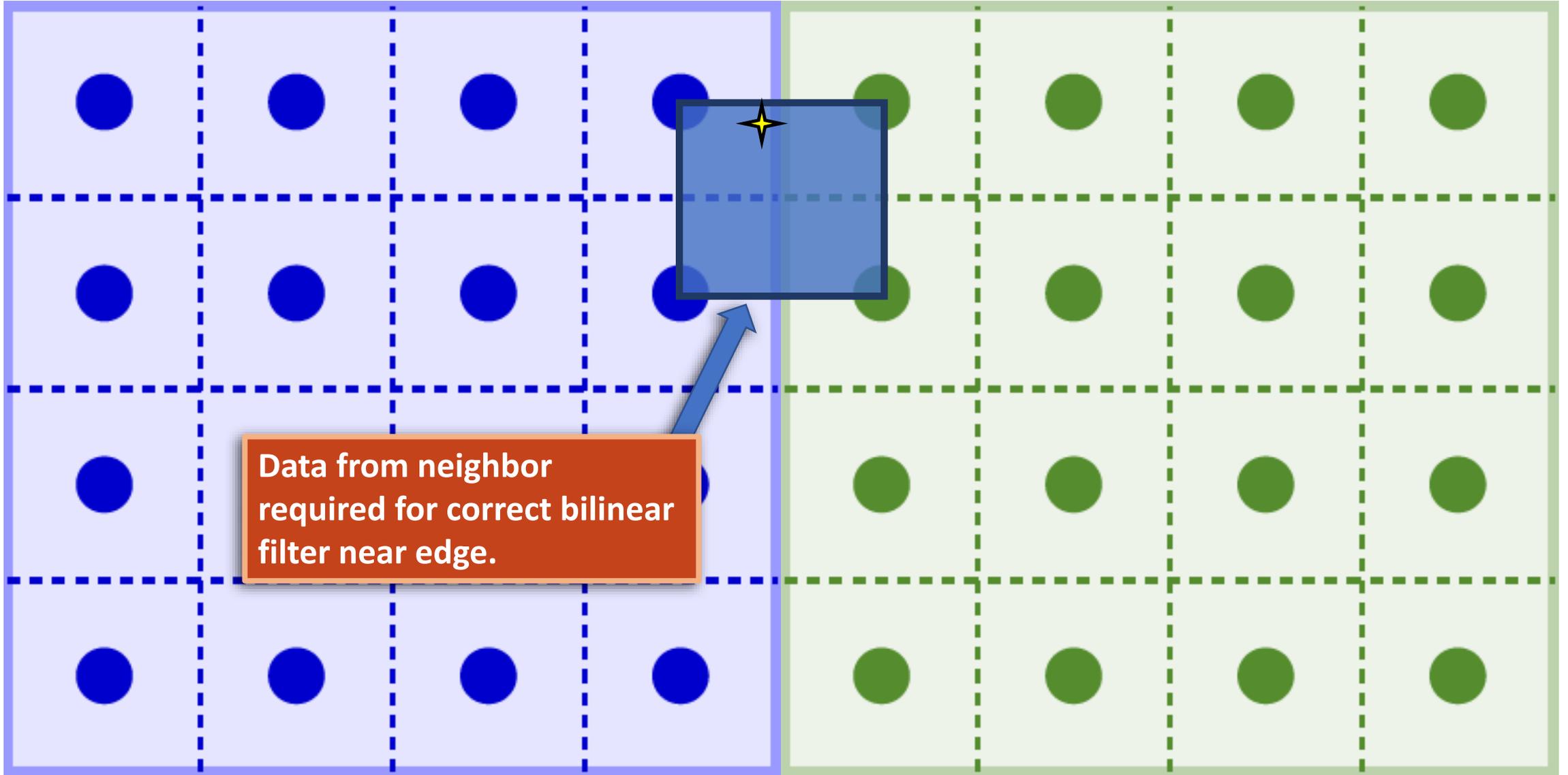
[preprint](#)



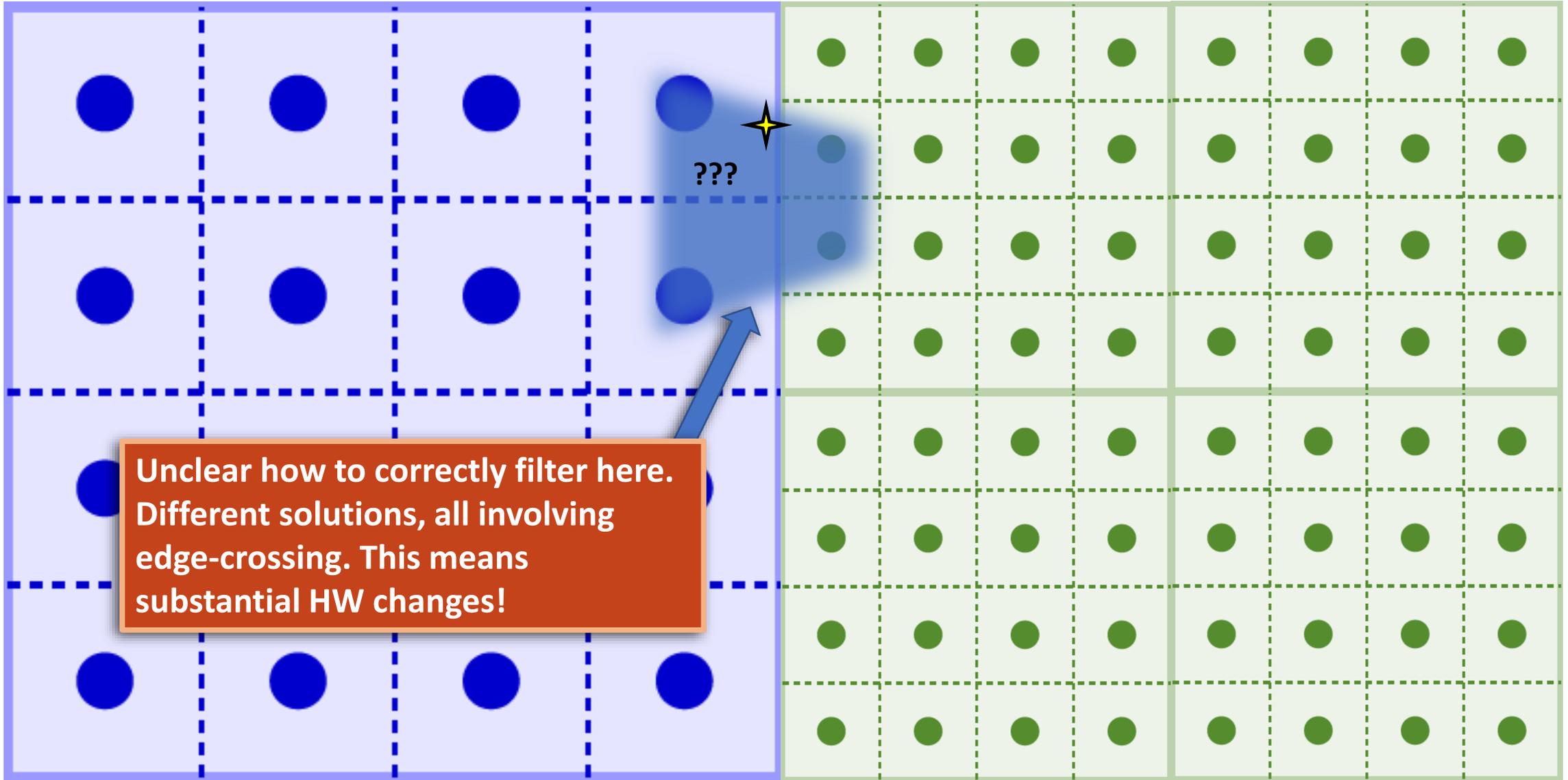
WHY IS IT HARDER WITH PTEX?

- Ptex and Mesh Colors have minor theoretical differences (they are duals of each-other), but this leads to significant difficulty in-practice.
- Main problem is that edge-crossing is *required* for correct filtering in Ptex.

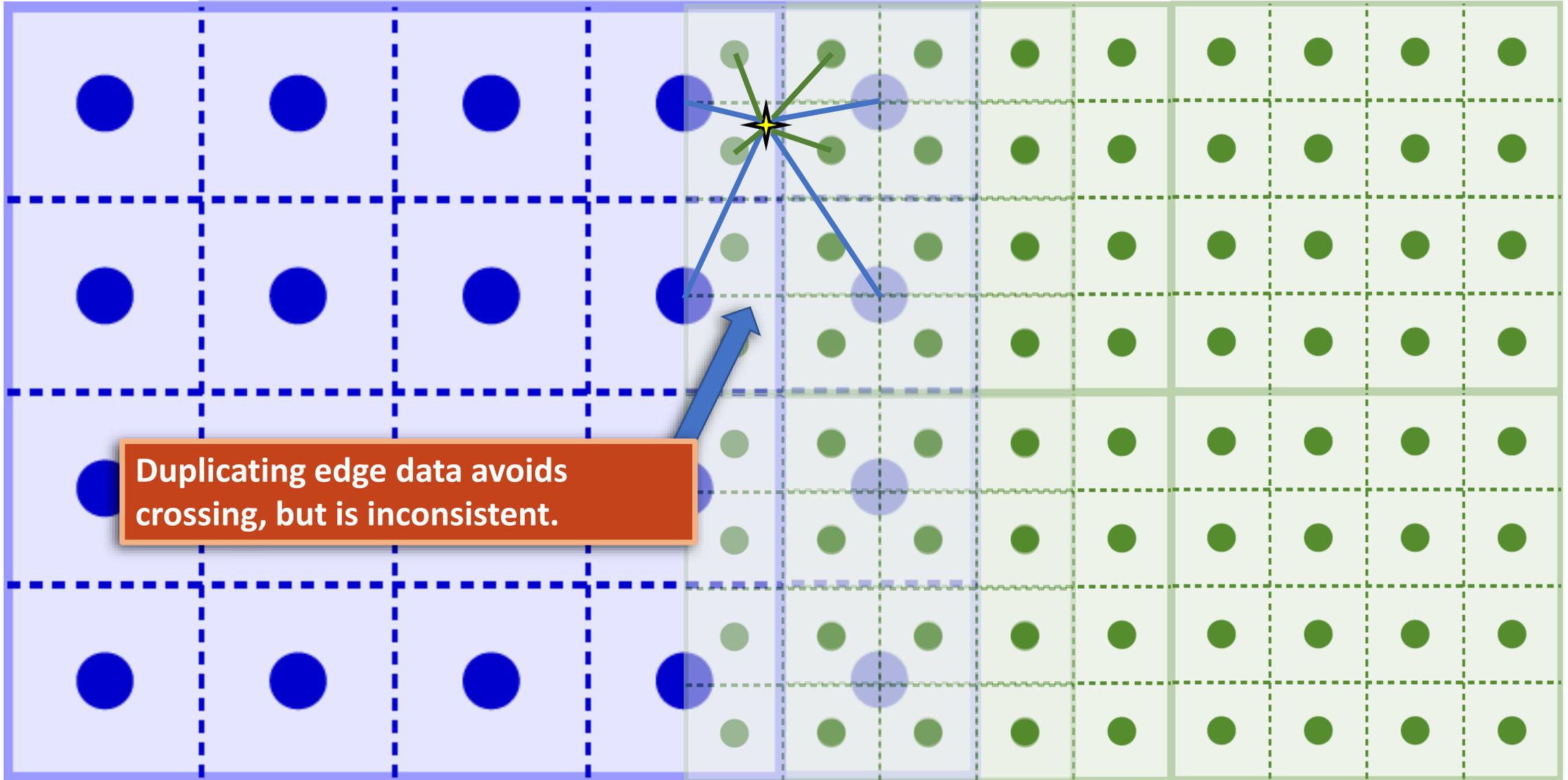
WHY IS IT HARDER WITH PTEX?



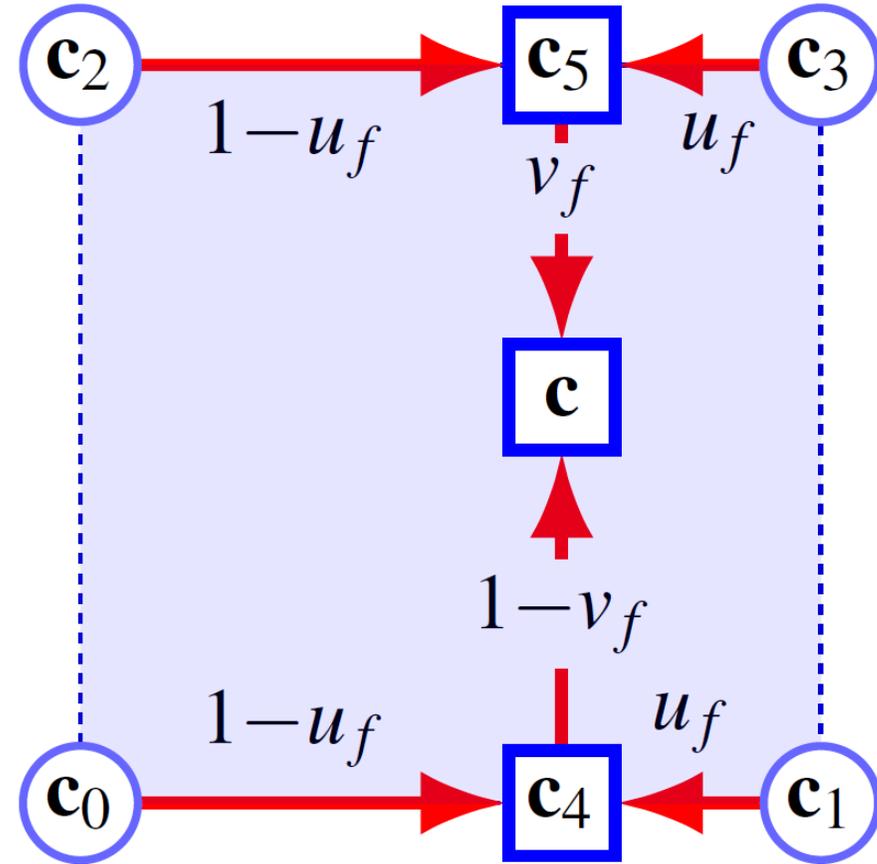
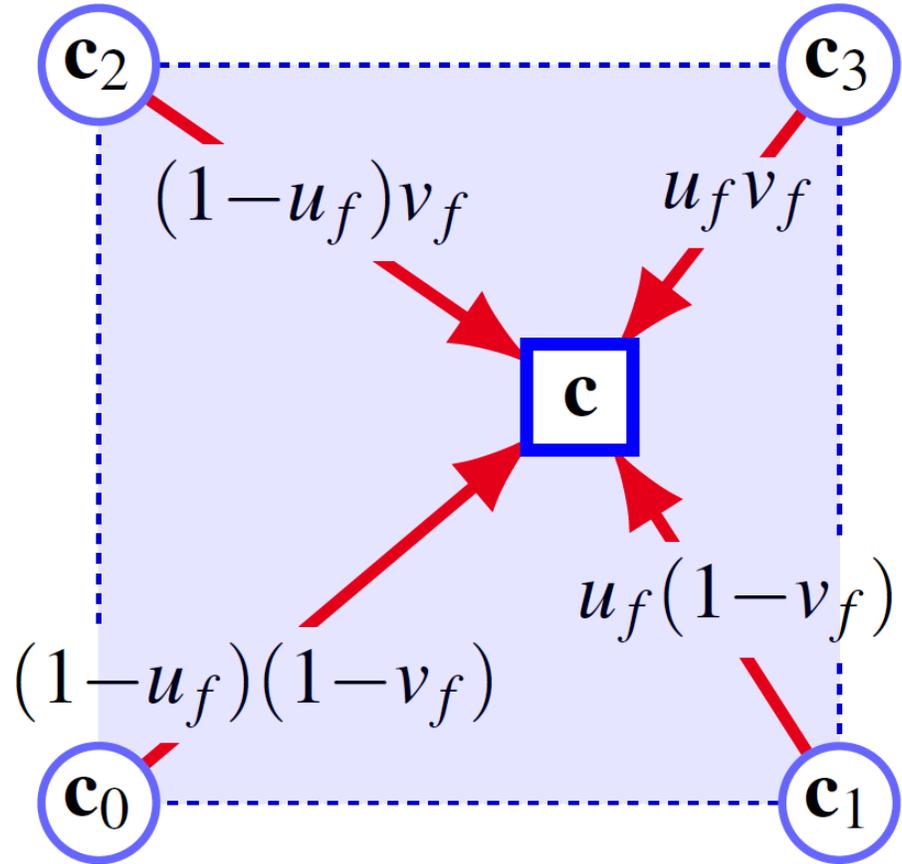
WHY IS IT HARDER WITH PTEX?



WHY IS IT HARDER WITH PTEX?



FILTERING (QUADRILATERAL PATCHES)



FILTERING (TRIANGULAR PATCHES)

