Improved Deep Image Compositing Using Subpixel Masks

Jonathan Egstad
DreamWorks Animation
jonathan.egstad@dreamworks.com

Mark Davis
DreamWorks Animation
mark.davis@dreamworks.com

Dylan Lacewell
DreamWorks Animation / NVIDIA
dlacewell@nvidia.com

OpenEXR 2.0 Deep Images

Deep images are 2-D images with the capacity to store multiple color & depth samples (*deep sample)* at each pixel location (*deep pixel*)

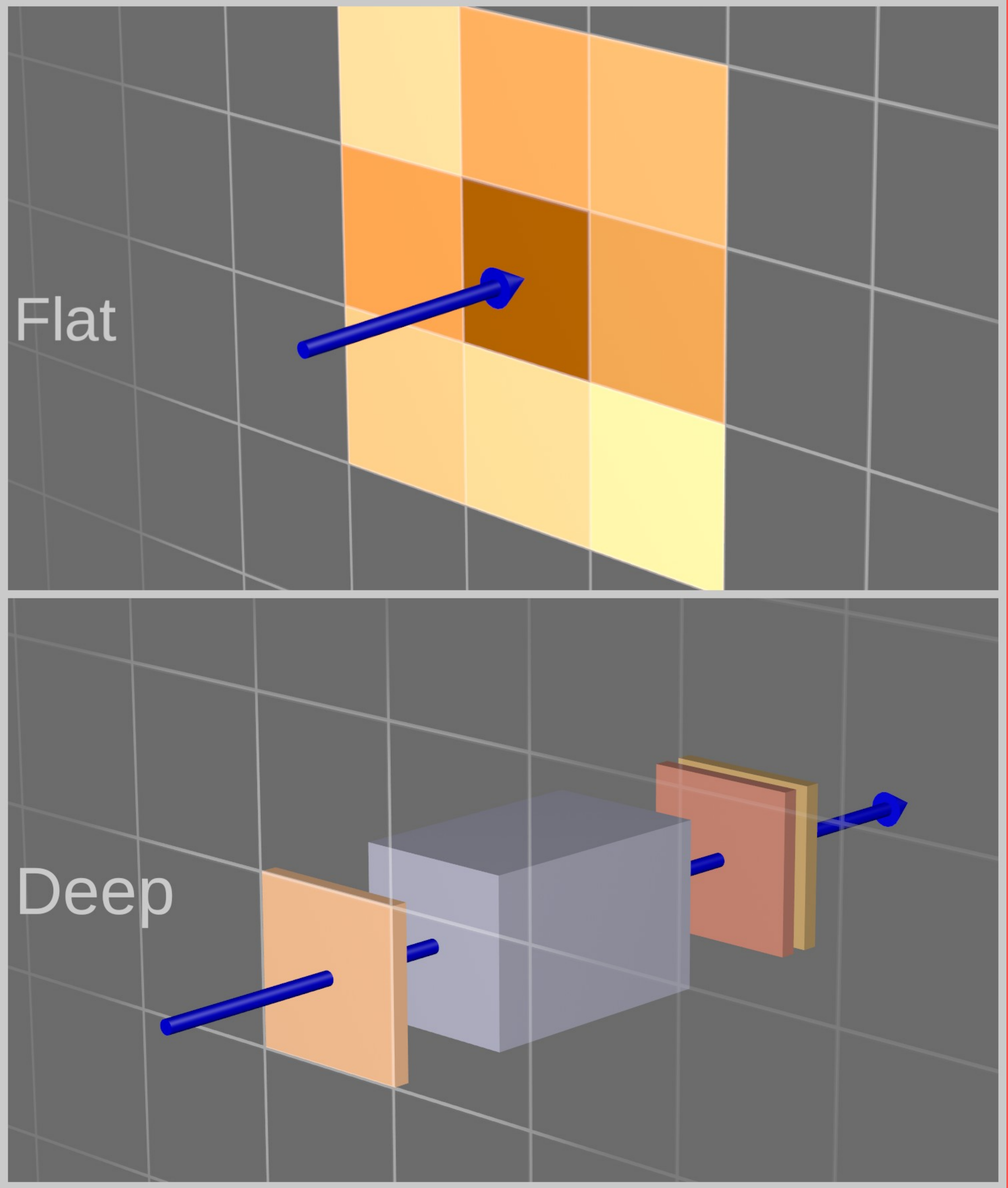
Production facilities developed ad-hoc deep workflows over the years, often based on deep-shadow techniques

Weta Digital's deep workflow was formalized and integrated into OpenEXR 2.0

For more information on OpenEXR 2.0 deep images see:

Hillman, P. 2013. **The Theory of OpenEXR Deep Samples**
http://www.openexr.com/TheoryDeepPixels.pdf

Kainz, F. 2013. **Interpreting OpenEXR Deep Pixels**
http://www.openexr.com/InterpretingDeepPixels.pdf



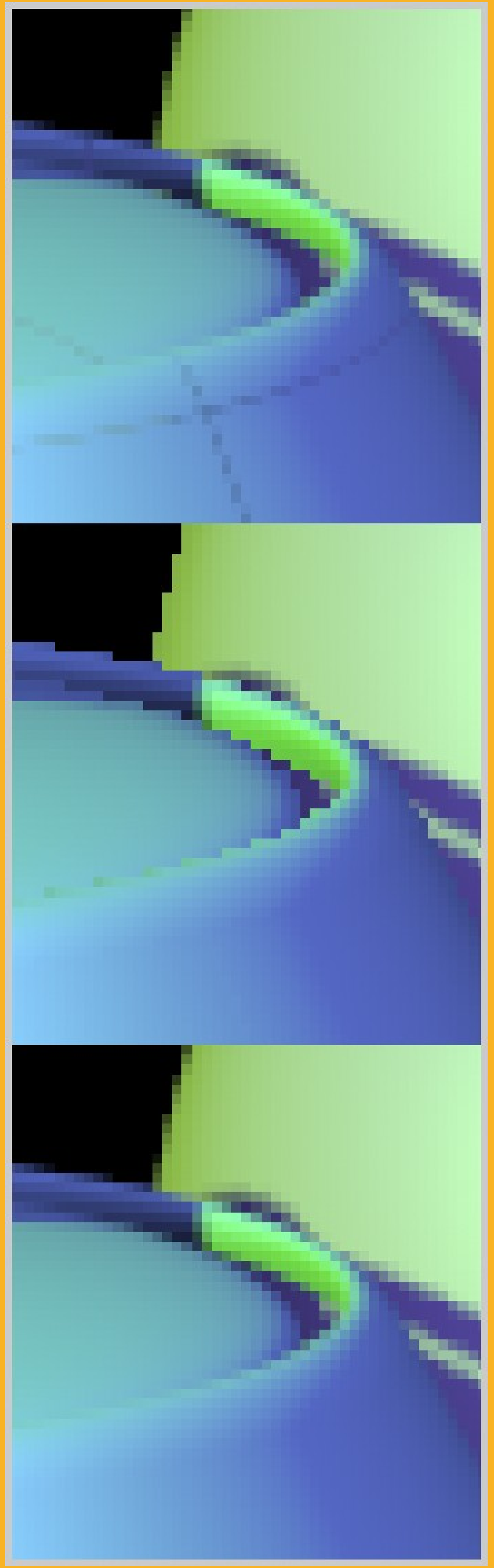
OpenEXR 2.0 Workflow Challenges

Works best when combining multiple volumetric renders

Not as well when combining volumetrics with one or more hard-surface renders

Challenges With Hard-Surface Samples

- Lack of subpixel spatial information to resolve overlaps
- Lack of spatial info prevents accurate pixel-filtering
- Transparency and pixel-coverage are combined and cannot be separated
- Intersections between opaque hard-surfaces will often exhibit aliasing



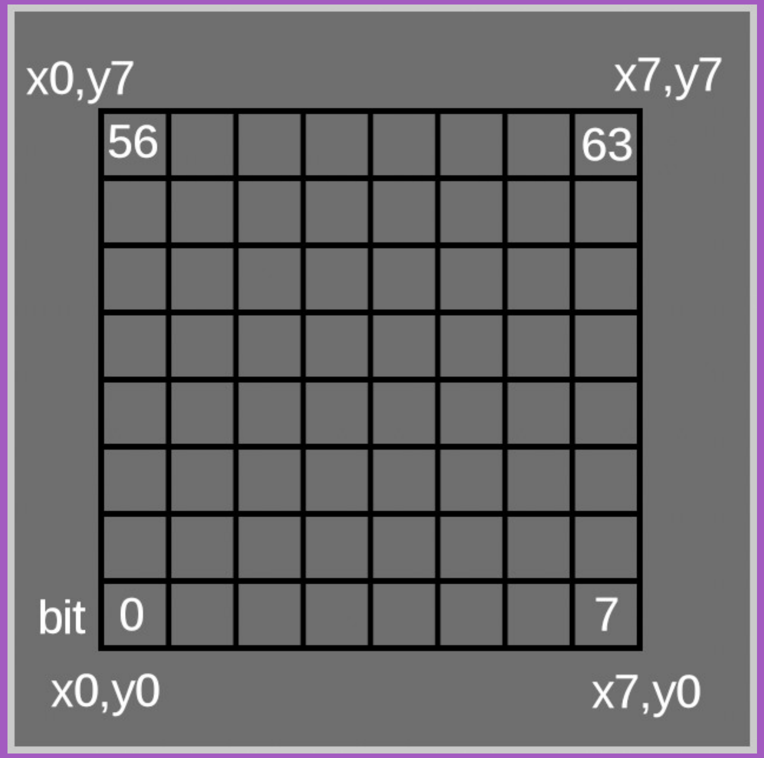
Subpixel Masks

Generated by rendering algorithm

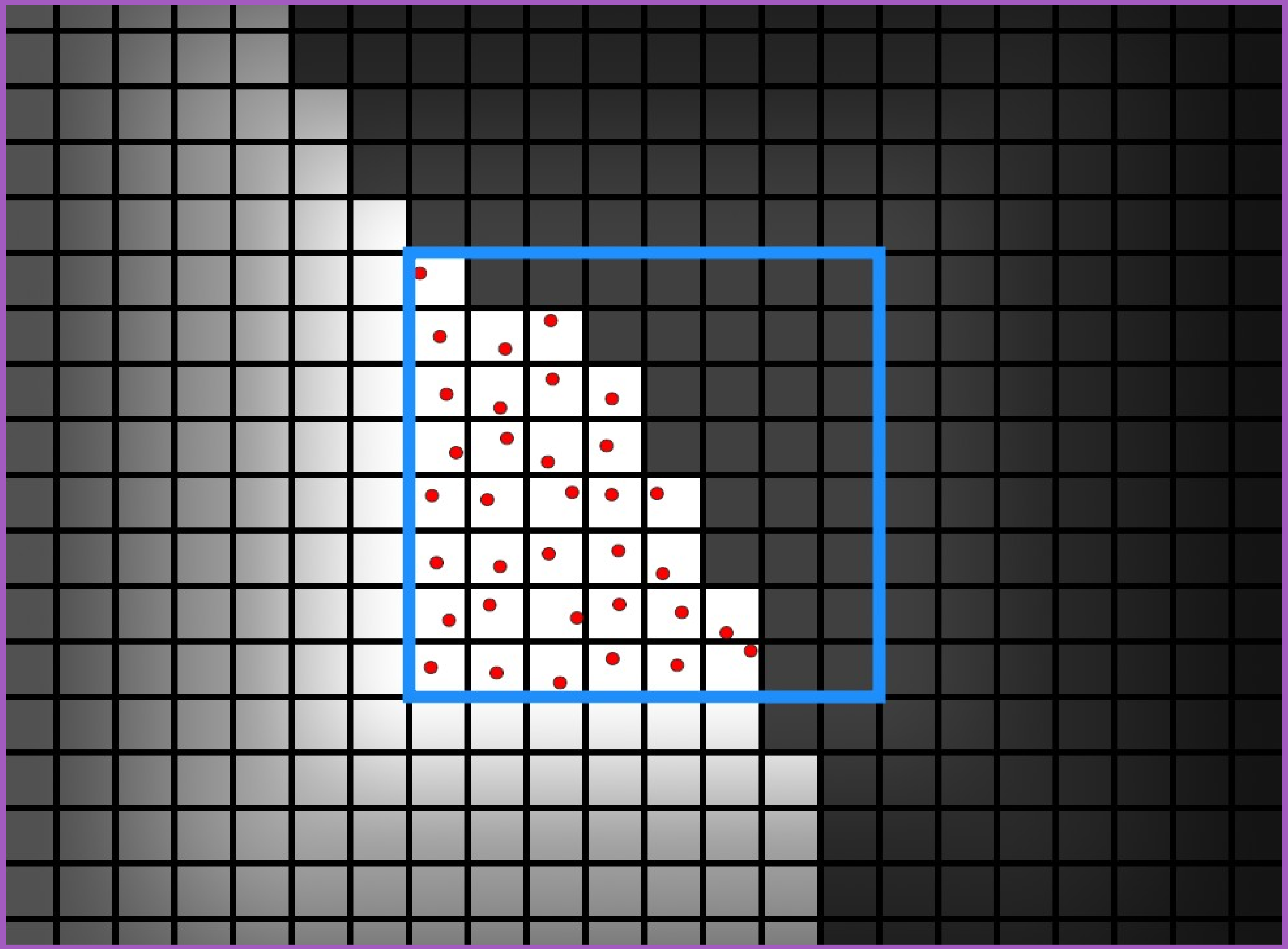
One mask per deep sample

Provides subpixel xy locations

Must ensure samples are stratified to avoid cracks



Pattern/Bit Orientation



Subpixel Masks

Pros:

Resolves overlaps at subpixel level

Separates pixel-coverage from transparency

Accurate pixel-filtering is possible

Mask patterns will often compress well

Cons:

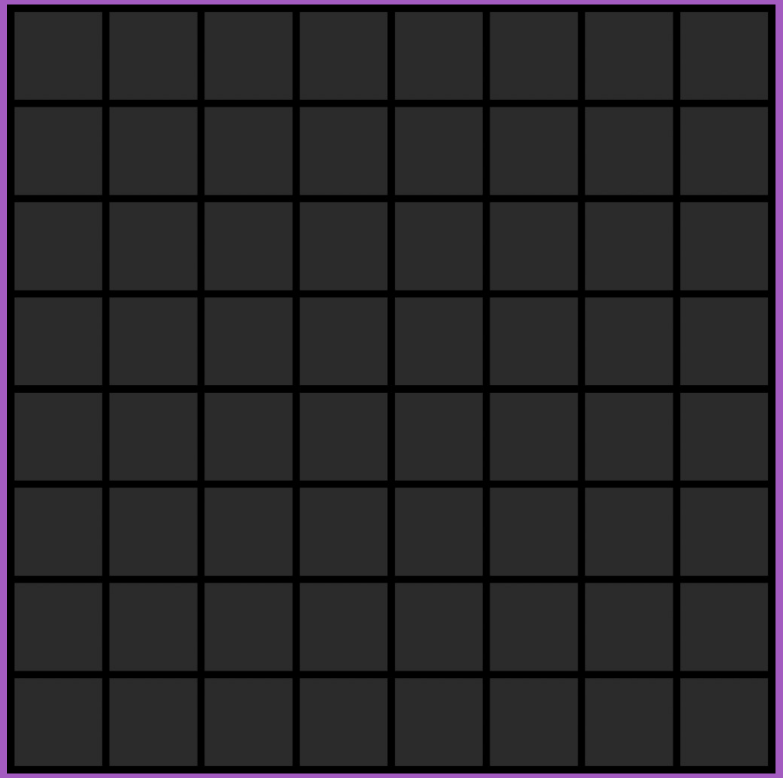
Fixed mask size for all samples

Separate pixel-coverage / transparency can complicate some compositing operations

Increased deep sample storage cost (can be offset by reduction in sample count)

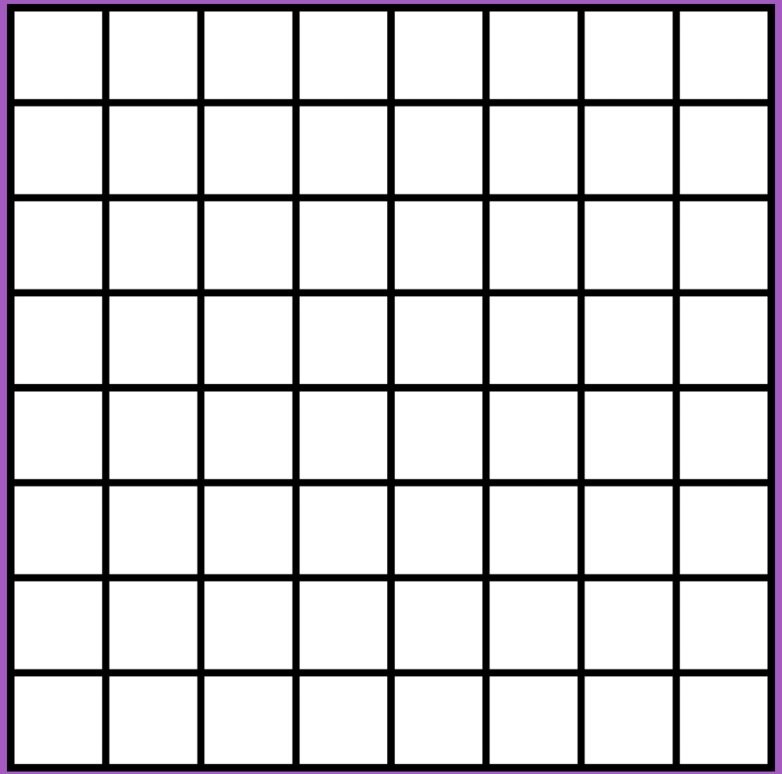
Bit-pattern storage not natively supported by OpenEXR 2.0

Subpixel Mask – Special Cases



All bits off - zero coverage
Indicates lack of subpixel information

All bits on - full coverage
Allows per-subpixel evaluation to be skipped (common for volumetric content)



Subpixel Mask Overlap Resolution

Step through each subpixel index and flatten the samples with that bit enabled

Each subpixel index can yield a unique set of enabled samples

Integrating the flattened results of a region of subpixels produces the final anti-aliased pixel

Overlap Resolution Example

Red and blue surfaces cover 50% of pixel

Both are opaque (alpha == 1.0)

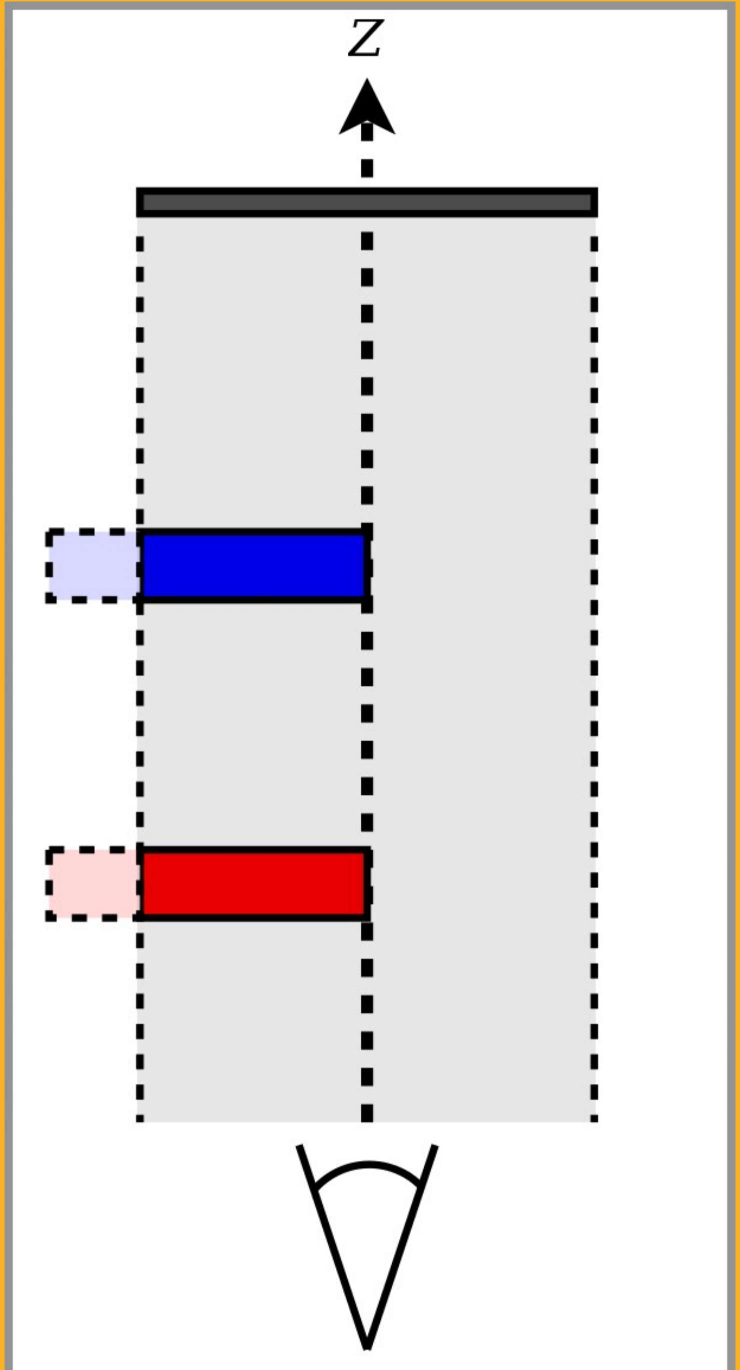
A:

Red surface completely occludes blue surface

B:

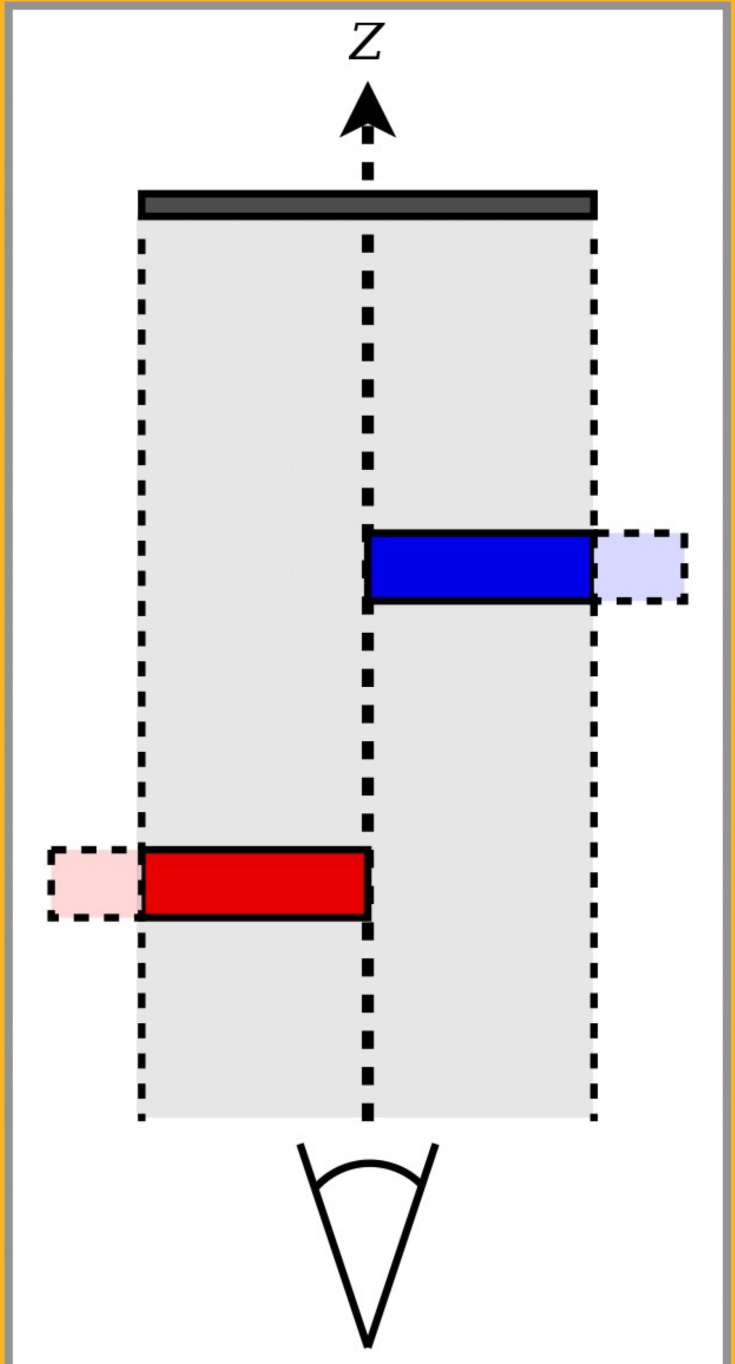
Red and blue surfaces do not occlude each other

A



Overlapping surfaces

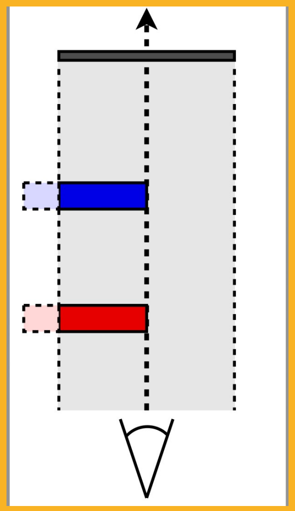
B



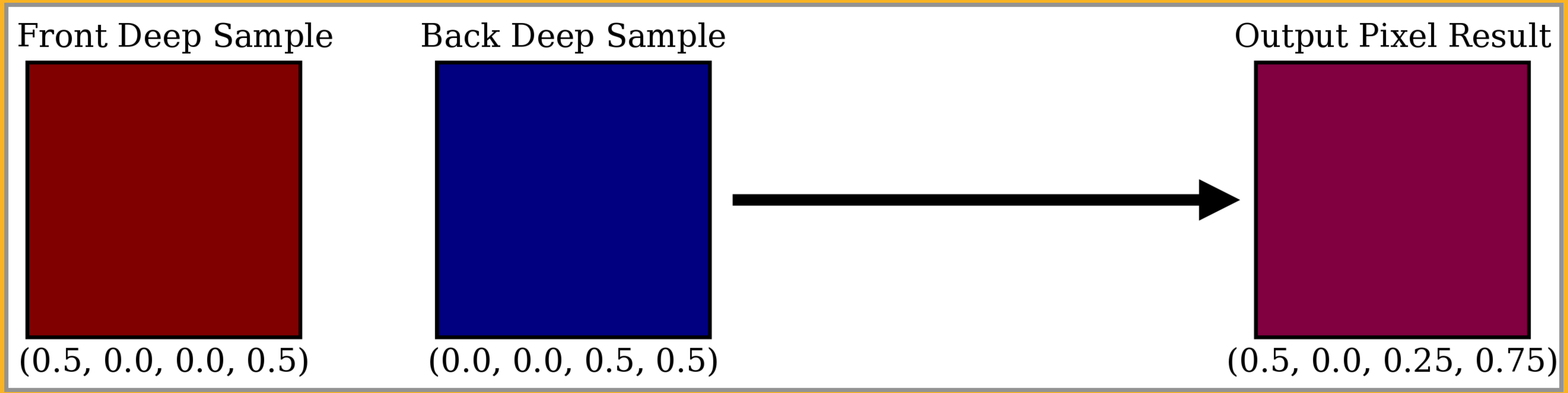
Common-edge surfaces

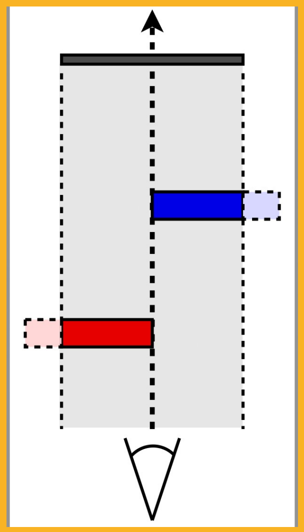
Overlap Resolution: Current OpenEXR Method

Both A and B produce the same incorrect result

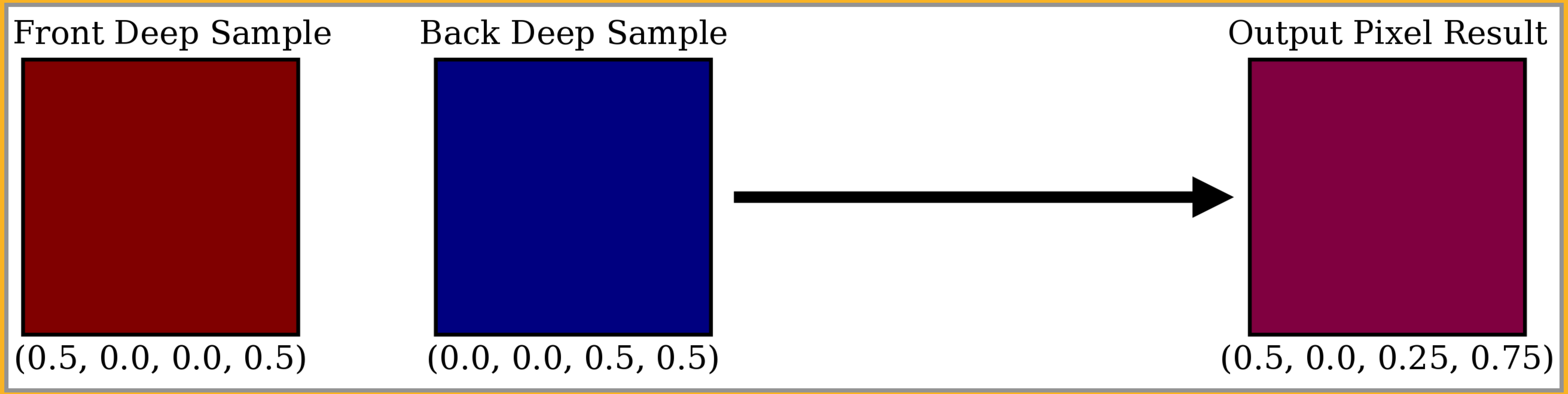


A



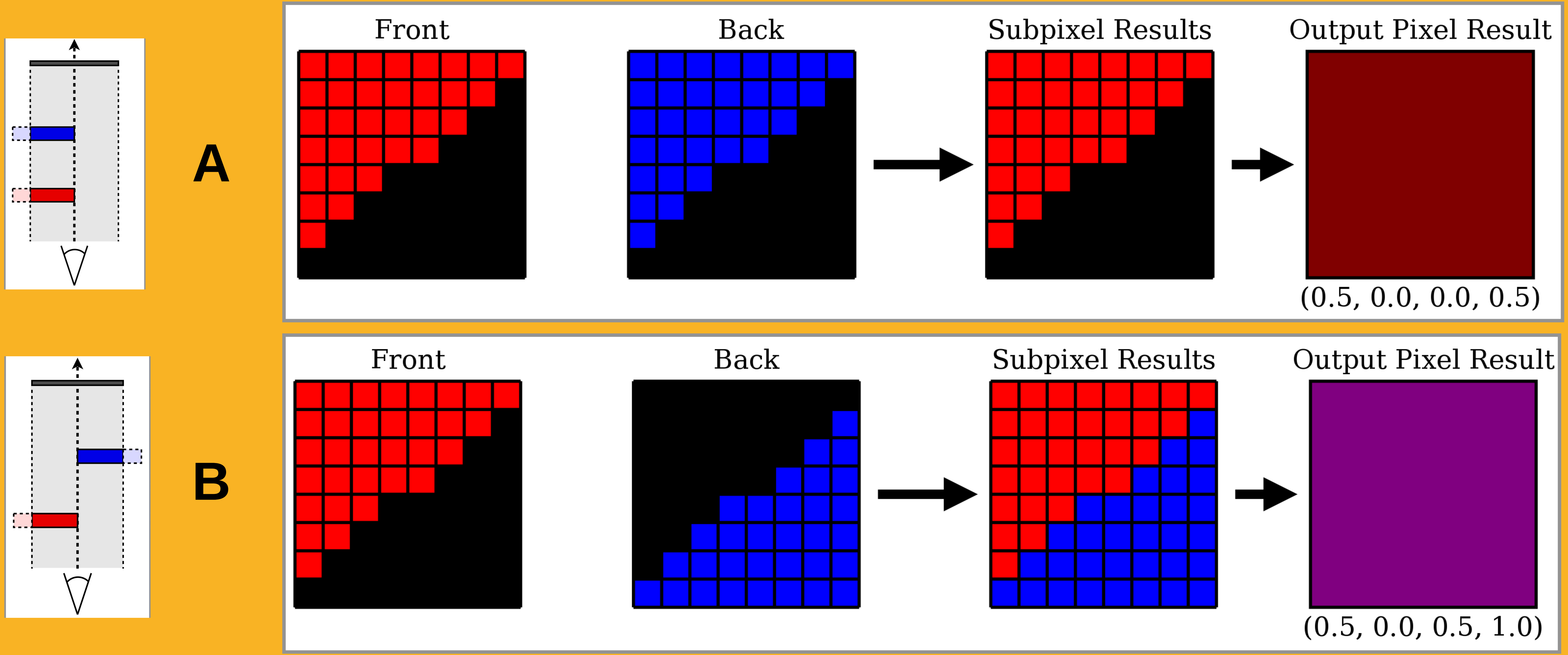


B

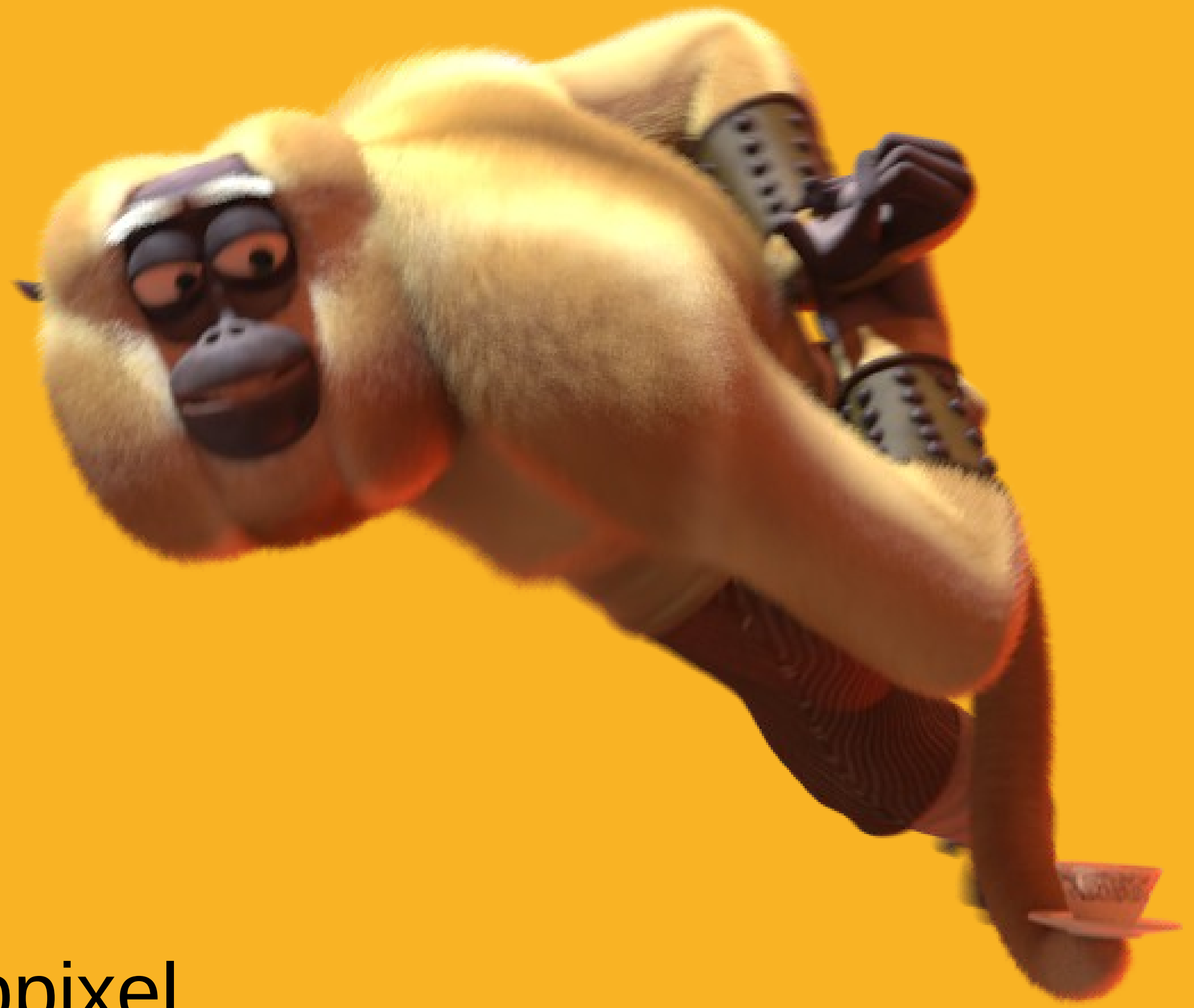


Overlap Resolution: Improved OpenEXR Method

A and B produce the correct result using subpixel masks



Sample Collapsing



Diverse sample set from the subpixel grid

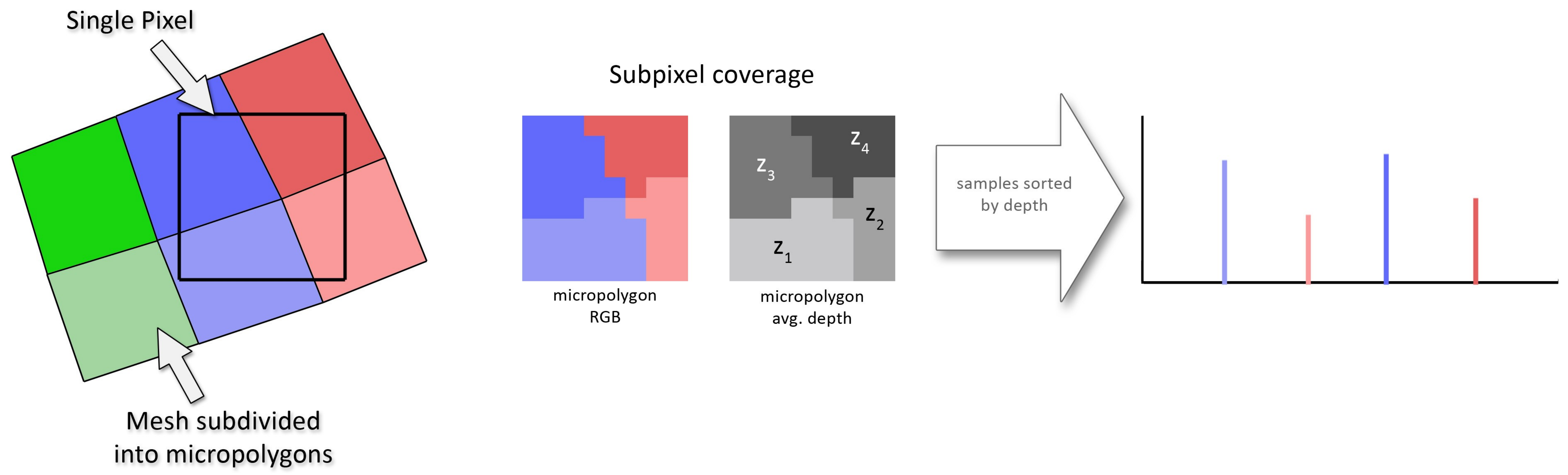
Varying primitive/surface ID, color, normal, depth

Worst case scenario: a unique deep sample per subpixel

Must retain the integrity of the uncollapsed data

Sample Collapsing: Static Micropolygon Example

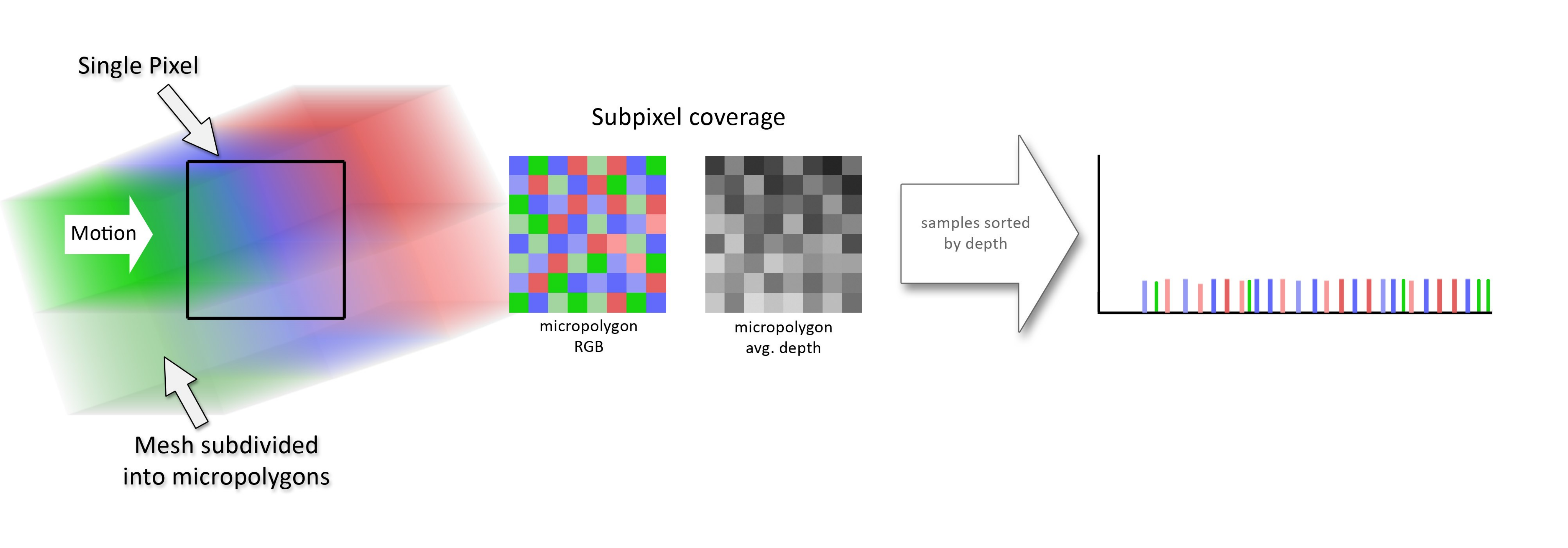
Naively inserting shaded micro-polygons



Sample Collapsing: With Motion Blur

Motion blur and/or highly-tessellated geometry increases sample count

A ray-tracer would exhibit similar problems.



Sample Collapsing: Clustering

Determine which samples can be collapsed into clusters
No single correct way to do this

- **Geometry ID**
- **Depth**
- **Color**
- **Normal**

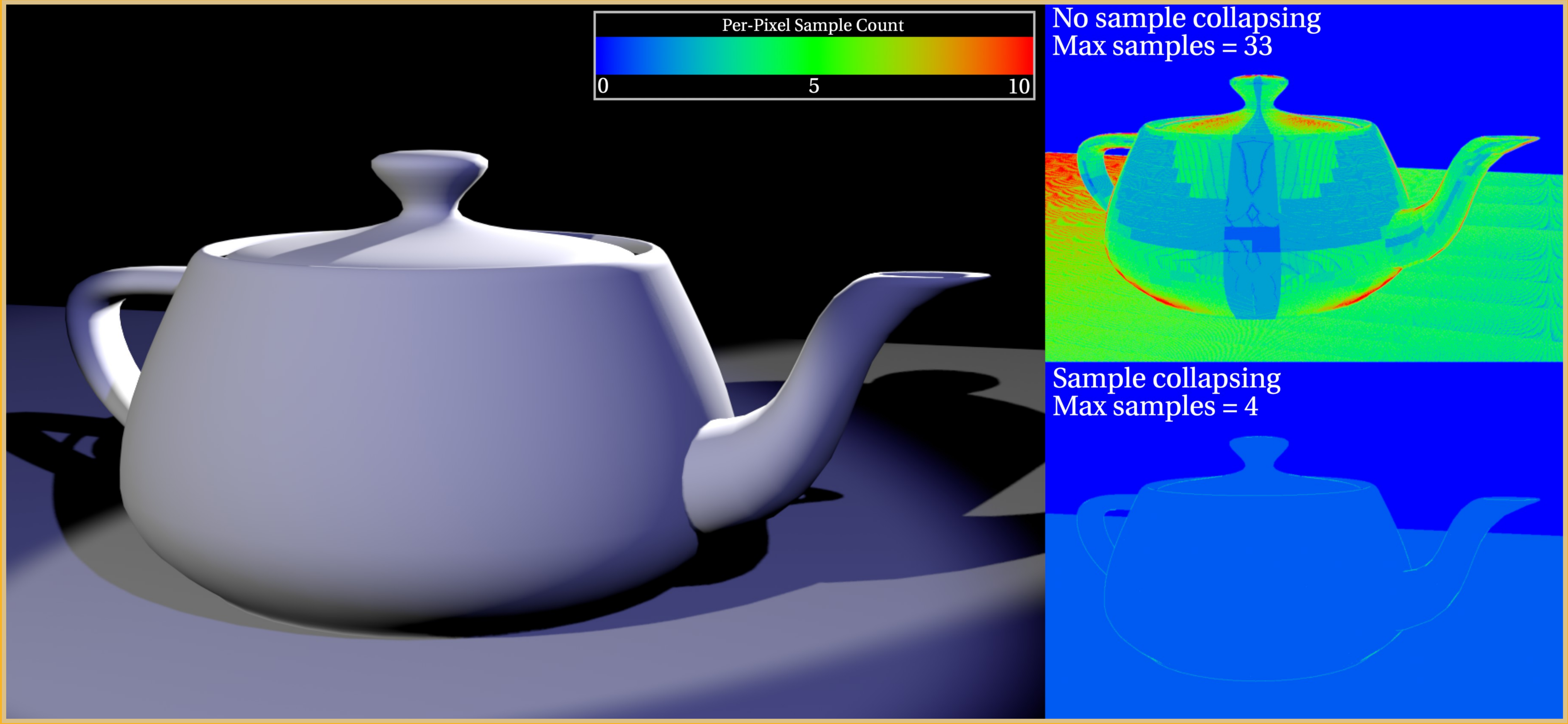
User controlled cluster limit

For each cluster:

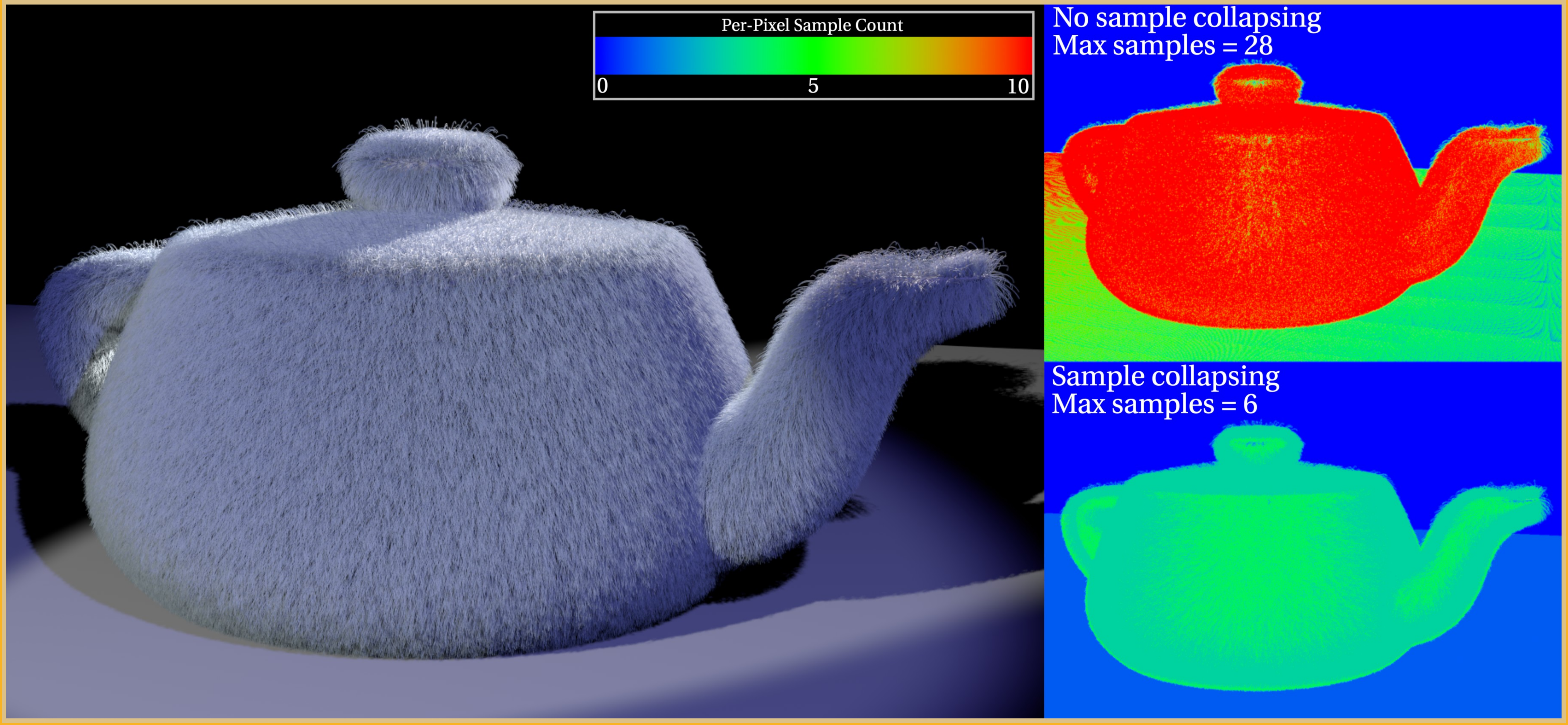
- Flatten the samples
- OR together the subpixel masks
- Determine the min/max Z

Each cluster becomes a single sample in the OpenEXR 2.0 file

Sample Collapsing: Results



Sample Collapsing: Results



Surface Flags

Per-sample flags provide additional information about a deep sample

Backwards-compatible - value 0x00 (0.0) indicates a 'legacy' OpenEXR 2.0 deep sample

Proposed flags:

- **0x01 (1.0) Hard-Surface -** use linear interpolation
- **0x02 (2.0) Matte** - act as a holdout to samples behind

Hard-Surface Flag:
Why Bother With Linear Interpolation?

By-product of collapsing samples (a Z range is created)

Shared Z for all subpixels causes aliased intersections (binary Z comparison)

Log interpolation intended for volumetric content

Log interpolation fails when alpha == 1.0, a common case with hard-surfaces

Hard-surface flag tells the flattener to use linear interpolation

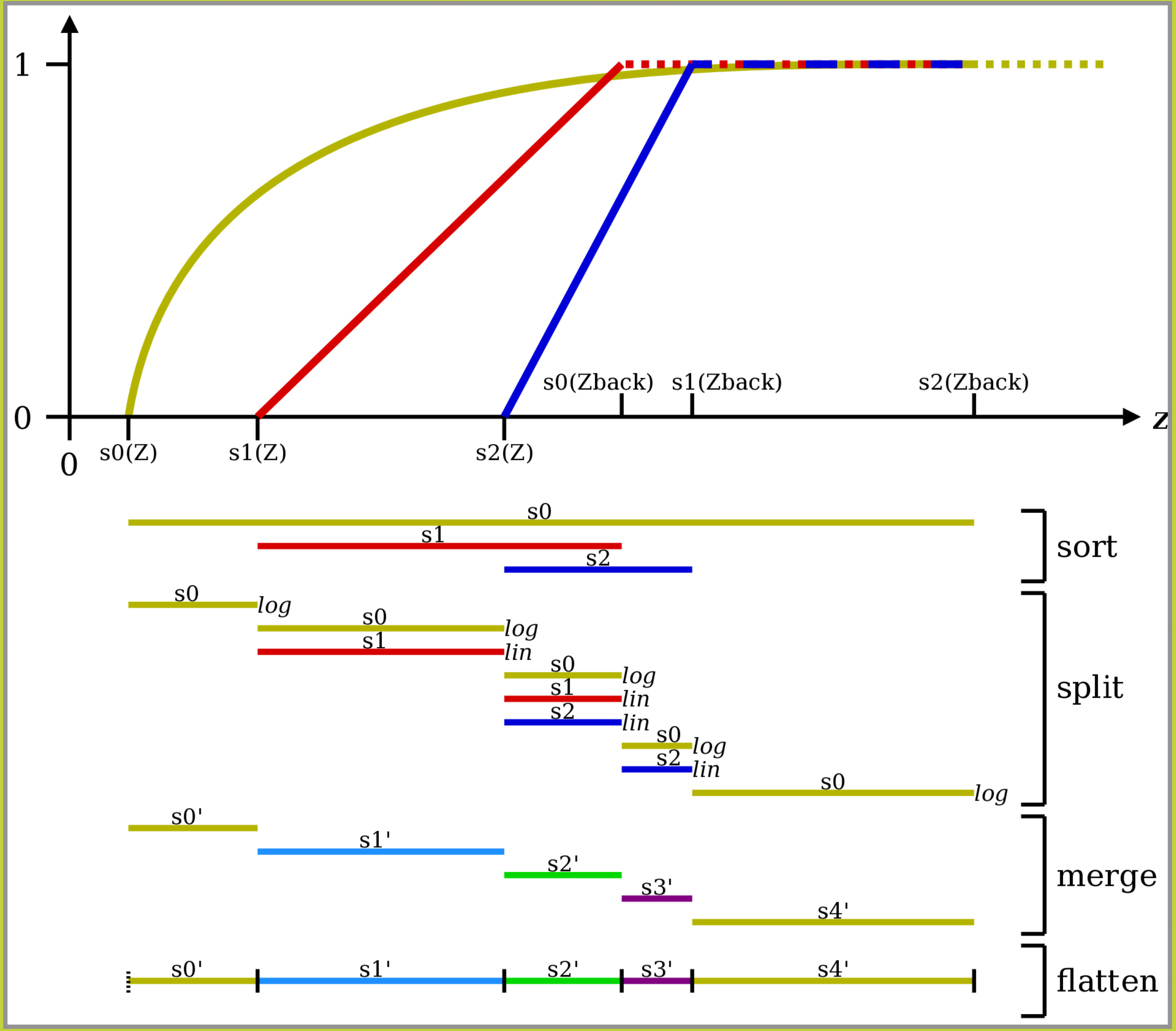
Linear interpolation allows Z-blending of hard-surface intersections to reduce aliasing

Linear/Log Interpolation During Flattening

Find overlapping samples and determine mix of surface types:

Log/Log: OpenEXR log-interpolation

Log/Lin & Lin/Lin: Numerical vs. analytic



Matte Flag – Holdout Management

Matte holdouts are performed in the rendering algorithm

An identifier indicates special-case handling of sample merge operation

Alpha channel is held out along with color channels

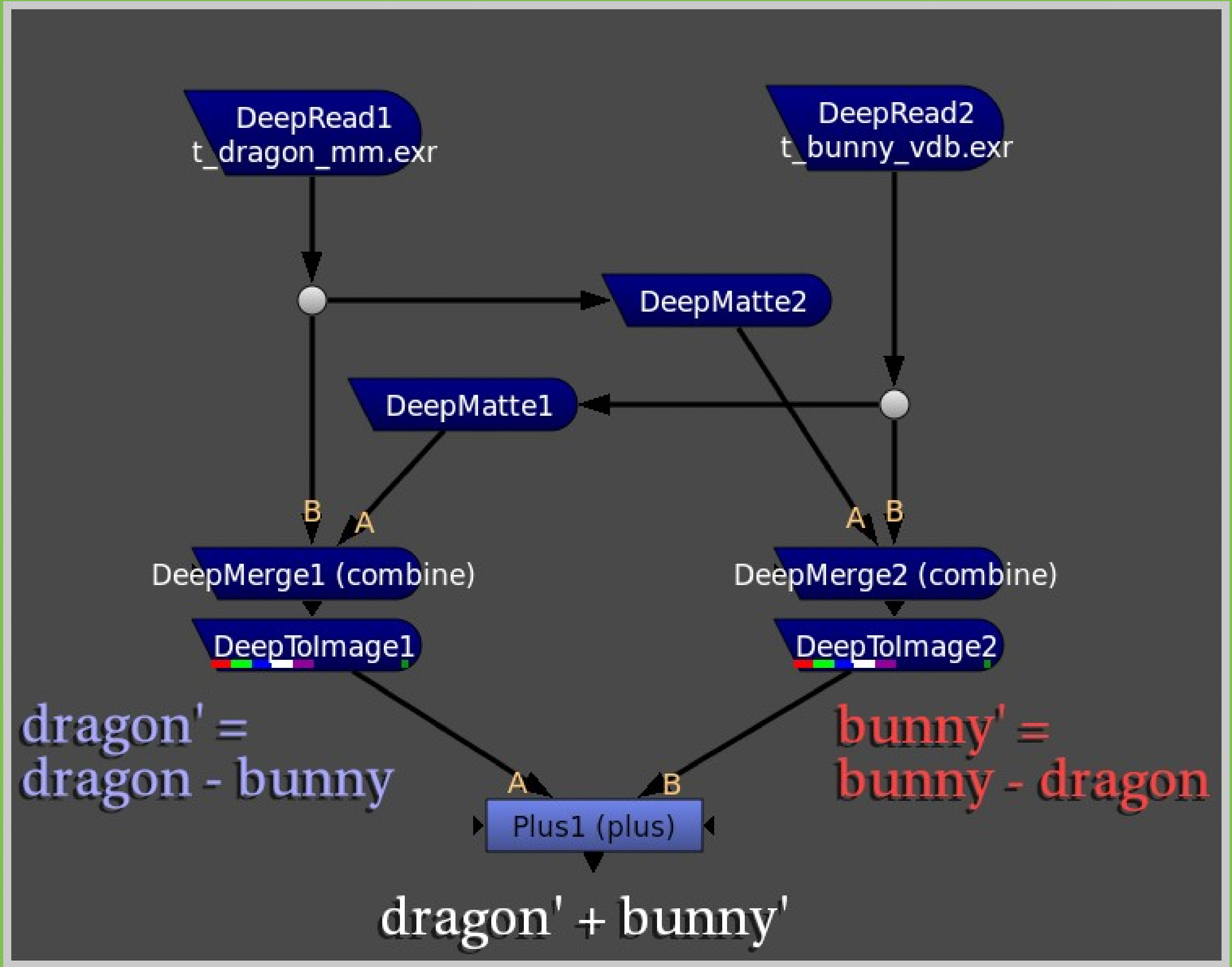
Current deep holdout workflow is destructive and can lead to loss of samples

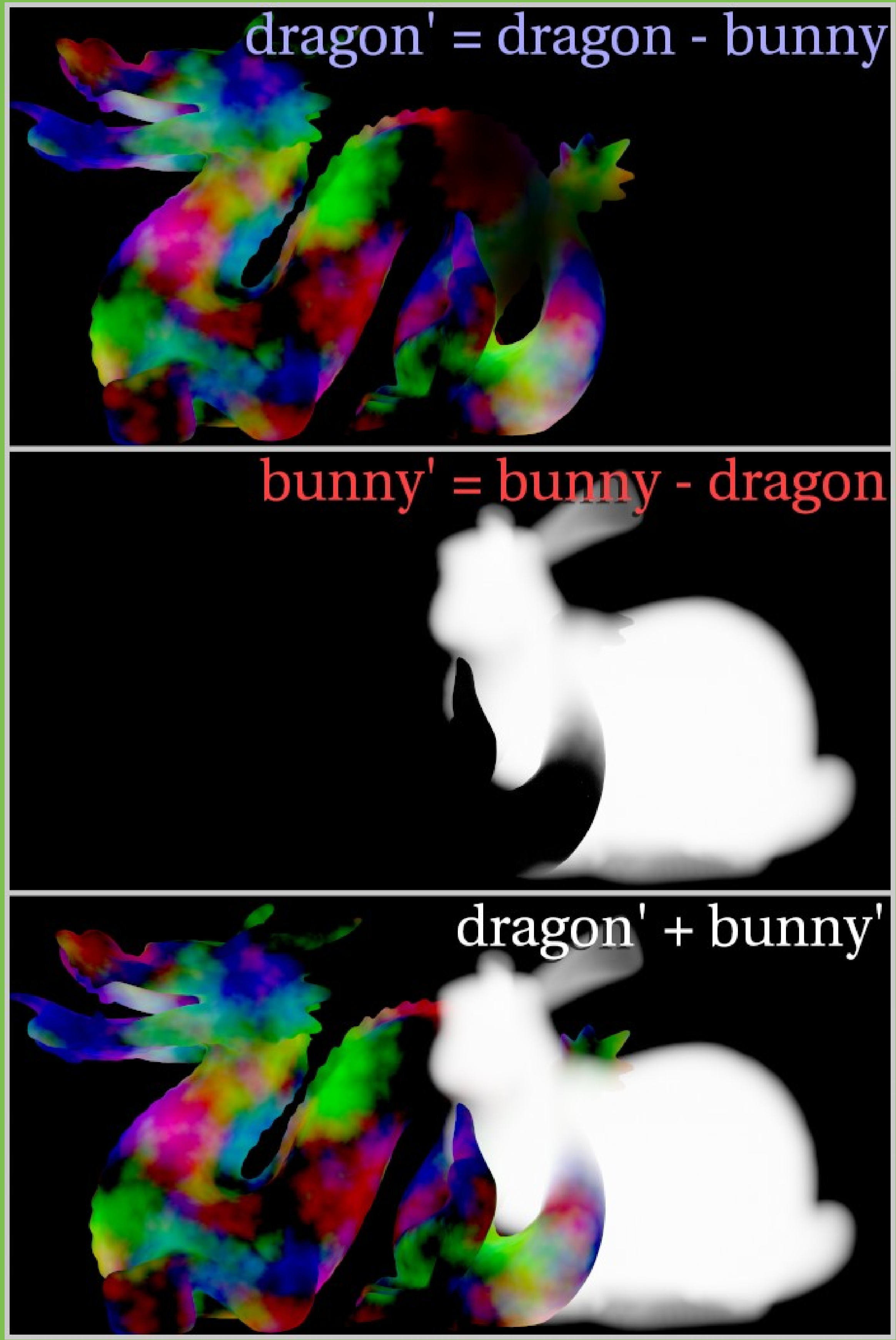
Non-destructive marking of samples allows holdout operation to be deferred

Matte Flag Holdout Example

Nuke script producing mutual cutouts between two renders

Order-independent





OpenEXR 2.0 Deep File I/O

OpenEXR only supports 16 or 32 bit data channels

8x8 bitmask is 64 bits so must be split across two 32-bit **float** channels

32-bit float data channels avoid integer/float conversions destroying bitmask pattern

Flag data stored in 16-bit half-float channel as integer-float values will survive conversions

Nuke Deep System Changes

We are working with the Foundry to get these modifications and new nodes released in a future version of Nuke

DeepToImage – modified flattening operator

DeepSurfaceType – modifies surface flag

DeepMatte – modifies matte flag

DeepPremult – premults/unpremults by pixel-coverage

DeepCamDefocus – camera defocus (bokeh) blur

exrWriterDeep – outputs subpixel mask channels as 32-bit float

Future Work

Extend OpenEXR 2.0 & Nuke to support per-sample metadata

Encourage support from renderer providers

Store more surface information to better define hard-surface intersections (e.g. surface normal)

DreamWorks

the dream is everything

DigiPro2015

Digital Production Symposium