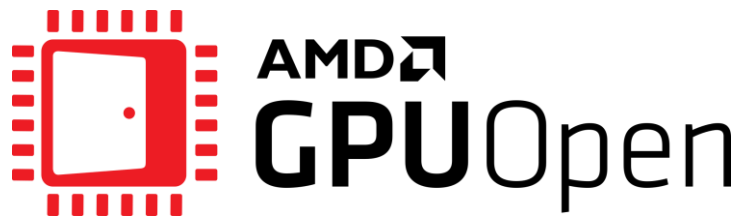


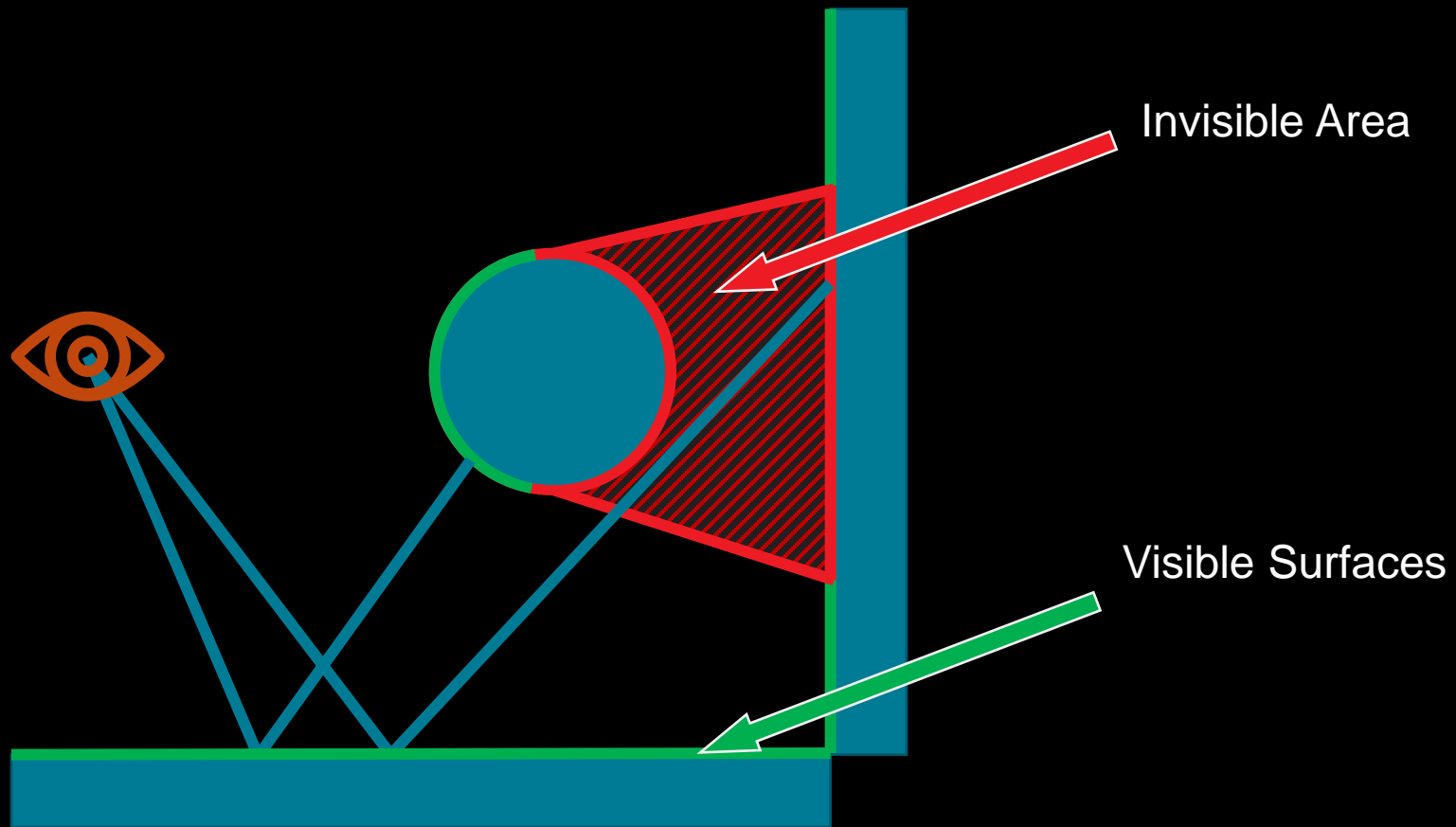
# HYBRID STOCHASTIC REFLECTIONS

ANTON SCHREINER

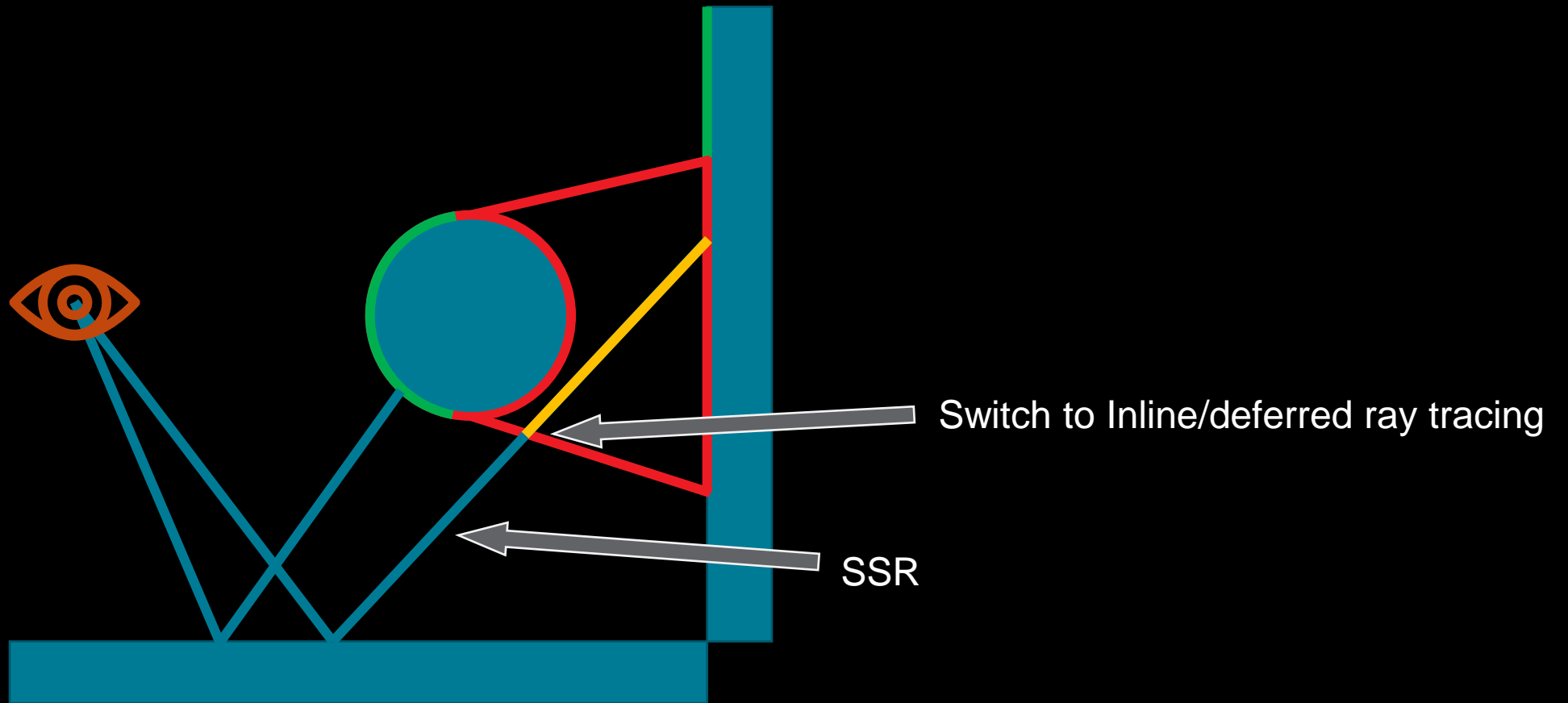
DAVID ZIMAN



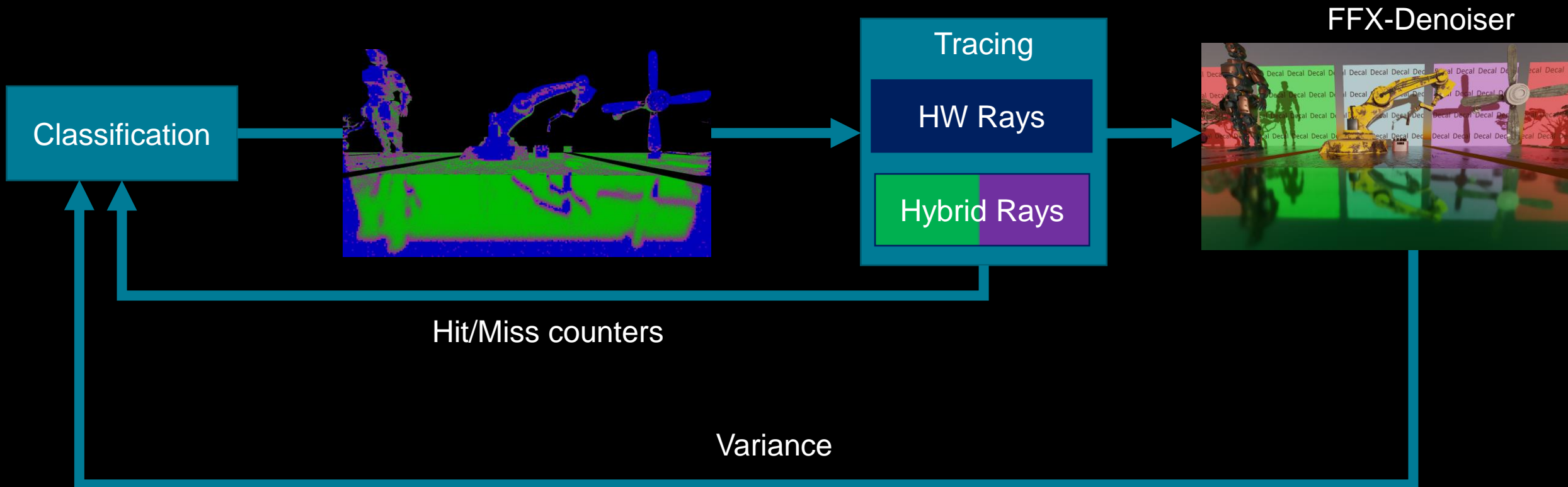
# SCREEN SPACE HYBRIDIZATION



# SCREEN SPACE HYBRIDIZATION

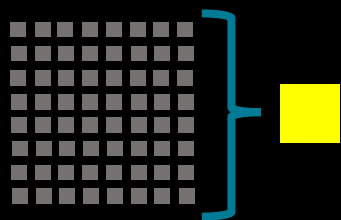


# Feedback based Classification

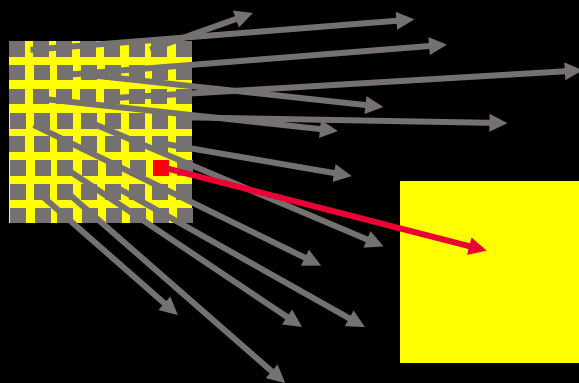


# FEEDBACK COUNTERS

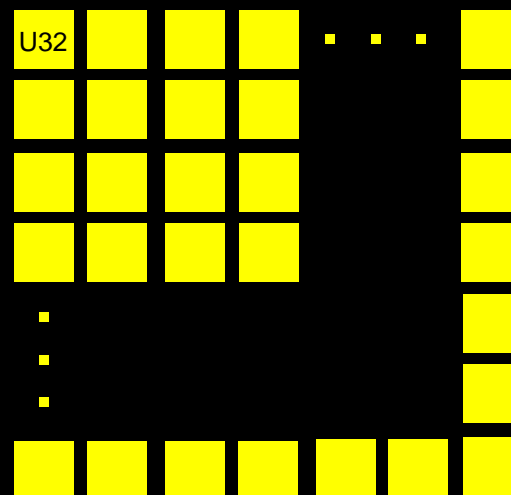
1 U32 per 8x8 tile of pixels



Reprojection:  
Pick a random point in 8x8 region to sample the motion vector



Organized into an image of R32\_UINT elements  
Fits in 4 8-bit counters that contain 2 frame statistics



# FEEDBACK COUNTERS

```
sw_hitcount_old      = sw_hitcount_new;
sw_misscount_old    = sw_misscount_new;
FFX_DNSR_Reflections_StoreHitCounter(dispatch_thread_id.xy / 8,
    [(uint(clamp(sw_hitcount_old, 0.0f, 255.0f)) << 8)
    | (uint(clamp(sw_misscount_old, 0.0f, 255.0f)) << 24)
]);
```

<- Classifier

```
if (!do_hw)
    hit_counter = FFX_HITCOUNTER_SW_HIT_FLAG;
else
    hit_counter = FFX_HITCOUNTER_SW_MISS_FLAG;
```

```
// One atomic increment per wave
uint hit_counter_sum = WaveActiveSum(hit_counter);
if (WaveIsFirstLane()) InterlockedAdd(g_rw_hit_counter[coords / 8], hit_counter_sum);
```

<- SSR

U32

Two frames of statistics




new\_hitcounter:8 | old\_hitcounter:8 | new\_misscounter:8 | old\_misscounter:8

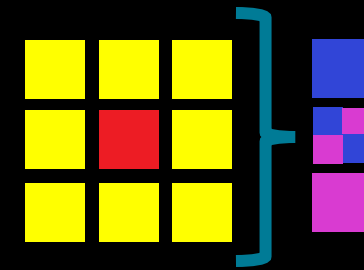
# SAFE BAND

Make the decision based on the neighborhood

- A safe hybrid band is needed to avoid fireflies in rapid movement

There are 3 classes as a result:

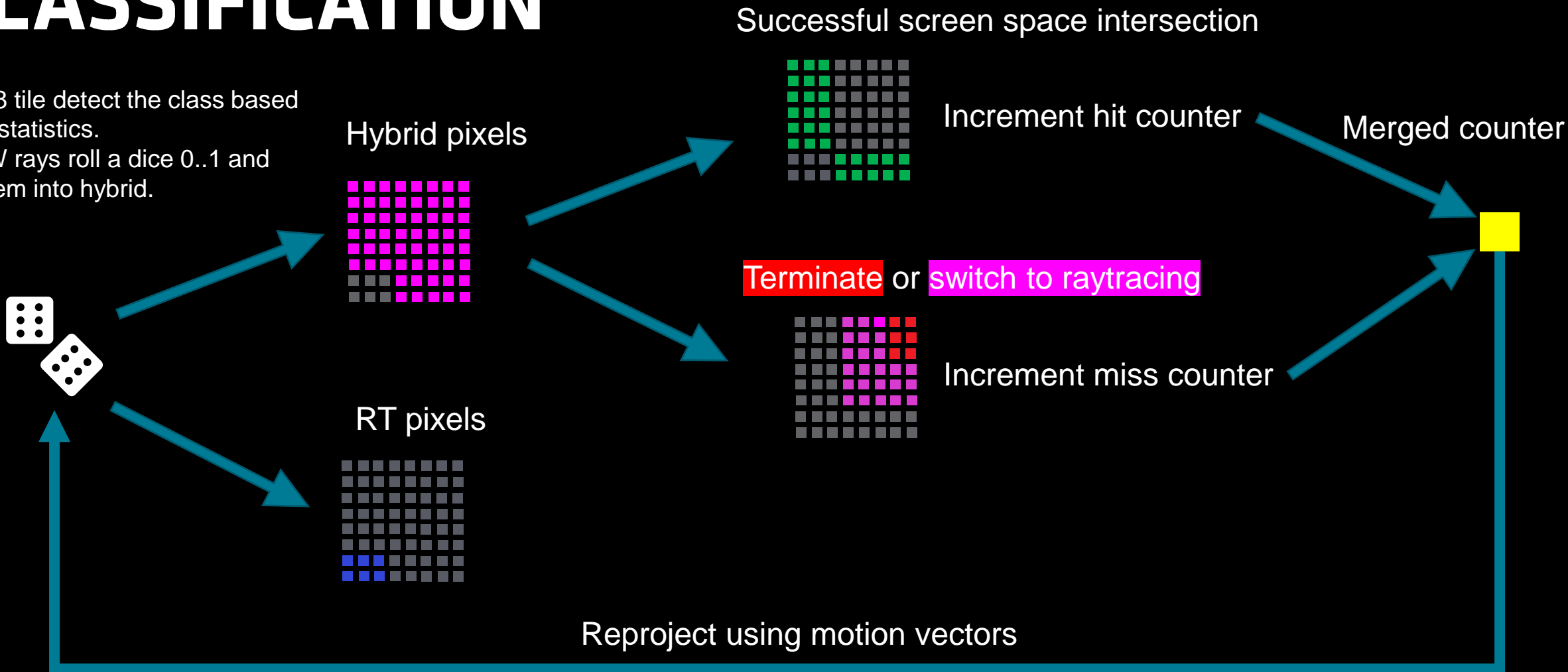
- Full Raytracing 
- Checkerboard hybrid 
- Full Hybrid 



Possible to fix ray type per material and not rely on the classification  
For example, you might want to only use RT for mirrors

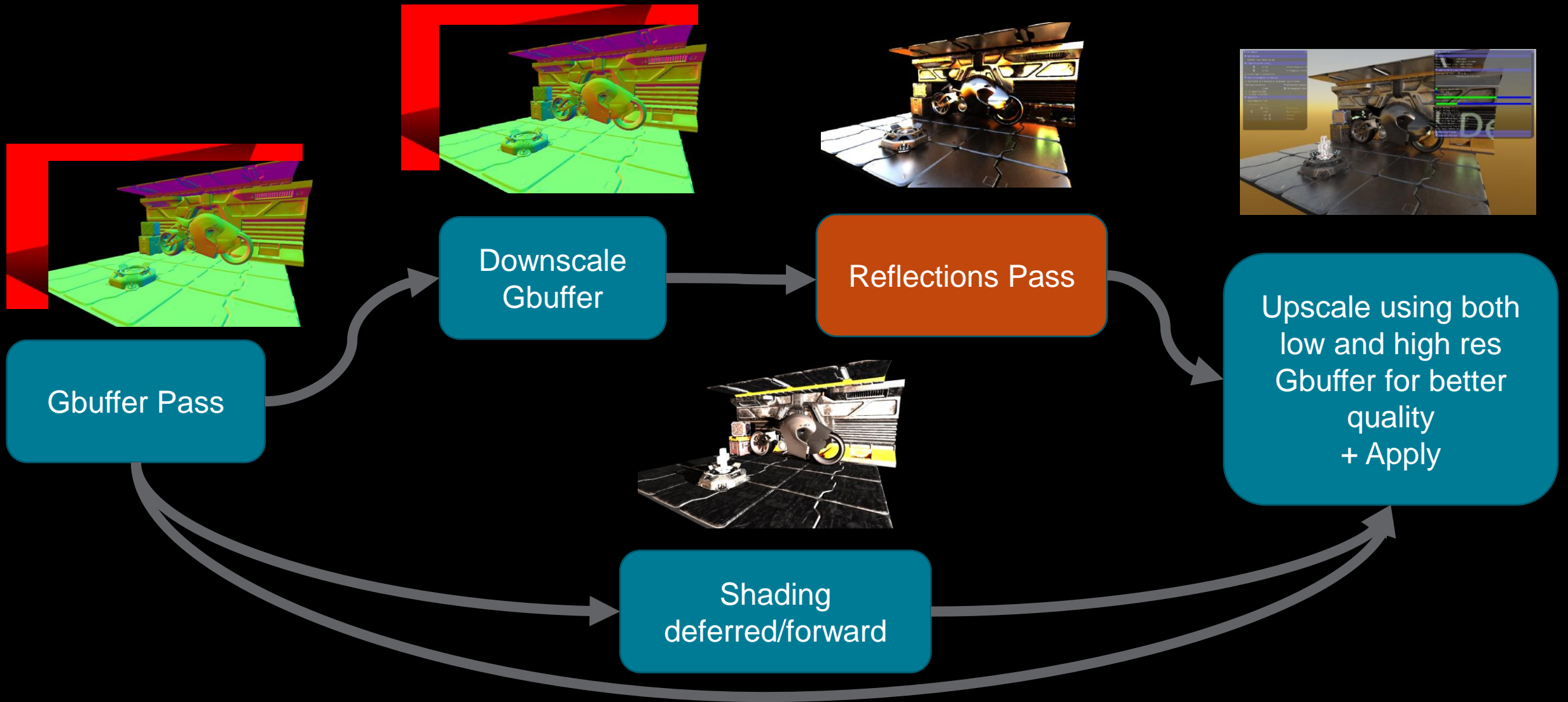
# CLASSIFICATION

Per 8x8 tile detect the class based on the statistics.  
For HW rays roll a dice 0..1 and turn them into hybrid.





# HSR Sample overview



# PASS BREAKDOWN

- Hi-Z – DepthDownsample.hlsl
- Gbuffer Downsampling – HalfResGbuffer.hlsl
- Classify pixels – ClassifyTiles.hlsl
- Ray tracing/FFX-SSSR – Intersect.hlsl
- FFX-Denoiser
  - Reproject.hlsl
  - Prefilter.hlsl
  - TemporalAccumulation.hlsl
- Application/Upscaling – ApplyReflections.hlsl



Derived from FFX-SSSR

# IMPLEMENTATION STEPS

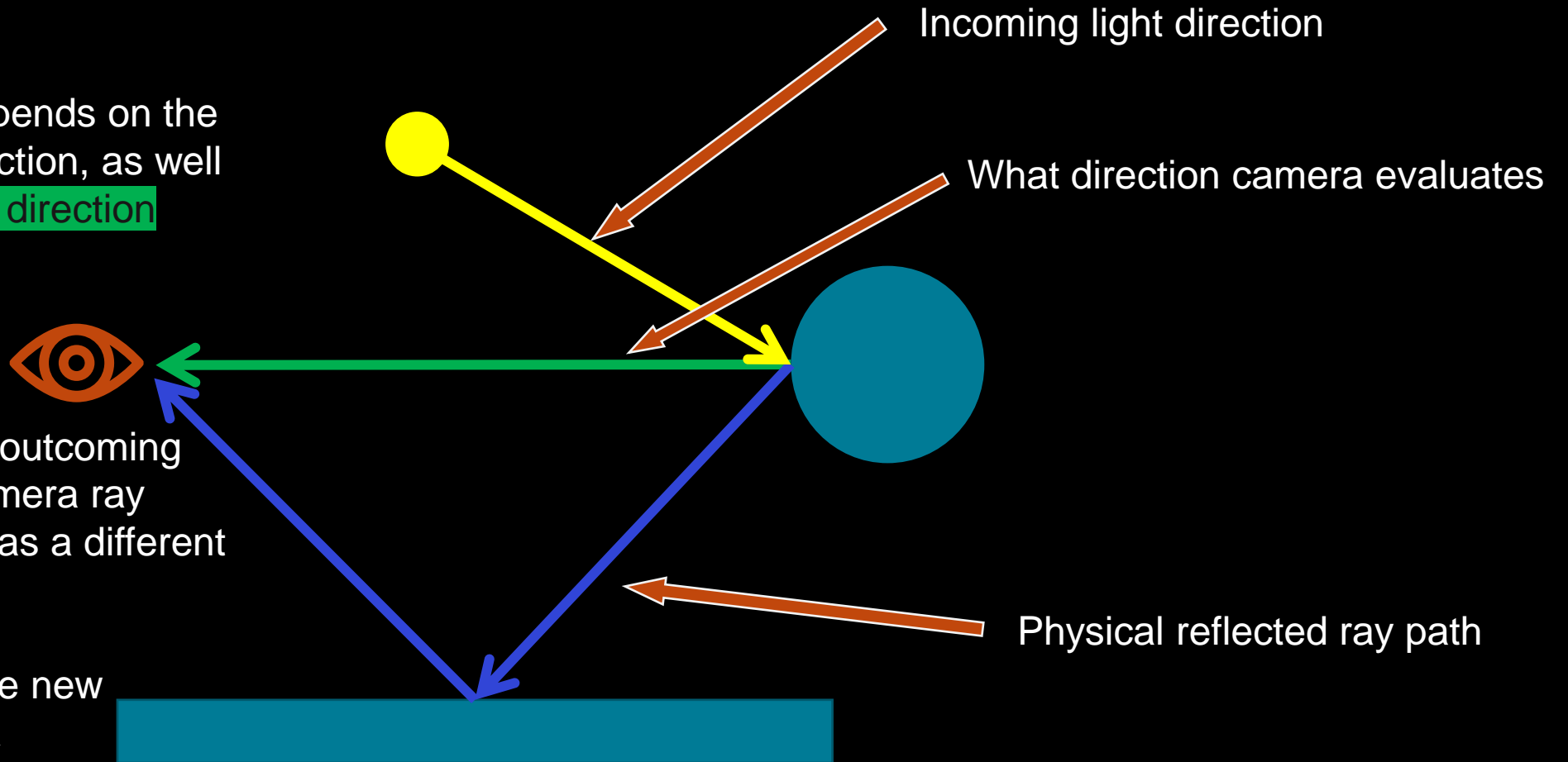
- Implement Ray tracing
  - Material evaluation
  - Geometry streaming/animation
- Integrate FFX-SSSR
  - Hi-Z computation(FFX-SPD)
  - Classification based tracing and denoising
- Implement Hybrid Raytracing
  - Add counters feedback to SSSR
  - Emit two ray lists in classification
- Add upscaling
  - FFX-FSR

# SHADING DISCREPANCIES

- SSR grabs the signal from previous final linear space buffer
- RT shades new fragments

# SHADING DISCREPANCIES

BRDF function depends on the incoming light direction, as well as outgoing light direction  
 $BRDF(D_i, D_o)$

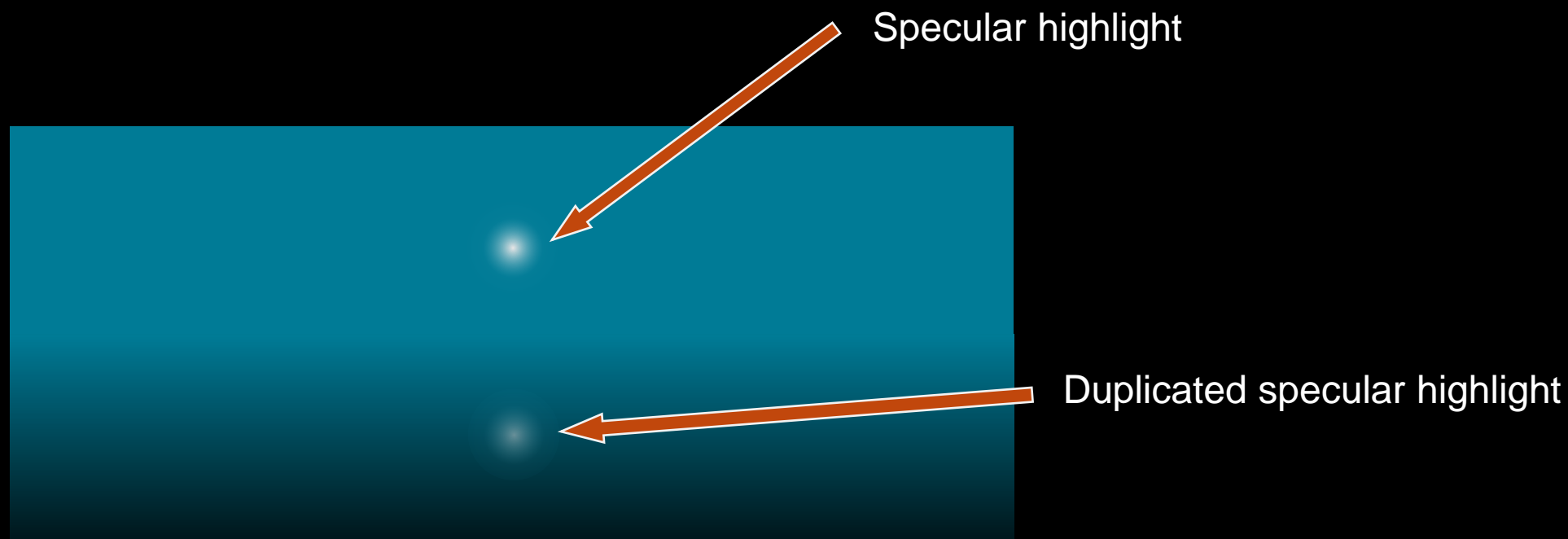


Screen buffer has outgoing direction of the camera ray  
But reflected ray has a different direction

Unless we evaluate new fragments for SSR  
RT is going to have a different value

# SHADING DISCREPANCIES

- SSR is going to duplicate specular highlights that shouldn't have been there



## DISCLAIMER

# DISCLAIMERS

The information contained herein is for informational purposes only and is subject to change without notice. While every precaution has been taken in the preparation of this document, it may contain technical inaccuracies, omissions and typographical errors, and AMD is under no obligation to update or otherwise correct this information. Advanced Micro Devices, Inc. makes no representations or warranties with respect to the accuracy or completeness of the contents of this document, and assumes no liability of any kind, including the implied warranties of noninfringement, merchantability or fitness for particular purposes, with respect to the operation or use of AMD hardware, software or other products described herein. No license, including implied or arising by estoppel, to any intellectual property rights is granted by this document. Terms and limitations applicable to the purchase or use of AMD's products are as set forth in a signed agreement between the parties or in AMD's Standard Terms and Conditions of Sale.

© 2021 Advanced Micro Devices, Inc. All rights reserved. AMD, the AMD Arrow logo, EPYC, Radeon, Ryzen, and combinations thereof are trademarks of Advanced Micro Devices, Inc. in the United States and/or other jurisdictions. Other names are for informational purposes only and may be trademarks of their respective owners.