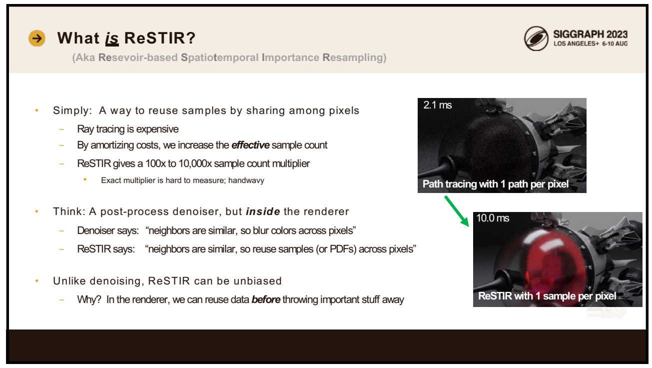
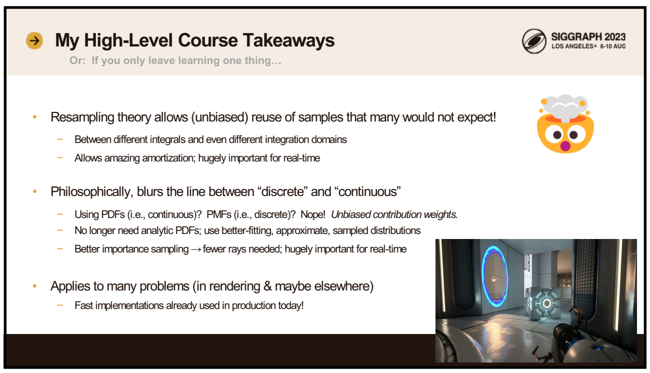

2.4 ms (on a RTX 4090)
7.8 ms
SIGGRAPH 2023
LOS ANGELES+ 6-10 AUG
Path Tracing (1 path per pixel)
ReSTIR PT (1 sample per pixel)
A Gentle Introduction to ReSTIR:
Path Reuse in Real-time
Chris Wyman, Markus Kettunen, Daqi Lin, Benedikt Bitterli,
Cem Yuksel, Wojciech Jarosz, and Pawel Kozlowski
© 2023 SIGGRAPH. ALL RIGHTS RESERVED.

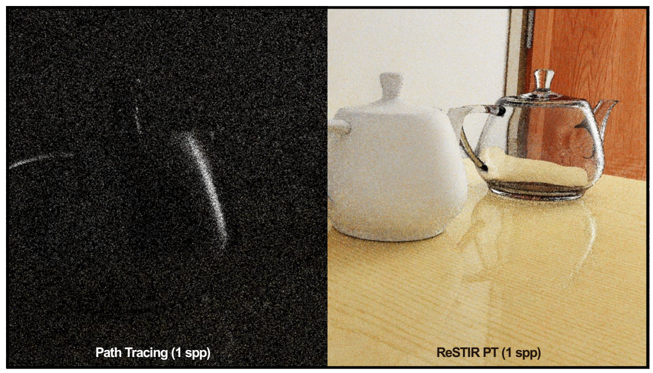
1


What is ReSTIR?
(Aka Resevoir-based Spatiotemporal Importance Resampling)
SIGGRAPH 2023
LOS ANGELES+ 6-10 AUG
Simply: A way to reuse samples by sharing among pixels
Ray tracing is expensive
By amortizing costs, we increase the effective sample count
ReSTIR gives a 100x to 10,000x sample count multiplier
Exact multiplier is hard to measure; handwavy
Think: A post-process denoiser, but inside the renderer
Denoiser says: "neighbors are similar, so blur colors across pixels"
ReSTIR says: "neighbors are similar, so reuse samples (or PDFs) across pixels"
Unlike denoising, ReSTIR can be unbiased
Why? In the renderer, we can reuse data before throwing important stuff away
2.1 ms
Path tracing with 1 path per pixel
10.0 ms
ReSTIR with 1 sample per pixel

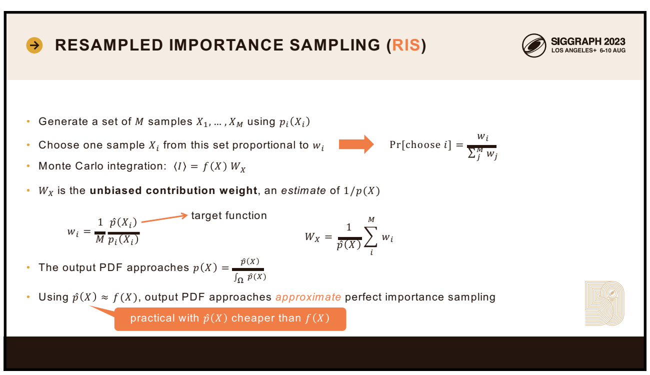
2


My High-Level Course Takeaways
Or: If you only leave learning one thing…
SIGGRAPH 2023
LOS ANGELES+ 6-10 AUG
Resampling theory allows (unbiased) reuse of samples that many would not expect!
Between different integrals and even different integration domains
Allows amazing amortization; hugely important for real-time
Philosophically, blurs the line between "discrete" and "continuous"
Using PDFs (i.e., continuous)? PMFs (i.e., discrete)? Nope! Unbiased contribution weights.
No longer need analytic PDFs; use better-fitting, approximate, sampled distributions
Better importance sampling → fewer rays needed; hugely important for real-time
Applies to many problems (in rendering & maybe elsewhere)
Fast implementations already used in production today!

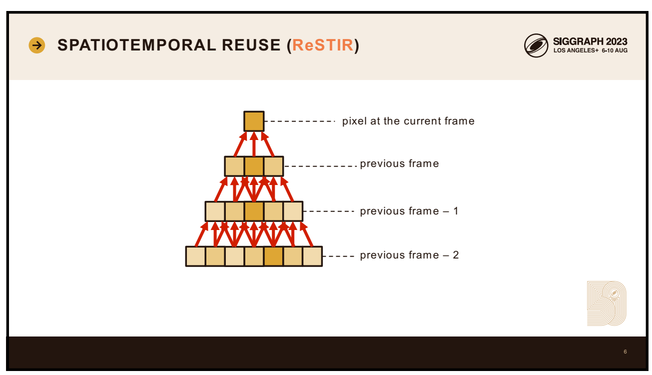
3


Path Tracing (1 spp)
ReSTIR PT (1 spp)

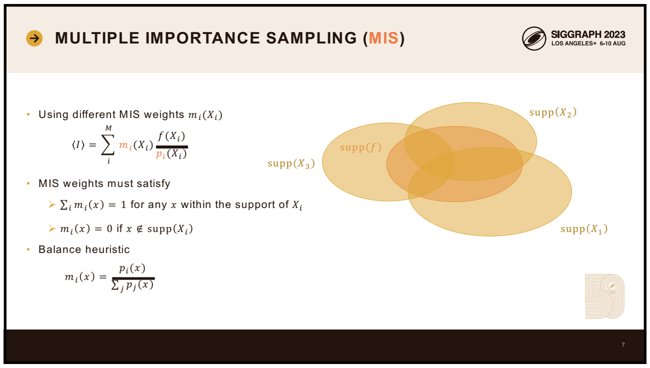
4


RESAMPLED IMPORTANCE SAMPLING (RIS)
SIGGRAPH 2023
LOS ANGELES+ 6-10 AUG
Generate a set of M samples $X_1, \ldots, X_M$ using $p_i(X_i)$
Choose one sample X_i from this set proportional to w_i
$\Pr[\text{choose } i] = \frac{w_i}{\sum_j^M w_j}$
Monte Carlo integration: $\langle I \rangle = f(X)\, W_X$
W_X is the unbiased contribution weight, an estimate of $1/p(X)$
$w_i = \frac{1}{M} \frac{\hat{p}(X_i)}{p_i(X_i)}$
target function
$W_X = \frac{1}{\hat{p}(X)} \sum_i^M w_i$
The output PDF approaches $p(X) = \frac{\hat{p}(X)}{\int_\Omega \hat{p}(X)}$
Using $\hat{p}(X) \approx f(X)$, output PDF approaches approximate perfect importance sampling
practical with $\hat{p}(X)$ cheaper than $f(X)$

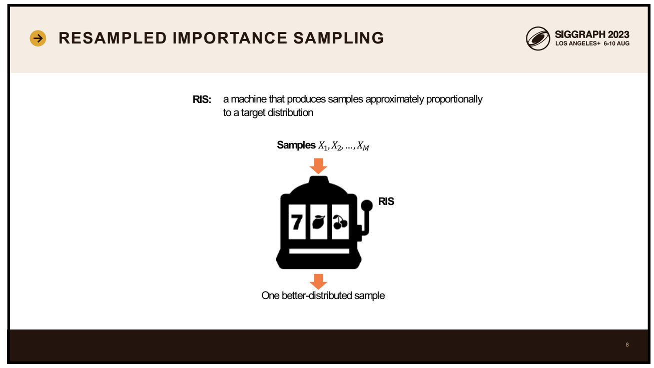
5


SPATIOTEMPORAL REUSE (ReSTIR)
SIGGRAPH 2023
LOS ANGELES+ 6-10 AUG
pixel at the current frame
previous frame
previous frame – 1
previous frame – 2

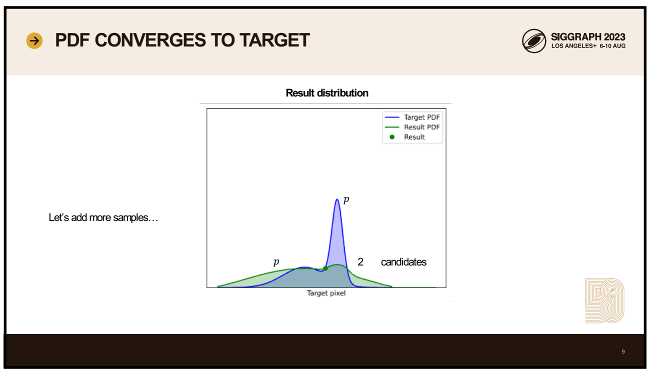
6


MULTIPLE IMPORTANCE SAMPLING (MIS)
SIGGRAPH 2023
LOS ANGELES+ 6-10 AUG
Using different MIS weights $m_i(X_i)$
$\langle I \rangle = \sum_i^M m_i(X_i) \frac{f(X_i)}{p_i(X_i)}$
MIS weights must satisfy
$\sum_i m_i(x) = 1$ for any x within the support of X_i
$m_i(x) = 0$ if $x \notin \mathrm{supp}(X_i)$
Balance heuristic
$m_i(x) = \frac{p_i(x)}{\sum_j p_j(x)}$
$\mathrm{supp}(X_2)$
$\mathrm{supp}(f)$
$\mathrm{supp}(X_3)$
$\mathrm{supp}(X_1)$
7

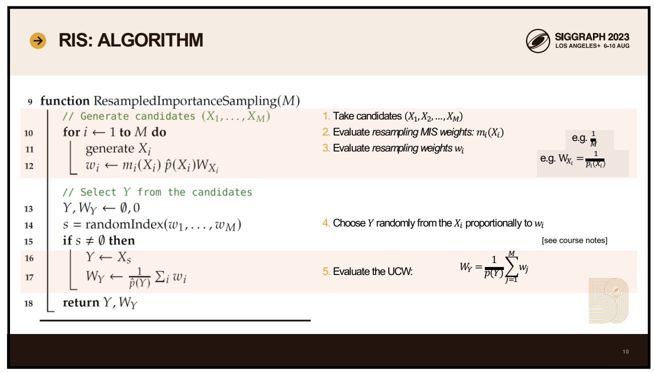
7


RESAMPLED IMPORTANCE SAMPLING
SIGGRAPH 2023
LOS ANGELES+ 6-10 AUG
RIS: a machine that produces samples approximately proportionally to a target distribution
Samples $X_1, X_2, \ldots, X_M$
RIS
One better-distributed sample

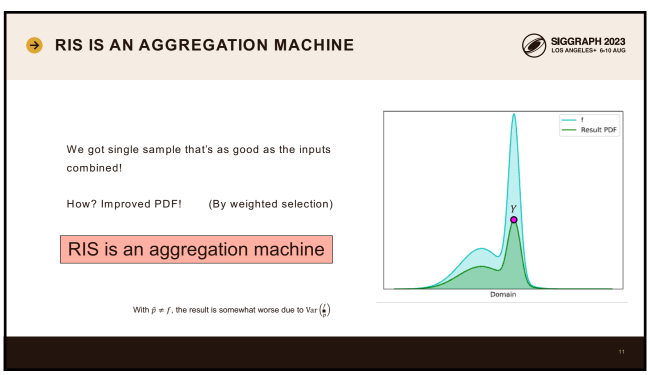
8


PDF CONVERGES TO TARGET
SIGGRAPH 2023
LOS ANGELES+ 6-10 AUG
Let's add more samples…
Result distribution
Target PDF
Result PDF
Result
p
p
2 candidates
Target pixel

9


RIS: ALGORITHM
SIGGRAPH 2023
LOS ANGELES+ 6-10 AUG
9 function ResampledImportanceSampling(M)
// Generate candidates $(X_1, \ldots, X_M)$
10 for $i \leftarrow 1$ to M do
11 generate X_i
12 $w_i \leftarrow m_i(X_i)\, \hat{p}(X_i) W_{X_i}$
// Select Y from the candidates
13 $Y, W_Y \leftarrow \emptyset, 0$
14 s = randomIndex$(w_1, \ldots, w_M)$
15 if $s \neq \emptyset$ then
16 $Y \leftarrow X_s$
17 $W_Y \leftarrow \frac{1}{\hat{p}(Y)} \sum_i w_i$
18 return Y, W_Y
1. Take candidates $(X_1, X_2, \ldots, X_M)$
2. Evaluate resampling MIS weights: $m_i(X_i)$
3. Evaluate resampling weights w_i
e.g. $\frac{1}{M}$
e.g. $W_{X_i} = \frac{1}{p_i(X_i)}$
4. Choose Y randomly from the X_i proportionally to w_i
[see course notes]
5. Evaluate the UCW:
$W_Y = \frac{1}{\hat{p}(Y)} \sum_{j=1}^{M} w_j$
10

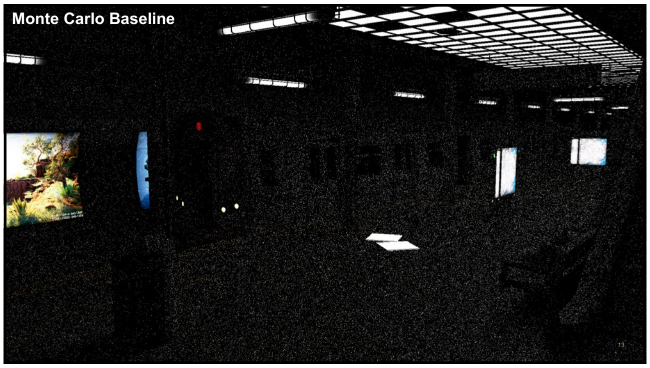
10


RIS IS AN AGGREGATION MACHINE
SIGGRAPH 2023
LOS ANGELES+ 6-10 AUG
We got single sample that's as good as the inputs combined!
How? Improved PDF! (By weighted selection)
RIS is an aggregation machine
With $\hat{p} \neq f$, the result is somewhat worse due to $\mathrm{Var}\left(\frac{f}{\hat{p}}\right)$
f
Result PDF
Y
Domain
11

11


THE PREMIER CONFERENCE & EXHIBITION ON COMPUTER GRAPHICS & INTERACTIVE TECHNIQUES
SIGGRAPH 2023
LOS ANGELES+ 6-10 AUG
RESTIR COURSE
RIS & DIRECT LIGHTING
© 2023 SIGGRAPH. ALL RIGHTS RESERVED.

12



13

14

RESERVOIR SAMPLING

SIGGRAPH 2023 LOS ANGELES+ 6-10 AUG

```
struct Reservoir
    Path sampleOut = ∅      // Selected sample
    float w_sum = 0         // Sum of weights
    void addSample(Path x, float w)
        w_sum = w_sum + w
        if rand() < w/w_sum then
            sampleOut = x
```

Render passes:
generateSamples()
shadePixel()

15

SPATIAL REUSE

SIGGRAPH 2023 LOS ANGELES+ 6-10 AUG

```
void reuseSpatially()
    Reservoir r
    for i = 1 to k do
        p = q + sampleRandomDisk()
        Sample s = pixelSample[p]
        w = m_space(s.x) · p̂(s.x) · s.W
        r.addSample(s, w)
    y = r.sampleOut
    W = 1/p̂(y) · r.w_sum
    pixelSample[q] = Sample{y, W}
```

Render passes:
generateSamples()
reuseSpatially()
reuseSpatially()
…
shadePixel()

16

SPATIAL REUSE

SIGGRAPH 2023 LOS ANGELES+ 6-10 AUG

```
void reuseSpatially()
    Reservoir r
    for i = 1 to k do
        p = q + sampleRandomDisk()
        Sample s = pixelSample[p]
        w = m_space(s.x) · p̂(s.x) · s.W
        r.addSample(s, w)
    y = r.sampleOut
    W = 1/p̂(y) · r.w_sum
    pixelSample[q] = Sample{y, W}
```

Render passes:
generateSamples()
reuseSpatially()
reuseSpatially()
…
shadePixel()

17

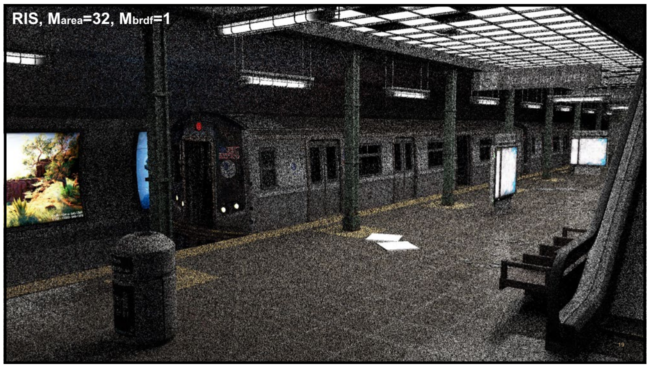
SPATIAL REUSE

SIGGRAPH 2023 LOS ANGELES+ 6-10 AUG

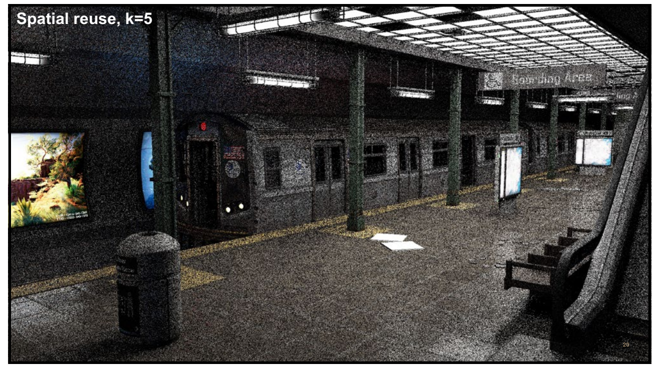
```
void reuseSpatially()
    Reservoir r
    for i = 1 to k do
        p = q + sampleRandomDisk()
        Sample s = pixelSample[p]
        w = m_space(s.x) · p̂(s.x) · s.W
        r.addSample(s, w)
    y = r.sampleOut
    W = 1/p̂(y) · r.w_sum
    pixelSample[q] = Sample{y, W}
```

Render passes:
generateSamples()
reuseSpatially()
reuseSpatially()
…
shadePixel()

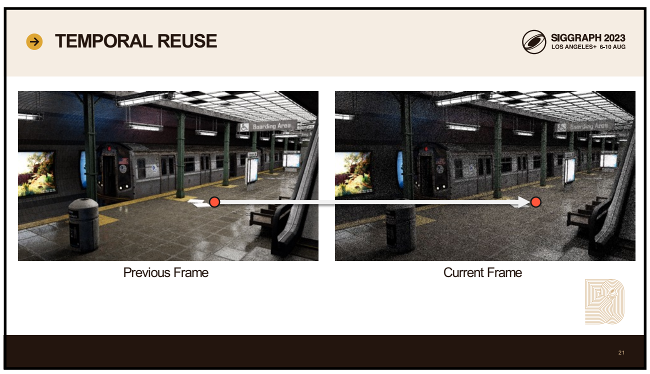
18



19

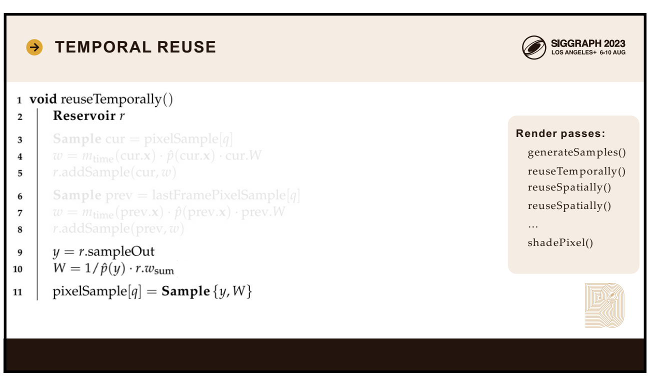


20

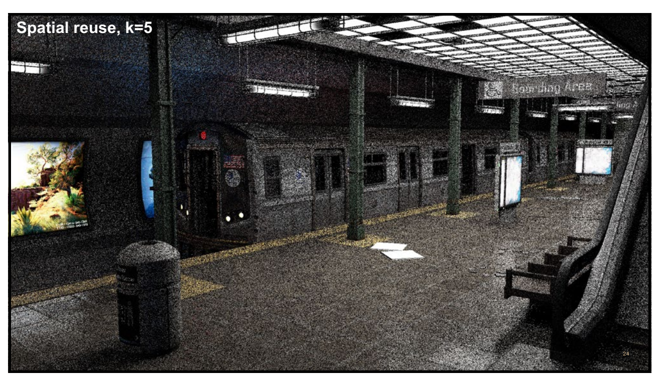


21

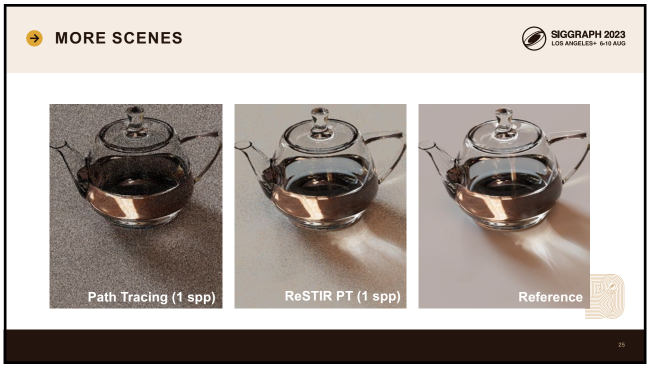
22



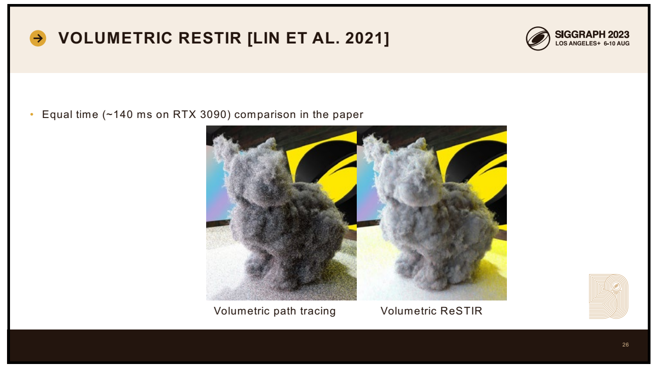
23



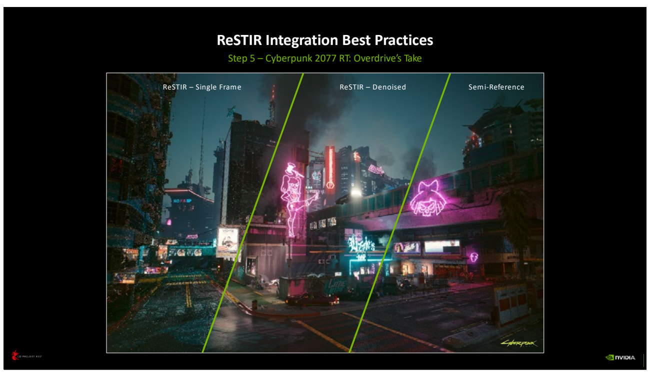
24


MORE SCENES
SIGGRAPH 2023
LOS ANGELES+ 6-10 AUG
Path Tracing (1 spp)
ReSTIR PT (1 spp)
Reference

25


VOLUMETRIC RESTIR [LIN ET AL. 2021]
SIGGRAPH 2023
LOS ANGELES+ 6-10 AUG
Equal time (~140 ms on RTX 3090) comparison in the paper
Volumetric path tracing
Volumetric ReSTIR

26


ReSTIR Integration Best Practices
Step 5 – Cyberpunk 2077 RT: Overdrive's Take
ReSTIR – Single Frame
ReSTIR – Denoised
Semi-Reference

27


NVIDIA ReSTIR RTX DI
Bitterli et al. 20 Spatiotemporal reservoir resampling
Real-Time Direct Lighting from millions of lights
BRING IT!
METROPOLITAN

28