

Specialieji efektai 2

T120B167 Žaidimų grafinių specialiųjų efektų kūrimas ir programavimas



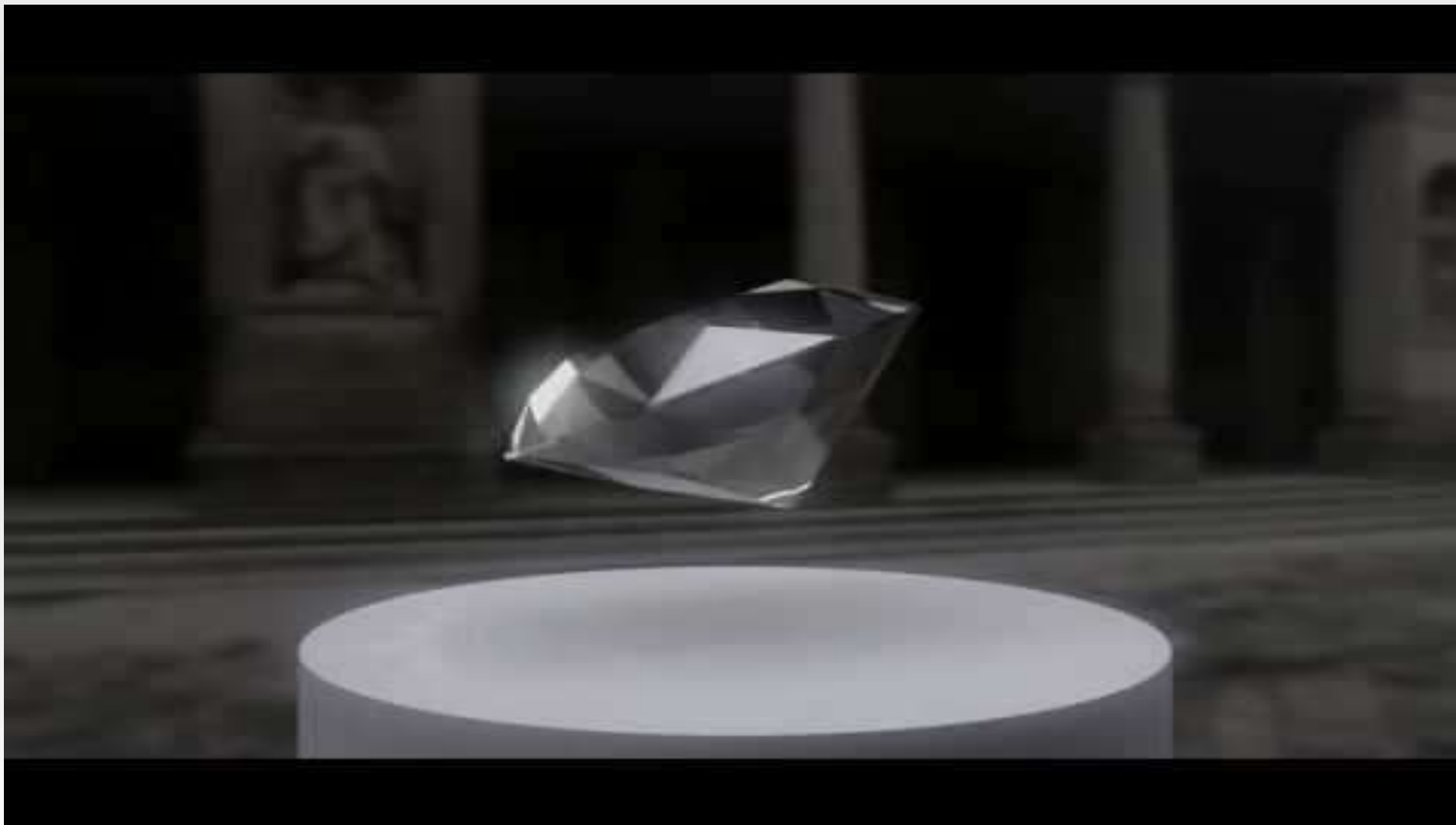
Rytis Maskeliūnas
Skype: rytmask
Rytis.maskeliunas@ktu.lt

© R. Maskeliūnas >2013
© A. Noreika <2013

Paskaitos tema

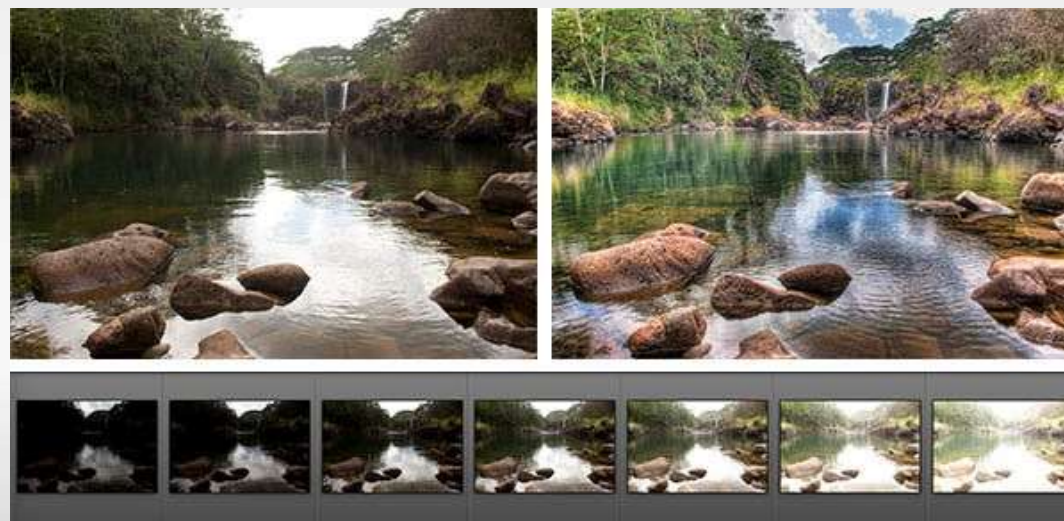


MULTIMEDIJOS
INŽINERIJOS
KATEDRA



http://www.youtube.com/watch?v=9779RtEN_uA

- Dinaminis diapazonas (angl. dynamic range) - tai skirtumas tarp šviesiausių ir tamsiausių vaizdo zonų.
- Žmogaus akis prisitaiko automatiškai
- Vaizdo apdorojimo sistemoje ar 3D grafikoje reikalingas specialus algoritmas
- HDR algoritmas - tai manipuliacijų metodai didelio dinaminio diapazono vaizdą transformuojantys į mažesnio diapazono renderiuojamą vaizdą



Ką galima padaryti su HDR?



MULTIMEDIJOS
INŽINERIJOS
KATEDRA

- Akinanti šviesa (Dazzling light)
- Akinantis atspindys Dazzling reflection
- Fresnel atspindys (Fresnel reflection)
 - Ryškus atspindys normalės kryptimi
- Ekspozicijos valdymo efektai (Exposure control effects)
- Realistikas DOF (Realistic depth-of-field effects)
- Realistiškas judesio blur'as (Realistic motion blur)

Dazzling Light



MULTIMEDIJOS
INŽINERIJOS
KATEDRA



Dazzling Reflection



MULTIMEDIJOS
INŽINERIJOS
KATEDRA



HDR Fresnel



MULTIMEDIJOS
INŽINERIJOS
KATEDRA

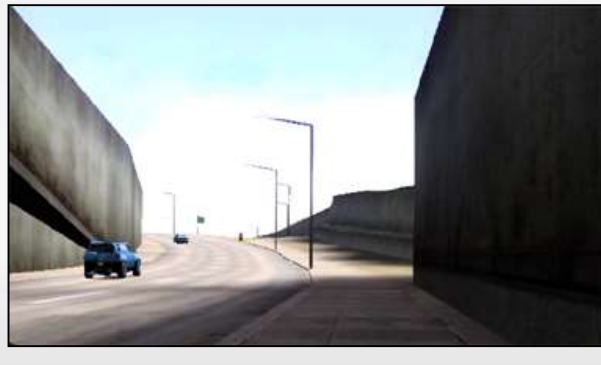


Ryškūs atspindžiai nuo
mažo atspindamumo
paviršių

Exposure Control



MULTIMEDIJOS
INŽINERIJOS
KATEDRA



HDR Depth-of-Field



MULTIMEDIJOS
INŽINERIJOS
KATEDRA



Presented by Masaki Kawase.
<http://www.daionet.gr.jp/~masa>



Glare Type (g/G) : 1 / 12 Camera
Exposure : Auto 0.910
Rotate Model (Ctrl+o)
Depth of Field (d/D) : 4 / 4 6-blade Iris (141 taps)
Focal Distance (PgUp/PgDown)

Presented by Masaki Kawase.
<http://www.daionet.gr.jp/~masa>

HDR Motion Blur



MULTIMEDIJOS
INŽINERIJOS
KATEDRA

29.40 fps (640x480). A16B16G16R16F (D16) (8x Multisample)
HAL (pure hw vp): RADEON 9700/9500 SERIES

Mouse Controls:

L : Rotate View
M | Shift+L : Translate
R | Ctrl+L : Scale/FOV
L+R | Alt+L : Rotate Model

Glare Type (g/G) : 5 / 12 Cross Screen Filter
Exposure : Auto 0.642
Open Model (Ctrl+o)
Depth of Field (d/D) : 3 / 3 6-blade Iris
Focal Distance (PgUp/PgDown)

Presented by Masaki Kawase.
<http://www.daionet.gr.jp/~masa>

17.66 fps (640x480). A16B16G16R16F (D16) (16x Multisample)
HAL (pure hw vp): RADEON 9700/9500 SERIES

Mouse Controls:

L : Rotate View
M | Shift+L : Translate
R | Ctrl+L : Scale/FOV
L+R | Alt+L : Rotate Model

Glare Type (g/G) : 1 / 12 Camera
Exposure : Auto 1.691
Open Model (Ctrl+o)
Depth of Field (d/D) : 3 / 3 6-blade Iris (37 taps)
Focal Distance (PgUp/PgDown)

Presented by Masaki Kawase.
<http://www.daionet.gr.jp/~masa>

Dinaminis diapazonas



MULTIMEDIJOS
INŽINERIJOS
KATEDRA



Dinaminis diapazonas (Dynamic Range)



MULTIMEDIJOS
INŽINERIJOS
KATEDRA

- Santykis nuo didžiausios iki mažiausios vertės kurs gali būti atvaizduojamas
- Rodomas vaizdas (Displayable image)
 - 2^8 Low dynamic range (LDR)
- Absoliutaus šviesio (absolute luminance) kadro buferis (Frame buffer)
 - Renderuoti sceną absoliučiai šviesioje erdvėje
 - $>2^{32}$ atvaizduoja visus šviesius
- Reliatyvaus (relative) šviesio kadro buferis
 - Panaudoti kintamus ekspozicijos renderiavimo metu
 - $>2^{15\sim 16}$ tamsūs regionai nesvarbu

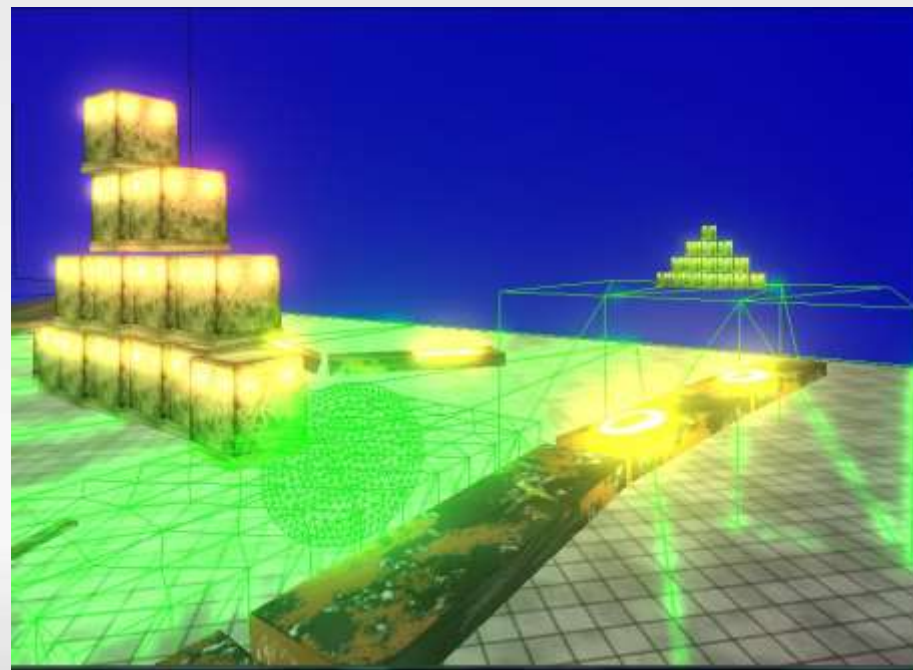
- Kadro buferiai (Frame buffers)
 - Spindesio (glare) generavimui
- Aplinkos žemėlapiai (Environment maps)
 - Siekiant apsaugoti šviesą nuo nematomumo (užslopinimo) kai paviršius yra labai mažai atspindintis
 - Siekiant gauti akinimą atspindį
- „Šviesą“ skleidžiančios tekstūros (Self-emission textures)
 - post-rendering glare generation
- Statinėms (Decal) tekstūroms nereikia HDR

HDR kadry buferiai



MULTIMEDIJOS
INŽINERIJOS
KATEDRA

- Spindesio generavimui
- Kai renderiuojama su reliatyviais šviesiais:
 - Idealiu atveju daugiau nei $2^{15\sim16}$
 - Žaidimuose
 - pakanka $2^{12\sim13}$ (4,000~10,000)



- Labai svarbu siekiant pavaizduoti:
 - Realistišką specular atspindį
 - Akinantį specular atspindį
- Specular atspindžio modeliavimui kai medžiaga ne metalas
 - Atspindys normalės kryptimi paprastai yra mažiau nei 4%
 - Ryški šviesa lieka ryški po tokio mažo atspindžio
- Norint išlaikyti akinimus (dazzles) po atspindžio nuo ~1-4%
 - Reikia naudoti platų dinaminį diapazoną



- Yra problemų su našumu.
- Pixel Shader
 - Pixel Shader 2.0 ar naujesnis
- HDR buferių formatai
 - Aukšto tikslumo integer/float buferiai
 - Ar žemo tikslumo integer buferiai

Problemos su aukšto tikslumo buferiais



MULTIMEDIJOS
INŽINERIJOS
KATEDRA

- Atminties naudojimas
 - A16B16G16R16 / A16B16G16R16F
 - 64bpp (bits per pixel)
 - Dukart daugiau nei A8R8G8B8
 - A32B32G32R32 / A32B32G32R32F
 - 128bpp
 - 4x A8R8G8B8
- Reikai ne mažiau nei du kart GPU atminties nei įprastinio pilnos spalvos buferio (full-color buffer) atveju

Problemos su aukšto tikslumo buferiais

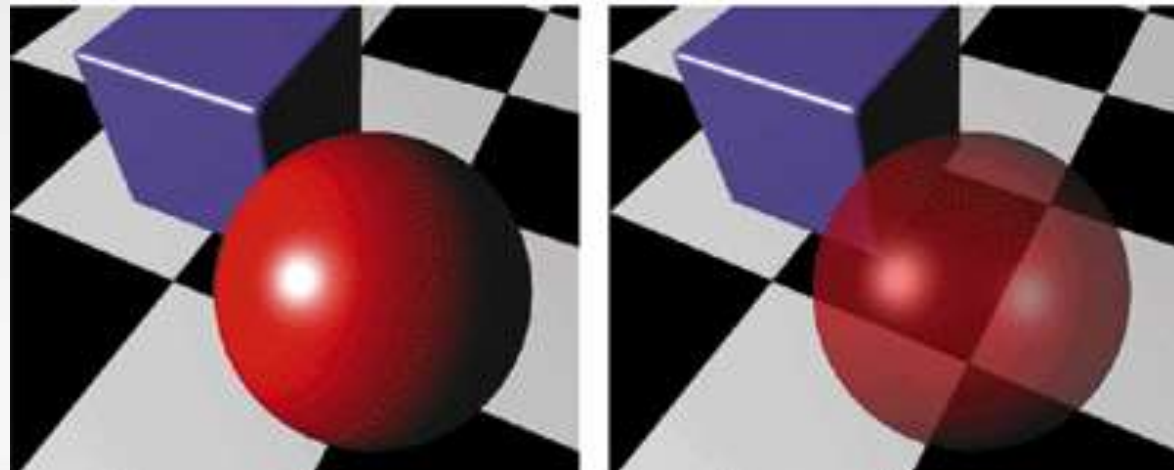


MULTIMEDIJOS
INŽINERIJOS
KATEDRA

- Apribojimai
 - Bėdos su Alpha blending panaudojimu (nepanaudojama)
 - Neveikia tekstūrų filtravimas su floating-point formatais
 - Tai įtakoja aplinkos žemėlapių ir spindinčių tekstūrų (self-emission textures) kokybę



- A8R8G8B8 / A2R10G10B10 etc.
- Valgo mažai atminties
- Veikia Alpha blending
 - Gali atsirasti nedidelių, sunkiai pastebimų klaidų



Kompresija su Tone Mapping



MULTIMEDIJOS
INŽINERIJOS
KATEDRA

- Renderiuojam tiesiai į rodomą formatą
- Nonlinear spalvos kompresija

$$Buffer(x, y) = \frac{Lum_{exposed}(x, y)}{Lum_{exposed}(x, y) + 1}$$

- Efektyviai platus dinaminis diapazonas
- Principas:
 - Reinhard, Erik, Mike Stark, Peter Shirley, and Jim Ferwerda, [*“Photographic Tone Reproduction for Digital Images”*](#)
- Nenaudojamas alpha kanalas
 - Todėl netinka kitam tikslui





- A8R8G8B8 (8 bits per color channel)
 - Alpha channel galima panaudoti kitam tikslui
 - Veikia bet kokioms sistemoms
 - Gali pritrūkti RGB tikslumo
- A2B10G10R10
A2R10G10B10 (10 bits per color channel)
 - Spalvų kanalų tikslumas pilnai pakankamas
 - Keli alpha bitai
 - Netinka kitai informacijai saugoti
 - Nevisos sistemos palaiko



- Idealiu atveju:
 - Dinaminis diapazonas daugiau nei 10,000 ar 20,000
 - Panaudotas tekstūrų filtravimas





- Santykinai maža rezoliucija
- Alpha channel/blending nelabai svarbu
- Naudojamas 16-bit integer formatas jei yra pakankamai atminties
 - Galima imti intervalu $[0, 256]$ ar $[0, 512]$
 - Greitas užkodavimas/dekodavimas
 - Galima naudoti tekstūrų filtravimą

- Panašu į tone mapping
- Užkoduojama kai renderiuojamas environment map'as

$$Buffer(x, y) = \frac{Lum_{exposed}(x, y)}{Lum_{exposed}(x, y) + Offset}$$

Offset : šviesio (luminance) kreivės faktorius (~2-4)

- Didesnis offset reiškia:
 - Aukšto šviesio (High-luminance) regionai turi didesnę rezoliuciją
 - Žemo šviesio (Low-luminance) - mažesnę

- Dekoduojama renderiuojant į frame buffer

- Iš environment map

$$Lum_{exposed}(x, y) = \frac{Buffer(x, y) \times Offset}{1 + \delta - Buffer(x, y)}$$

δ : maža vertė, kad nebūtų dalyba iš nulio

- Svarbu neperlenkti lazdos
 - Gali atsirasti Mach banding ant atspindžių nuo didelio ploto šviesos šaltinių



- RGB eksponentė saugoma alpha kanale

- Bazė (base) nuo 1.04 iki 1.08

$$Lum_{exposed} = Buffer.rgb \times base^{Buffer.a \times 256 - offset} \quad offset : \sim 64-128$$

- Base=1.04 turi dinaminį diapazoną $\sim 23,000$ (1.04^{256})
- Didesnė base vertė:
 - Didesnis HDR
 - Žemesnė rezoliucija (gali matytis Mach banding)
- Užkoduojama kai renderiuojama į environment map
- Dekoduojama kai renderiuojama į frame buffer

E8R8G8B8 HLSL



MULTIMEDIJOS
INŽINERIJOS
KATEDRA

```
// a^n = b
#define LOG(a, b)    ( log((b)) / log((a)) )

#define EXP_BASE      (1.06)
#define EXP_OFFSET    (128.0)

// Pixel Shader (6 instruction slots)
// rgb already exposure-scaled
float4 EncodeHDR_RGB_RGBE8(in float3 rgb)
{
    // Compute a common exponent
    float fLen = dot(rgb.rgb, 1.0) ;
    float fExp = LOG(EXP_BASE, fLen) ;

    float4 ret ;
    ret.a = (fExp + EXP_OFFSET) / 256 ;
    ret.rgb = rgb / fLen ;

    return ret ;
}
```

```
// More accurate encoding
#define EXP_BASE      (1.04)
#define EXP_OFFSET    (64.0)

// Pixel Shader (13 instruction slots)
float4 EncodeHDR_RGB_RGBE8(in float3 rgb)
{
    float4 ret ;

    // Compute a common exponent
    // based on the brightest color channel
    float fLen = max(rgb.r, rgb.g) ;
    fLen = max(fLen, rgb.b) ;
    float fExp = floor( LOG(EXP_BASE, fLen) ) ;

    float4 ret ;
    ret.a = clamp( (fExp + EXP_OFFSET) / 256, 0.0, 1.0 ) ;
    ret.rgb = rgb / pow(EXP_BASE, ret.a * 256 - EXP_OFFSET) ;

    return ret ;
}
```

E8R8G8B8 Dekodavimas



MULTIMEDIJOS
INŽINERIJOS
KATEDRA

```
// Pixel Shader (5 instruction slots)
```

```
float3 DecodeHDR_RGBE8_RGB(in float4 rgbe)
```

```
{
```

```
    float fExp = rgbe.a * 256 - EXP_OFFSET ;
```

```
    float fScale = pow(EXP_BASE, fExp) ;
```

```
    return (rgbe.rgb * fScaler) ;
```

```
}
```

```
// If R16F texture format is available,
```

```
// you can use texture to convert alpha to scale factor
```

```
float3 DecodeHDR_RGBE8_RGB(in float4 rgbe)
```

```
{
```

```
    // samp1D_Exp: 1D float texture of 256x1
```

```
    //      pow(EXP_BASE, uCoord * 256 - EXP_OFFSET)
```

```
    float fScale = tex1D(samp1D_Exp, rgbe.a).r ;
```

```
    return (rgbe.rgb * fScale) ;
```

```
}
```

Spinduliuojančios tekstūros (Self-Emission Textures)



MULTIMEDIJOS
INŽINERIJOS
KATEDRA

- Tekstūros spinduliuojantiems objektams
- Išnaudojamas alpha kanalas

$$Lum = Tex.rgb \times Tex.a \times scale$$

scale : ~16-128

- Pakankamai ryšku kad spindėtų
 - Užkoduojama offline
- Greitas dekodavimas



Renderavimas su Tone Mapping



MULTIMEDIJOS
INŽINERIJOS
KATEDRA

Blizgi atspindžio medžiaga

```
struct PS_INPUT_GlossReflect
```

```
{  
    float2 tcDecal          : TEXCOORD0 ;  
    float3 tcReflect        : TEXCOORD1 ;  
    float2 tcLightMap       : TEXCOORD2 ;  
    float2 tcFresnel        : TEXCOORD3 ;  
  
    // Exposure-scaled lighting results  
    // Use TEXCOORD to avoid clamping  
    float3 cPrimaryDiffuse : TEXCOORD6 ;  
    float3 cOtherDiffuse   : TEXCOORD7 ;  
};
```

```
float4 PS_GlossReflect(PS_INPUT_GlossReflect vIn) : COLOR0  
{  
    float4 vDecalMap = tex2D(samp2D_Decal, vIn.tcDecal) ;  
    float3 vLightMap = tex2D(samp2D_LightMap, vIn.tcLightMap) ;  
  
    float3 vDiffuse = vIn.cPrimaryDiffuse * vLightMap + vIn.cOtherDiffuse ;  
    vDiffuse *= vDecalMap ;  
  
    // HDR-decoding of environment map  
    float3 vSpecular = DecodeHDR_RGBE8_RGB(  
        texCUBE(sampCUBE_EnvMap, vIn.tcReflect) ) ;  
    float3 vRoughSpecular = texCUBE(sampCUBE_DullEnvMap, vIn.tcReflect) ;  
  
    float fReflectance = tex2D( samp2D_Fresnel, vIn.tcFresnel ).a ;  
    fReflectance *= vDecalMap.a ;  
  
    vSpecular = lerp(vSpecular, vRoughSpecular, fShininess) ;  
    float3 vLum = lerp(vDiffuse, vSpecular, fReflectance) ;  
  
    // HDR tone-mapping encoding  
    float4 vOut ;  
    vOut.rgb = vLum / (vLum + 1.0) ;  
    vOut.a = 0.0 ;  
  
    return vOut ;  
}
```




Spinduliuojanti medžiaga

```
struct PS_INPUT_SelfIllum
```

```
{  
    float2 tcDecal          : TEXCOORD0 ;  
    float2 tcLightMap       : TEXCOORD2 ;  
  
    // Exposure-scaled lighting results  
    // Use TEXCOORD to avoid clamping  
    float3 cPrimaryDiffuse : TEXCOORD6 ;  
    float3 cOtherDiffuse   : TEXCOORD7 ;  
};
```

```
float4 PS_SelfIllum(PS_INPUT_SelfIllum vIn) : COLOR0  
{  
    float4 vDecalMap = tex2D(samp2D_Decal, vIn.tcDecal) ;  
    float3 vLightMap = tex2D(samp2D_LightMap, vIn.tcLightMap) ;  
  
    float3 vDiffuse = vIn.cPrimaryDiffuse * vLightMap +  
                    vIn.cOtherDiffuse ;  
    vDiffuse *= vDecalMap ;  
  
    // HDR-decoding of self-emission texture  
    // fEmissiveScale : self-emission luminance * exposure  
    float3 vEmissive = vDecalMap.rgb * vDecalMap.a *  
                      fEmissiveScale ;  
  
    // Add the self-emission  
    float3 vLum = vDiffuse + vEmissive ;  
  
    // HDR tone-mapping encoding  
    float4 vOut ;  
    vOut.rgb = vLum / (vLum + 1.0) ;  
    vOut.a = 0.0 ;  
  
    return vOut ;  
}
```

- Nustatomi aukšto šviesio regionai

$$Glare_{src} = \max\left(\frac{Buffer.rgb - Threshold}{1 - Threshold}, 0\right)$$

Threshold : ~0.5-0.8

- Dalyba (1 - *Threshold*) normalizavimui
- Generiuojamas spindėjimas (glare)
 - Naudojamas integer buffer tekstūrų filtravimui
- Generuojamas rodomas vaizdas
 - Pridedamas spindėjimas prie frame buffer

Spindējimo kompozīcija



MULTIMEDIJOS
INŽINERIJOS
KATEDRA

```
struct PS_INPUT_Display
{
    float2 tcFrameBuffer    : TEXCOORD0 ;
    float2 tcGlare          : TEXCOORD1 ;
} ;

float4 PS_Display(PS_INPUT_Display vIn) : COLOR0
{
    float3 vFrameBuffer = tex2D(samp2D_FrameBuffer, vIn.tcFrameBuffer)
    ;
    float3 vGlare = tex2D(samp2D_Glare, vIn.tcGlare) ;

    // Add the glare
    float4 vOut ;
    vOut.rgb = vFrameBuffer + vGlare * vGlare ;
    vOut.a = 0.0 ;

    return vOut ;
}
```

Spindesio generavimas



MULTIMEDIJOS
INŽINERIJOS
KATEDRA

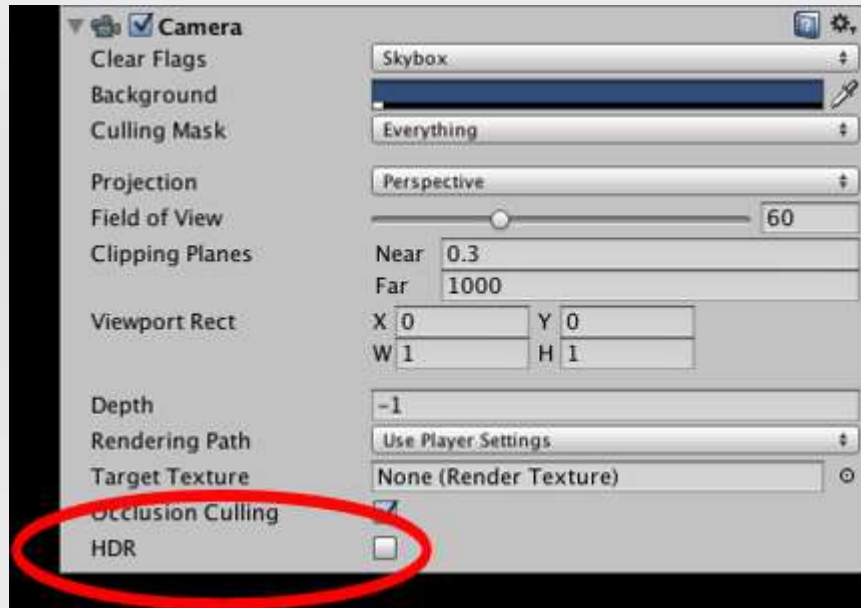
- Reikia daug renderiavimo etapų
 - Integer buferių tikslumo bėdos
 - Clamping
 - Vidurkinimo klaidos
 - 16-bit integer gali nepakakti
 - Svarbu išlaikyti tinkamą diapazoną



HDR Unity3D



MULTIMEDIJOS
INŽINERIJOS
KATEDRA



- Išbandome HDR ir Tonemapping
- Išbandome HDR ir BLOOM



Daugybiniai Gauso filtrai (Multiple Gaussian Filters)



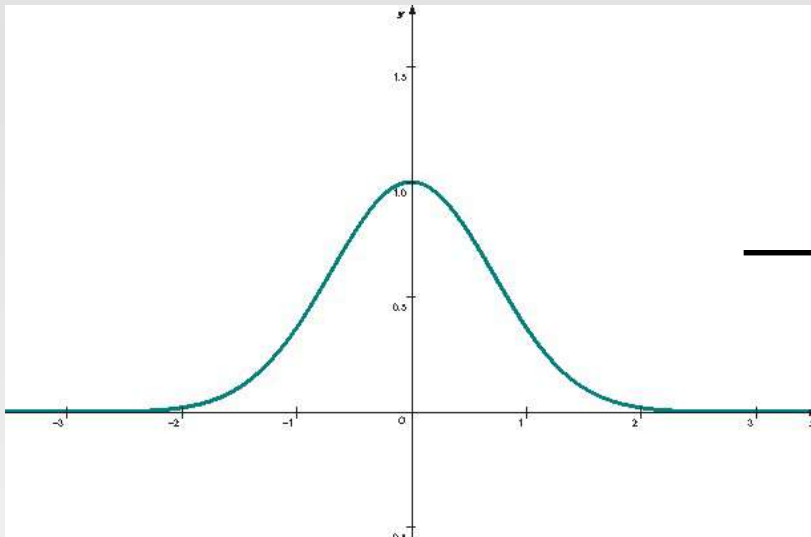
MULTIMEDIJOS
INŽINERIJOS
KATEDRA

- Bloom'o generavimas
- Pavienis Gauso filtras neduoda labai gražių rezultatų
 - Mažas veikimo plotas
 - Nepakankamai ryšku aplink šviesos poziciją
- Galima sujungti kelis – taip sukuriant daugybinius Gauso filtrus
 - Galima naudoti su skirtingu veikimo plotu
 - Atsiranda didesnio ir aštresnio (labiau) fokusuoto spindesio galimybė

Daugybiniai Gauso filtrai



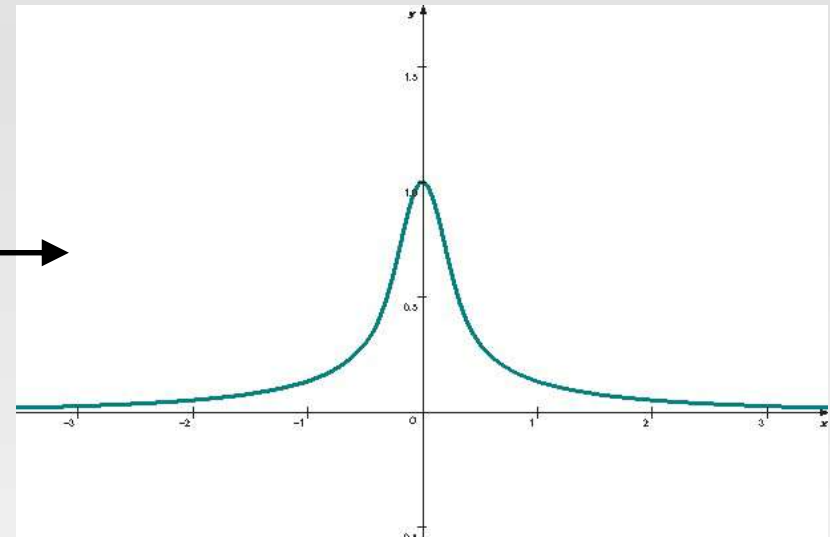
MULTIMEDIJOS
INŽINERIJOS
KATEDRA



$$e^{-r^2}$$

$$r = \sqrt{x^2 + y^2}$$

atstumas iš šaltinio



$$e^{-(0.25r)^2} + 2e^{-(0.5r)^2} + \\ 4e^{-r^2} + 8e^{-(2r)^2} + 16e^{-(4r)^2}$$

Daugybiniai Gauso filtrai



MULTIMEDIJOS
INŽINERIJOS
KATEDRA



Originalus
vaizdas



$$e^{-r^2}$$



$$e^{-(0.25r)^2} + 2e^{-(0.5r)^2} + \\ 4e^{-r^2} + 8e^{-(2r)^2} + 16e^{-(4r)^2}$$

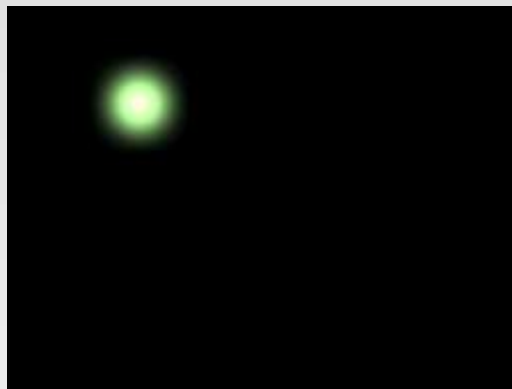


- Plataus diametro filtras valgo daug resursų
 - Didelis diametras reiškia stiprų žemos eilės (low-pass) filtrą
 - Rerusus galima taupyti panaudojus blur filtrą žemos rezoliucijos paveikslo versijai ir padidinti jį per bilinear filtravimą → Klaidos ir artifaktai praktiškai nematomi
 - Keiskite paveikslo rezoliuciją, o ne filtro diametrą
 - $1/4 \times 1/4$ (1/16 resursų)
 - $1/8 \times 1/8$ (1/64 resursų)
 - $1/16 \times 1/16$ (1/256 resursų)
 - $1/32 \times 1/32$ (1/1024 resursų)
 - ...

Daugybiniai Gauso filtrai



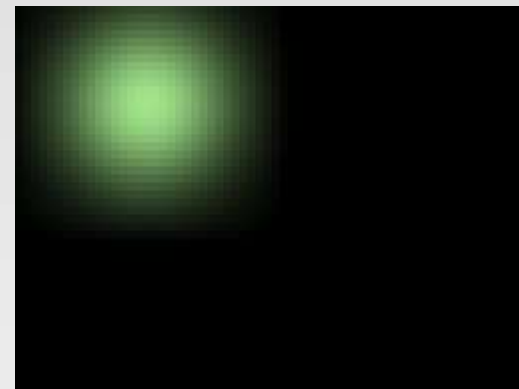
MULTIMEDIJOS
INŽINERIJOS
KATEDRA



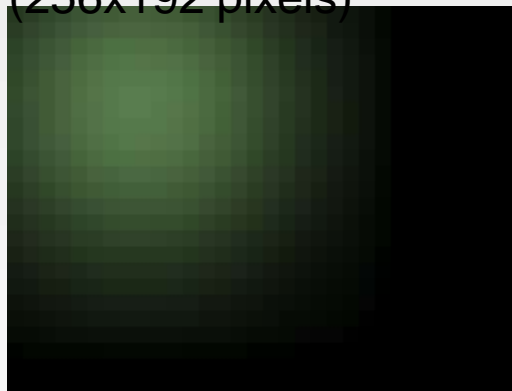
$1/4 \times 1/4$
(256x192 pixels)



$1/8 \times 1/8$
(128x96 pixels)



$1/16 \times 1/16$
(64x48 pixels)

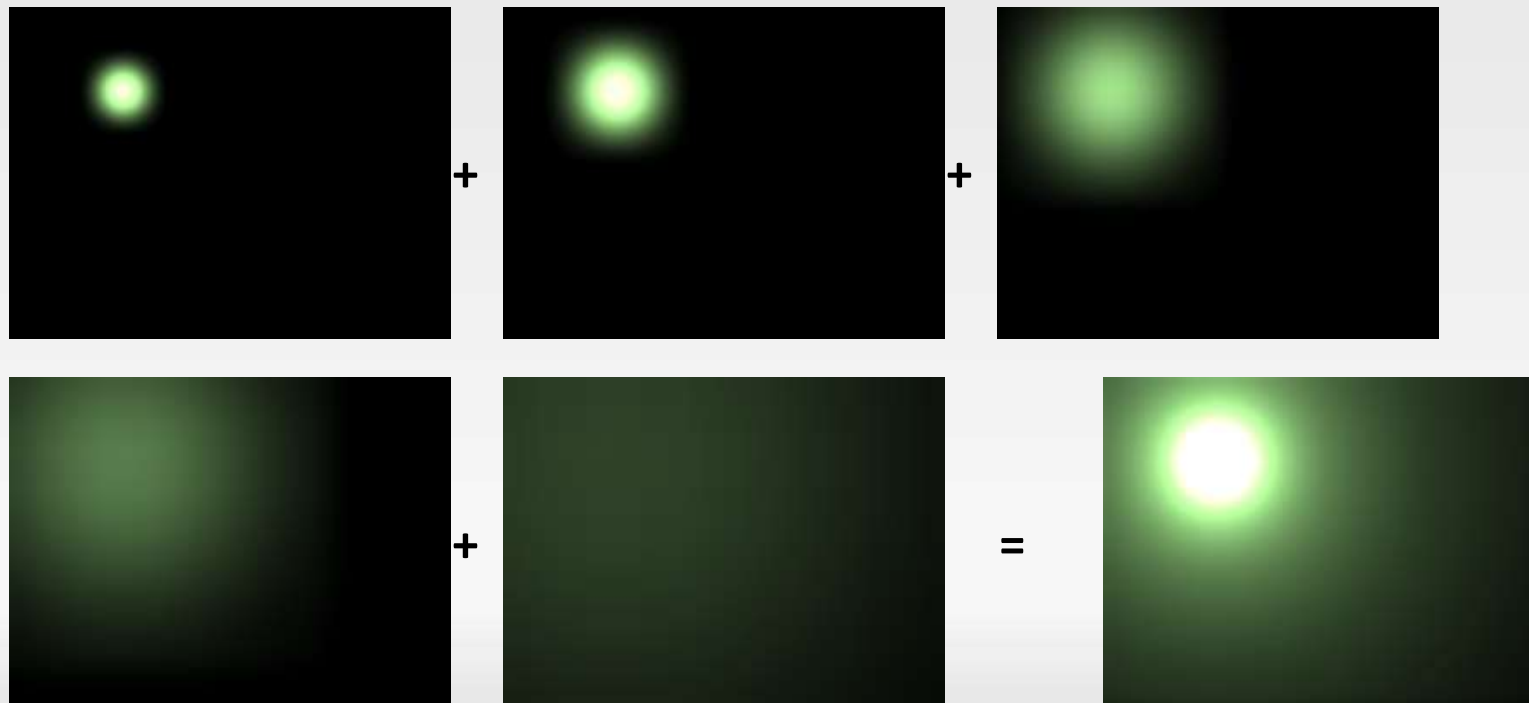


$1/32 \times 1/32$
(32x48 pixels)



$1/64 \times 1/64$
(16x12 pixels)

- Galima išdidinti su bilinear filtravimu ir apjungti rezultatus
 - Klaida pakankamai maža



Rezultatas pakankamai geras!



MULTIMEDIJOS
INŽINERIJOS
KATEDRA



Blur shader (Pro Only)

♥ Add to Wish List

Category: Shaders

Publisher: Anamaria Todor

Rating: ★★★★★ (158)

Price: Free

Open in Unity



Requires Unity 4.1.0 or higher.

As Unity only provides blur as a post-processing effect, I developed a simple shader that would display the image behind it with a gaussian blur effect applied to it. The intensity of the effect can be tweaked using a slider.

IF YOU ARE SEEING THE DEFAULT PINK SHADER: This Shader uses GrabPass for taking the image behind it. GrabPass internally uses RenderTextures, which are a Unity Pro feature. So unless you have Unity Pro, I'm afraid you will not be able to use this shader.

For a demonstration, access my blog.



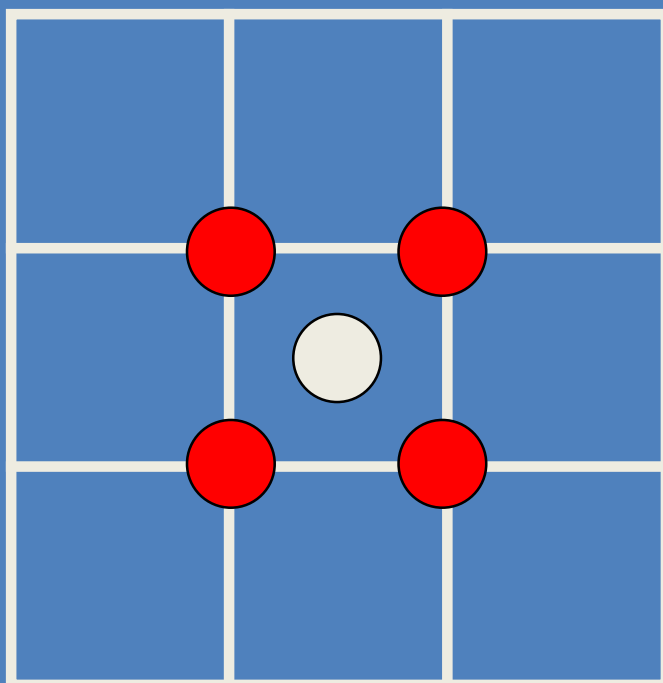


- 2D Gauso filtras gali būti išskaidytas į du 1D filtrus x ir y kryptimis
 - Galima efektyviai implementuoti dviem etapais
 - Kaip:
 - Mitchell, Jason L., [*“Real-Time 3D Scene Post-Processing”*](#)
- Naudojamas aukšto tikslumo formatas žemos rezoliucijos buferiams
- Dauga atskaitų valgo daug resursų
 - Ypač aukštos rezoliucijos grafikai
 - Galima panaudoti vieno etapo kūgio filtrą

Kūgio (Cone) filtras



MULTIMEDIJOS
INŽINERIJOS
KATEDRA



1/16	2/16	1/16
2/16	4/16	2/16
1/16	2/16	1/16



Pikselis kurį reikia surenderiuoti



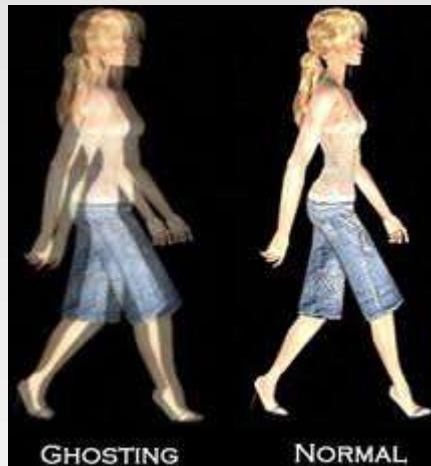
Tekstūros atskaitos taškai

Kiti blizgesio tipai



MULTIMEDIJOS
INŽINERIJOS
KATEDRA

- Afterimage
- Stars (light streaks)
- Ghosts



- Plačiau:
 - Kawase, Masaki, [“Frame Buffer Postprocessing Effects in DOUBLE-S.T.E.A.L \(Wreckless\)”](#)



- Vaizdų apdorojimas
 - Nuskaitomi aukšto šviesio (high-luminance) regionai iš HDR frame buffer
 - Image-based post-processing
- Spraitai
 - Tikrinamas šviesos šaltinių matomumas
 - Pagal geometrijos informaciją
 - Galima nustatyti matomumą pagal surenderiuotų pikselių skaičių
 - Kiekvienam matomam šviesos šaltiniui pašomi spindesio spraitai

Vaizdų apdorojimas



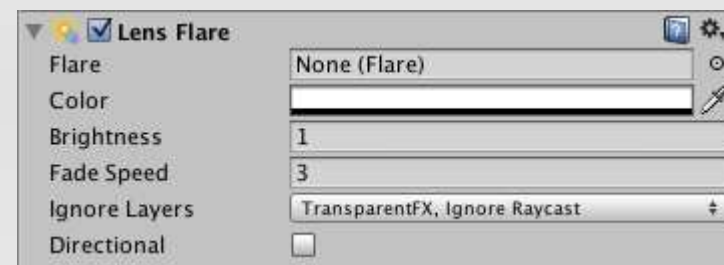
MULTIMEDIJOS
INŽINERIJOS
KATEDRA



Spraitai



MULTIMEDIJOS
INŽINERIJOS
KATEDRA



- **Vaizdų apdorojimas**
 - Galima panaudoti atspindimoms šviesoms (reflected lights) ir tam tikrą plotą apšviečiančioms šviesoms (area lights)
 - Resursai neproporcingi šviesų šaltinių skaičiui
 - Atspindžio formą galima pareguliuoti naudojant skirtingus filtrus
- **Spraitai**
 - Aukštos kokybės renderiavimas
 - Neribotas dinaminis diapazonas
 - Mažai resursų prie mažai šviesos šaltinių ir atv.

- Vaizdų apdorojimas
 - Pixel filling valgo resursus
 - Aliasingo artifaktai
 - Itin pastebima mažiems šviesos šaltiniams
 - Ribotas dinaminis diapazonas
 - Sunku valdyti labai šviesias ir labai fokusuotas (ryškias) šviesas
- Spraitai
 - Resursų naudouja proporcingai šviesos šaltinių skaičiui
 - Sunku panaudoti arbitriškai suformuotas šviesas
 - Sunku pritaikyti netiesiogines šviesas
 - Neišeina padaryti akinamo atspindžio



- Vaizdų apdorojimas
 - Area light
 - Arbitriškai suformuoti šviesos šaltiniai
 - Šviesos šaltiniai su dideliu projektuojamu plotu
 - Atspindima šviesa
 - Daug šviesų šaltinių
- Spraitai
 - Labai ryški ir aiški šviesa (spindulys)
 - Šviesos šaltiniai su mažu apšviečiamu plotu
 - Mažas šviesos šaltinių skaičius

Ką rinktis pagal šviesos tipą



MULTIMEDIJOS
INŽINERIJOS
KATEDRA

- Vaizdų apdorojimas gerai tinka:
 - Apšvietimui (Illuminations) / neono ženklams
 - Pastatų langams nakties metu
 - Daugelio mažų šviesų šaltinių grupei
 - Šviesai atspindimai nuo paviršiaus
- Spratai tinka:
 - Saulei
 - Gatvių lempoms
 - Automobilio žibintams
- Nors netolimoje ateityje... Vaizdų apdorojimas bus naudojamas viskam...

Unity HDRI Pack

Unity Essentials/Asset Packs
Unity Technologies

Not enough ratings

Free


[Open in Unity](#)




Requires Unity 5.5.0 or higher.

Unity HDRI Pack

This is a pack of 7 LatLong, 8192x4096 HDR images shot in different locations around the world.

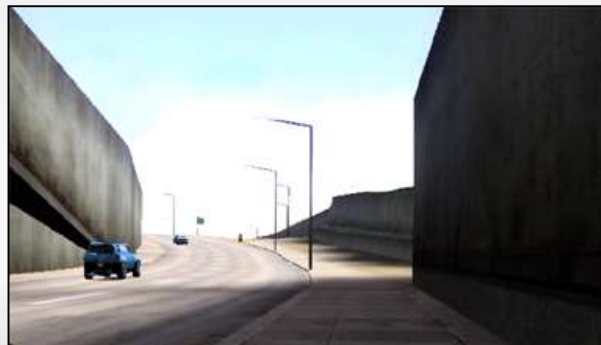
These images were captured using the method described in an "Artist-Friendly Workflow for Panoramic HDRI" presented at





□ <https://www.assetstore.unity3d.com/en/#!/content/72511>

- Vaizdas lauke žiūrint iš tamsios patalpos atrodo labai ryškus ir šviesus
- Tuo pačiu kambario vidus atrodo labai tamsus
- Tamsos adaptacija ir šviesos adaptacija?
 - Trumpam nieko nematome įėję į tamsų kambarį
 - Trumpam esam apakinti vėl išėję į ryškios saulės nušviestą lauką (ypač žiemą ☺)

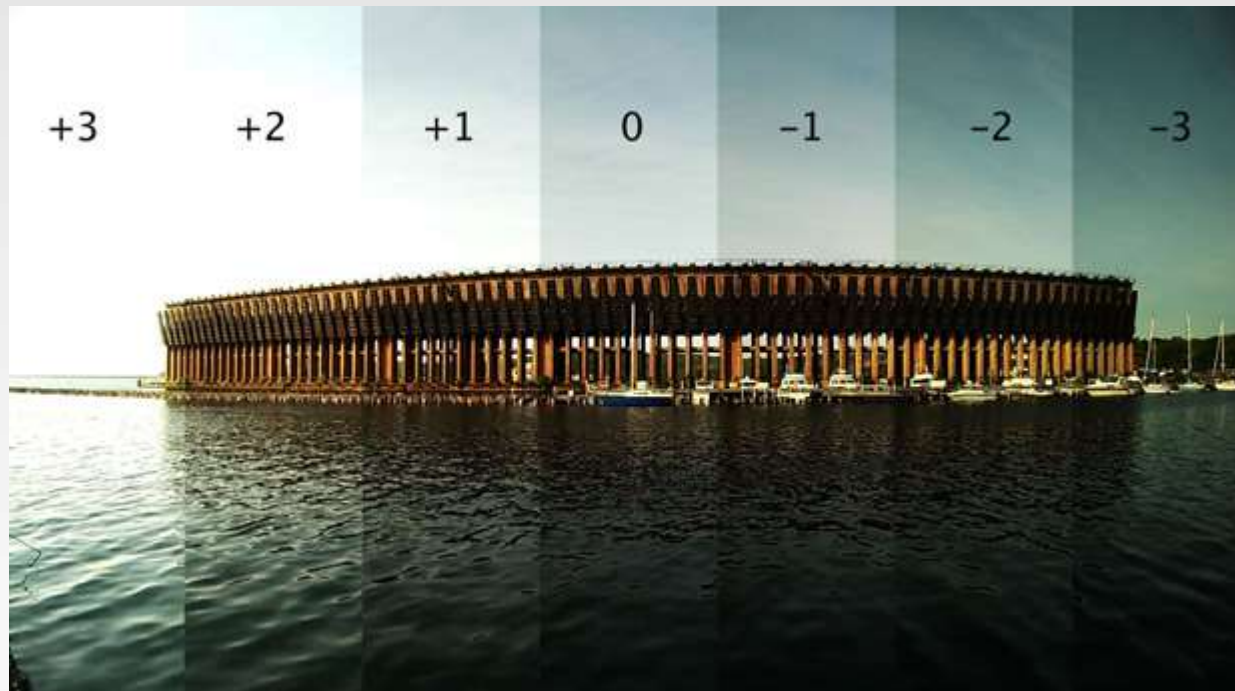


Ekspozicijos valdymas



MULTIMEDIJOS
INŽINERIJOS
KATEDRA

- Reikia nustatyti vidutinį scenos apšvietimą
- Suskaičiuoti teisingą ekspozicijos lygį
- Sureguliuoti ekspozicijos lygį



- Down-sample frame buffer
 - Mažinam iki 16x16 ar mažiau
- Verčiam HDR buffer'į į skaisčius (luminances)
 - Jei naudojame alfa kaip papildomą informaciją:

$$Lum(x, y) = \frac{Buffer(x, y).rgb \times (1 + Buffer(x, y).a \times Scale)}{Exp_{current}}$$

- Jei naudojame tone mapping RGB spalvų kompresijai:

$$Lum(x, y) = \frac{Buffer(x, y).rgb}{1 + \delta - Buffer(x, y).rgb} \times \frac{1}{Exp_{current}}$$

- Skaičiuojam vidutinį apšvietimą (skaistį) per visą ekraną

$$Lum_{avg} = \exp\left(\frac{1}{N} \sum_{x,y} \log(\delta + Lum(x, y))\right)$$

δ : maža konstanta, kad negauti neapibrėžtumo

- Plačiau:
 - DirectX 9.0 SDK Summer 2003 Update, [“HDRLighting Sample”](#)
 - Reinhard, Erik, Mike Stark, Peter Shirley, and Jim Ferwerda. [“Photographic Tone Reproduction for Digital Images”](#)
- Trial and error procesas 😊

Teisinga ekspozicija



MULTIMEDIJOS
INŽINERIJOS
KATEDRA

$$Exp = \frac{Scale}{Lum_{avg}}$$

Exp : tinkamas ekspozicijos lygis
scenai

Scale : Arbitrinis skalės faktorius



- Po truputį keičiam:

$$Exp_{new} = Exp_{current} + (Exp - Exp_{current}) \times Ratio$$

Ratio = adaptavimo greitis (~0.02-0.1)



- Trial and error
 - Skirtingas greitis adaptacijai prie šviesos ir tamsos
 - Ekspozicija skaičiuojama pagal tinkamą dydį, pvz. 0.5 s prieš
 - Derinamas jautrumas

$$\text{LogExp} = \log(\text{Exp})$$

$$\text{LogExp}_{\text{current}} = \log(\text{Exp}_{\text{current}})$$

$$\text{LogExp}_{\text{new}} = \text{LogExp}_{\text{current}} + (\text{LogExp} - \text{LogExp}_{\text{current}}) \times \text{Ratio}$$

$$\text{Exp}_{\text{new}} = \exp(\text{Base} + (\text{LogExp}_{\text{new}} - \text{Base}) \times \text{Sensitivity})$$

Base : Bazinio ekspozicijos lygio logaritmas

Sensitivity : adaptavimo santykis (~0-1)

1. Tonemapping efektas
2. Derinam exposure parametraž
3. Bandome šeiderį (moodle)



HDR žaidimuose



MULTIMEDIJOS
INŽINERIJOS
KATEDRA

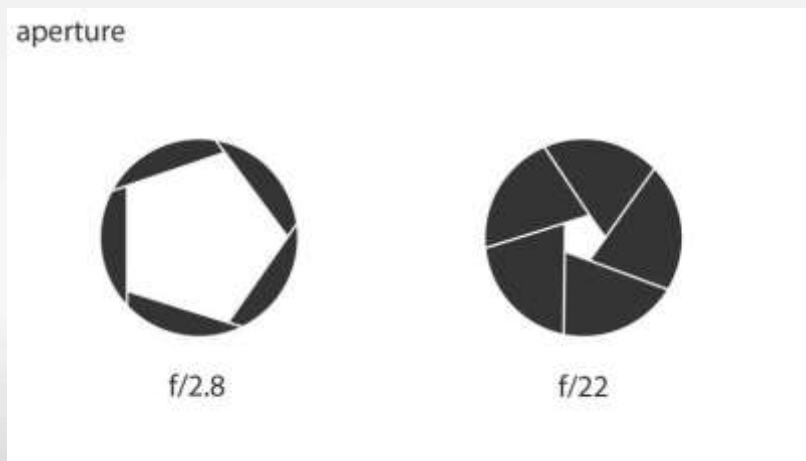


<https://www.youtube.com/watch?v=BDIXSouz5LA>

- Turi būti grafiškai patrauklu nei realistiškai tikslu
 - Tikslumas auditorijos daugumai nesvarbus
 - Tikslu = nebūtinai gražu
- Kaina
 - Svarbu suprasti HDR efektus
 - Naudoti fake HDR kur galima
 - Pvz., Panaudoti spraitus saulės atspindžiams
 - Naudoti HDR efektus kur reikia sukurti įspūdį

- Integer formatai turi ribotą dinaminį diapazoną
 - Renderiuojama reliatyvaus šviesios erdvėje
 - Renderiavimo metu panaudojamas ekospozicijos dydžių kaitaliojimas
 - Aukštas tikslumas nebūtinai tamsiems regionams
 - Šie regionai ir lieka tamsūs galutiname vaizde
 - Svarbu pasirinkti efektyvų diapazoną

- Visas operacijas galima daryti float
 - Environment maps / frame buffers / glare generation
 - Renderiavimas absoliutaus šviesios erdvėje
 - Paveikslu paremtas spindesio generavimas
- Efektyvus HDR panaudojimas
 - Depth-of-field pagal diafragmą
 - Motion blur su aukštu nenukarpytu šviesiu



- <http://devlog.foureyedcat.com/post/129500666432/double-exposure-effect-in-unity-with-the-standard>
- Reinhard, Erik, Mike Stark, Peter Shirley, and Jim Ferwerda, [*“Photographic Tone Reproduction for Digital Images”*](#)
- Mitchell, Jason L., [*“Real-Time 3D Scene Post-Processing”*](#)
- DirectX 9.0 SDK, [*“HDRLighting Sample”*](#)
- Debevec, Paul E., [*“Paul Debevec Home Page”*](#)
- Kawase, Masaki, [*“Frame Buffer Postprocessing Effects in DOUBLE-S.T.E.A.L \(Wreckless\)”*](#)