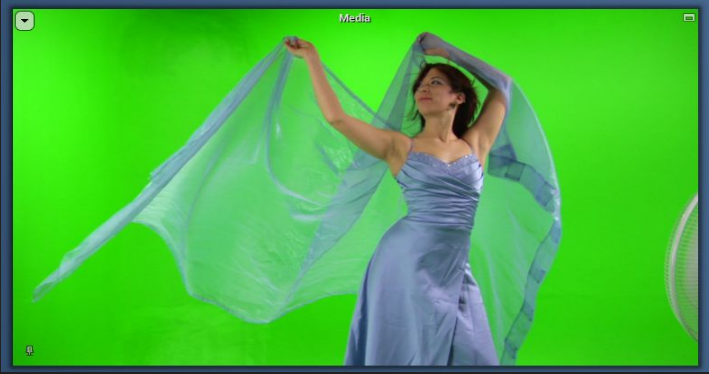
CX

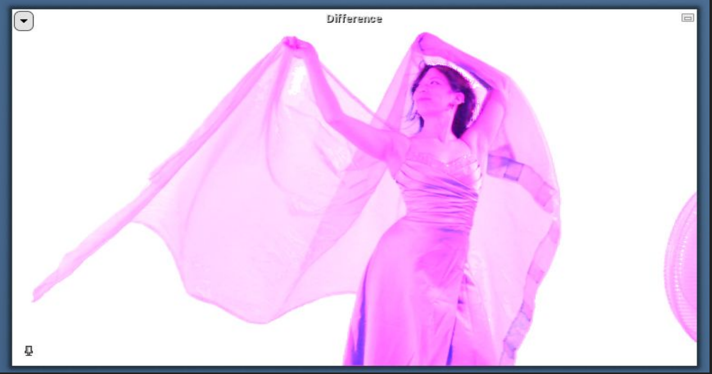
Composure X

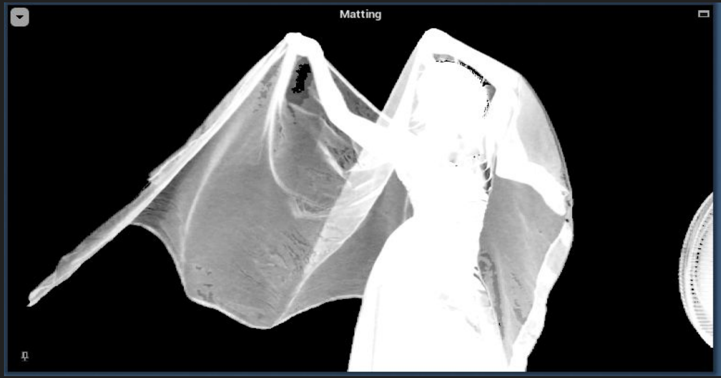
Realtime Compositing Workflow

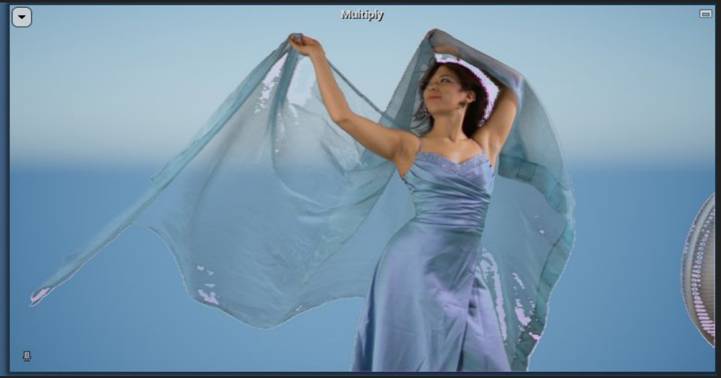
@dpredie

Building real-time compositing tools?



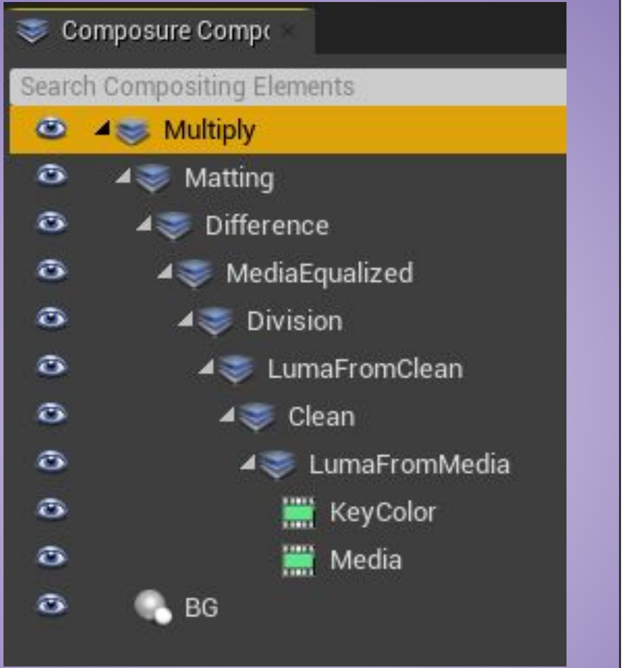






@dpredie

Layers or Passes?



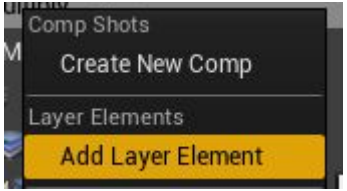
Composure Compo

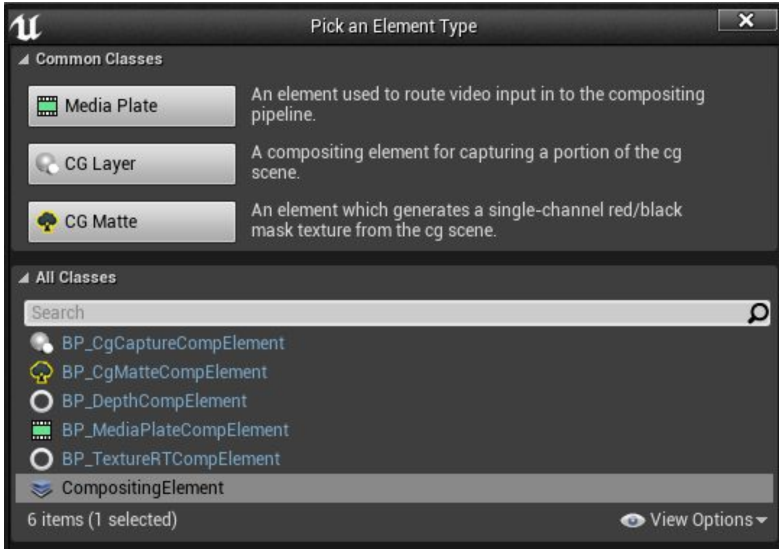
Search Compositing Elements

- Multiply
  - Matting
    - Difference
      - MediaEqualized
        - Division
          - LumaFromClean
            - Clean
              - LumaFromMedia
                - KeyColor
                - Media
- BG

@dpredie

Layers:





Pick an Element Type

Common Classes

Media Plate	An element used to route video input in to the compositing pipeline.
CG Layer	A compositing element for capturing a portion of the cg scene.
CG Matte	An element which generates a single-channel red/black mask texture from the cg scene.

All Classes

Search

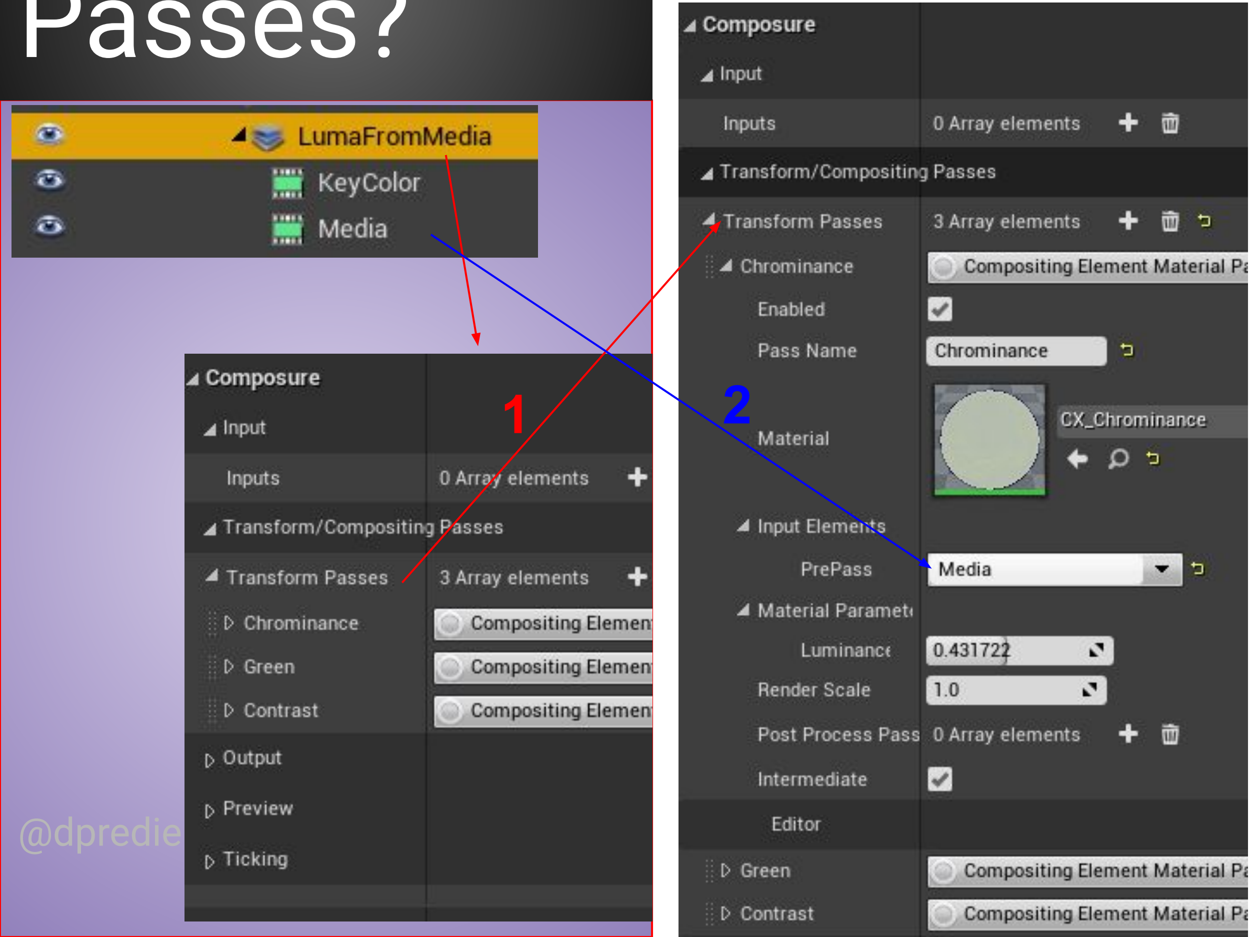
- BP_CgCaptureCompElement
- BP_CgMatteCompElement
- BP_DepthCompElement
- BP_MediaPlateCompElement
- BP_TextureRTCompElement
- CompositingElement

6 items (1 selected)

View Options

Layers or Passes?

Passes:



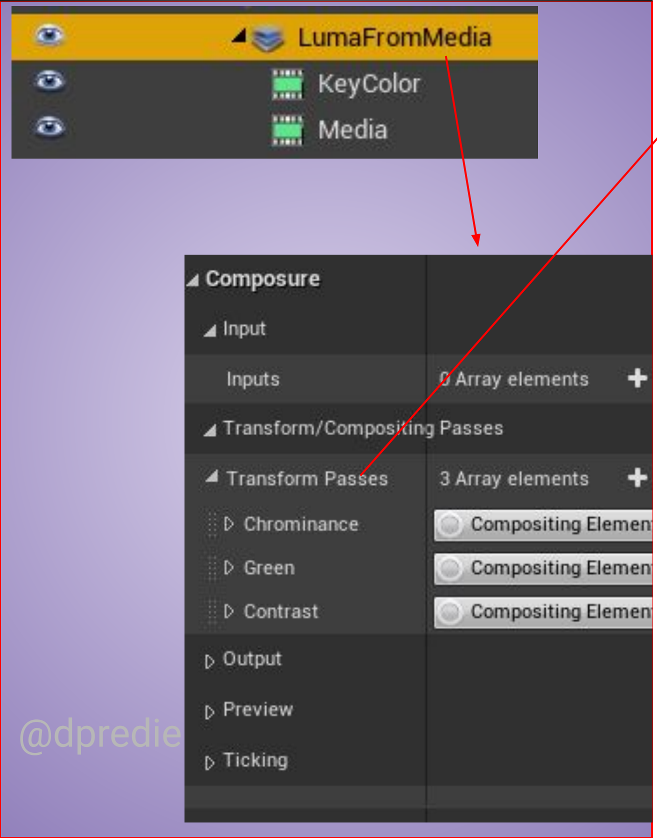
Composure
- Input
  - Inputs: 0 Array elements
- Transform/Compositing Passes
  - Transform Passes: 3 Array elements
    - Chrominance: Compositing Elemen
    - Green: Compositing Elemen
    - Contrast: Compositing Elemen
- Output
- Preview
- Ticking

Composure
- Input
  - Inputs: 0 Array elements
- Transform/Compositing Passes
  - Transform Passes: 3 Array elements
    - Chrominance: Compositing Element Material Pa
      - Enabled: ☑
      - Pass Name: Chrominance
      - Material: CX_Chrominance
      - Input Elements
        - PrePass: Media
      - Material Paramet
        - Luminance: 0.431722
      - Render Scale: 1.0
      - Post Process Pass: 0 Array elements
      - Intermediate: ☑
      - Editor
    - Green: Compositing Element Material Pa
    - Contrast: Compositing Element Material Pa

1. Each Layer can have multiple passes

2. Each Pass can access result of any child Layers (but not passes inside it)

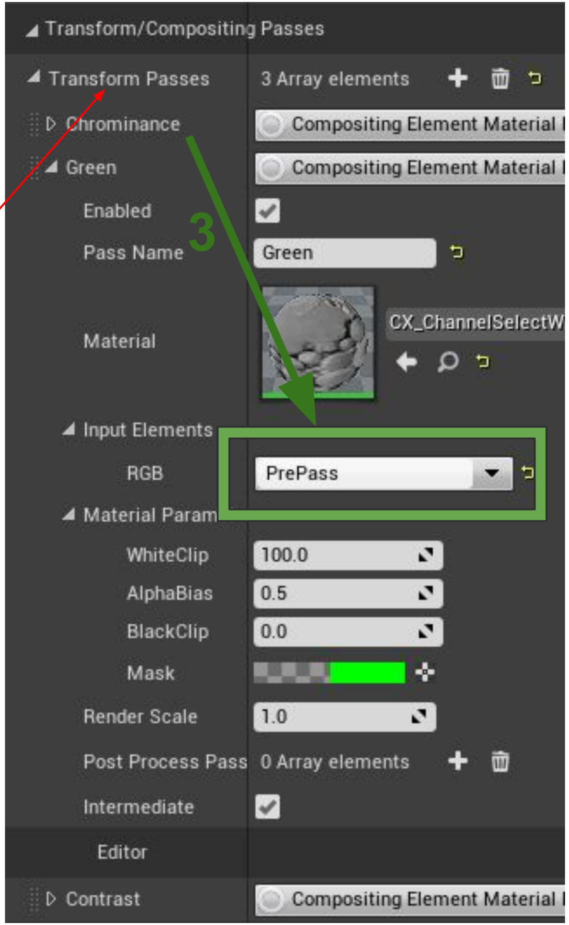
Layers or Passes?



LumaFromMedia
KeyColor
Media

Composure
Input
Inputs 0 Array elements
Transform/Compositing Passes
Transform Passes 3 Array elements
Chrominance Compositing Elemen
Green Compositing Elemen
Contrast Compositing Elemen
Output
Preview
Ticking

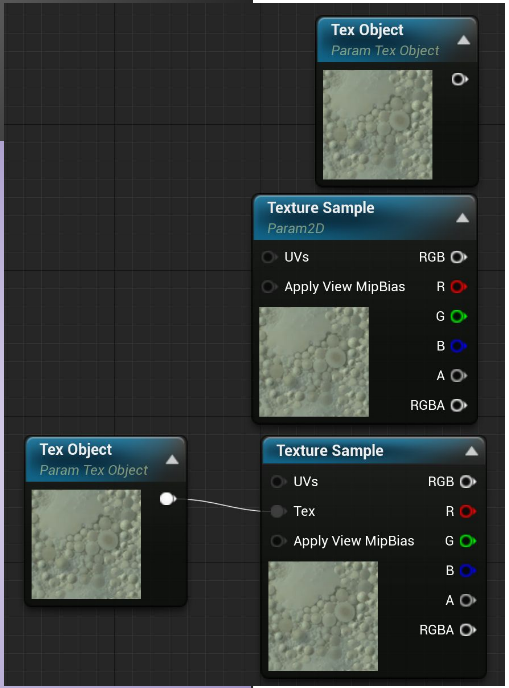
Passes:



Transform/Compositing Passes
Transform Passes 3 Array elements
Chrominance Compositing Element Material
Green Compositing Element Material
Enabled ✓
Pass Name Green
Material CX_ChannelSelectW
Input Elements
RGB PrePass
Material Param
WhiteClip 100.0
AlphaBias 0.5
BlackClip 0.0
Mask 🟩
Render Scale 1.0
Post Process Pass 0 Array elements
Intermediate ✓
Editor
Contrast Compositing Element Material

1. Each Layer can have multiple passes
2. Each Pass can access result of any child Layers (but not passes inside it)
3. Each Pass can access result of 1 pass preceding it

Why Build Multi Layers?



Tex Object:
Full Texture information (can loop/access neighboring pixel)

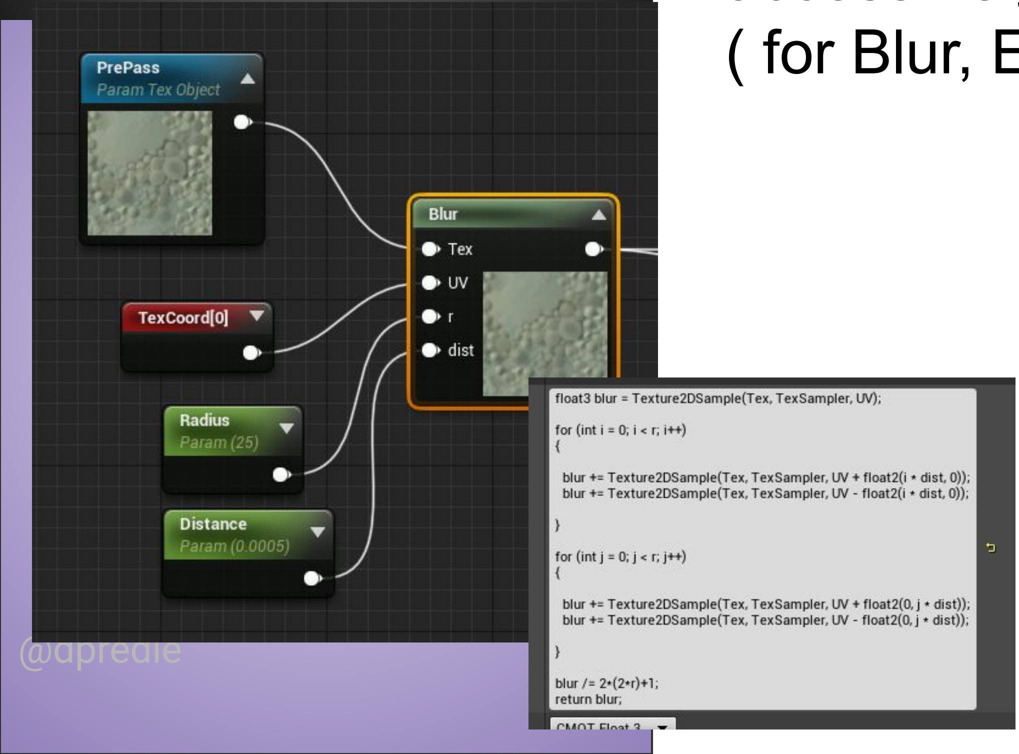
Texture Sample:
single pixel's color (cannot access neigboring pixel for manipulation)

You can generate Sample from Tex Obj, but you cannot make Tex Obj from a sample in single Blueprint.

Why Build Multi Layers?

Tex Object

With Tex Object Parameter you can access neighboring pixel for manipulation (for Blur, Erode, etc).

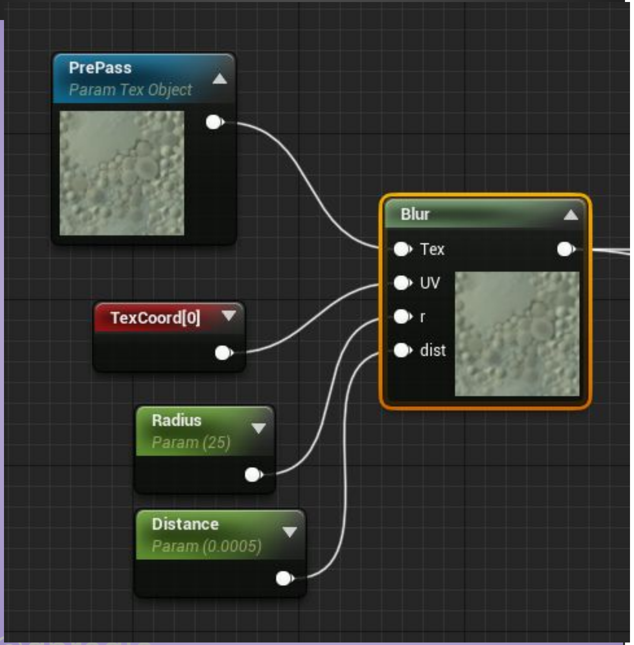


```
float3 blur = Texture2DSample(Tex, TexSampler, UV);

for (int i = 0; i < r; i++)
{

  blur += Texture2DSample(Tex, TexSampler, UV + float2(i * dist, 0));
  blur += Texture2DSample(Tex, TexSampler, UV - float2(i * dist, 0));

}

for (int j = 0; j < r; j++)
{

  blur += Texture2DSample(Tex, TexSampler, UV + float2(0, j * dist));
  blur += Texture2DSample(Tex, TexSampler, UV - float2(0, j * dist));

}

blur /= 2*(2*r)+1;
return blur;
```

result of these mathematical nodes are Texture Sample, so you cannot perform multiple Tex Object manipulation in 1 Material BP

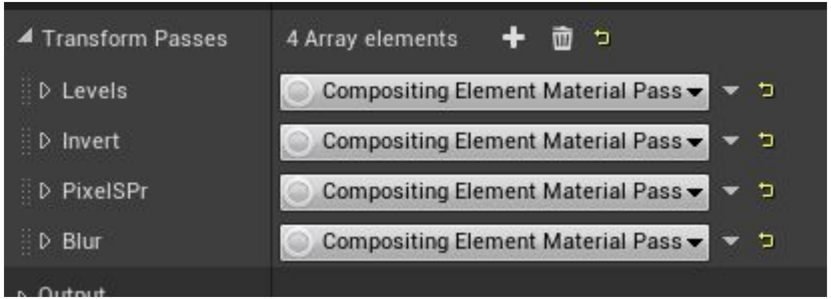
Why Build Multi Layers?

Tex Object



For compositing workflow you need to do multiple manipulation passes (preblur → Erode → blur, etc), which is not possible within single BP

as each Layer/Pass will output another Tex Object available for the next Layer/Pass to manipulate

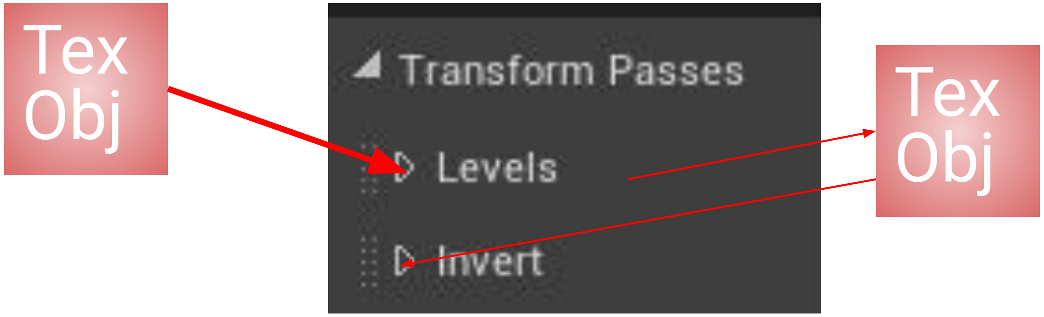


Transform Passes	4 Array elements
Levels	Compositing Element Material Pass
Invert	Compositing Element Material Pass
PixelSPr	Compositing Element Material Pass
Blur	Compositing Element Material Pass
Output	

Why Build Multi Layers?

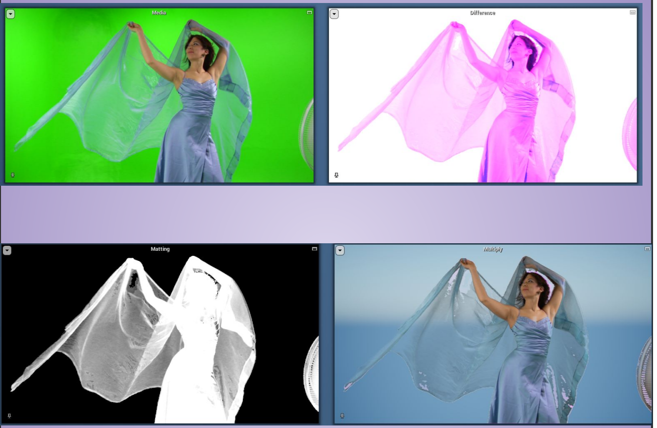
@dpredie

Tex Object



… and so on

Why Build Multi Layers?



Concerns on Performance

This workflow is best for **prototyping** the keyer/compositor

you have instant feedback on the layers and can go back to enable/disable each layer/passes

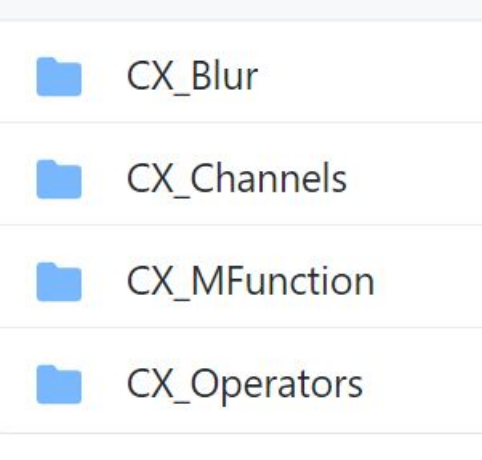
Once finalized, *you can build the compositor in fewer materials to optimize performance* (fewer renders to Tex Obj)

Compositing in Composure

CX - Workflow

https://github.com/dpredie/ComposureX

Copy Paste Content/ComposureX folder to Windows Explorer UE Project Content/

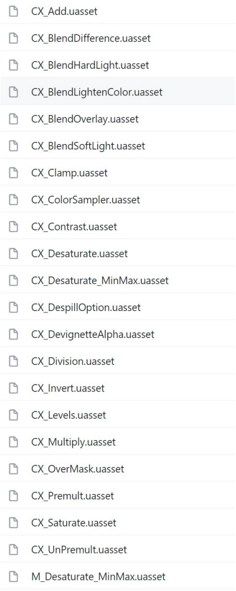


CX_Blur

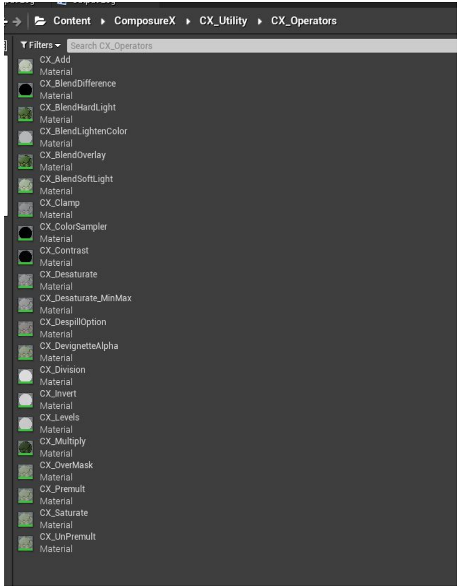
CX_Channels

CX_MFunction

CX_Operators



CX_Add.uasset
CX_BlendDifference.uasset
CX_BlendHardLight.uasset
CX_BlendLightenColor.uasset
CX_BlendOverlay.uasset
CX_BlendSoftLight.uasset
CX_Clamp.uasset
CX_ColorSampler.uasset
CX_Contrast.uasset
CX_Desaturate.uasset
CX_Desaturate_MinMax.uasset
CX_DespillOption.uasset
CX_DevignetteAlpha.uasset
CX_Division.uasset
CX_Invert.uasset
CX_Levels.uasset
CX_Multiply.uasset
CX_OverMask.uasset
CX_Premult.uasset
CX_Saturate.uasset
CX_UnPremult.uasset
M_Desaturate_MinMax.uasset



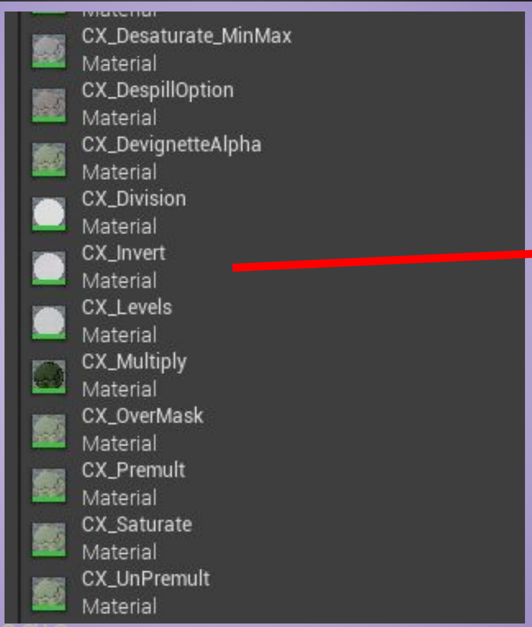
CX_Add
Material
CX_BlendDifference
Material
CX_BlendHardLight
Material
CX_BlendLightenColor
Material
CX_BlendOverlay
Material
CX_BlendSoftLight
Material
CX_Clamp
Material
CX_ColorSampler
Material
CX_Contrast
Material
CX_Desaturate
Material
CX_Desaturate_MinMax
Material
CX_DespillOption
Material
CX_DevignetteAlpha
Material
CX_Division
Material
CX_Invert
Material
CX_Levels
Material
CX_Multiply
Material
CX_OverMask
Material
CX_Premult
Material
CX_Saturate
Material
CX_UnPremult
Material

@dpredie

Compositing in Composure

CX - Utilities

Each CX Utilities can be used in Composure Layers/Passes as Materials similar to offline compositor nodes



Material

CX_Desaturate_MinMax
Material

CX_DespillOption
Material

CX_DevignetteAlpha
Material

CX_Division
Material

CX_Invert
Material

CX_Levels
Material

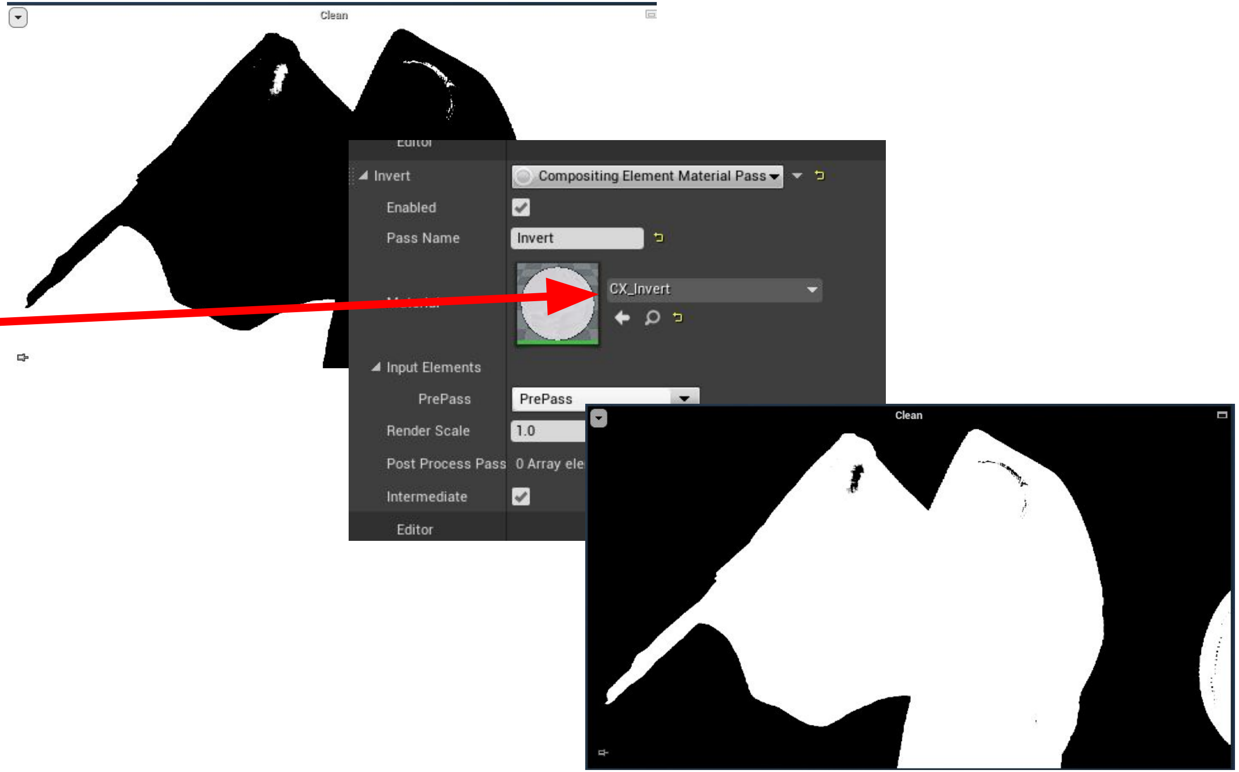
CX_Multiply
Material

CX_OverMask
Material

CX_Premult
Material

CX_Saturate
Material

CX_UnPremult
Material

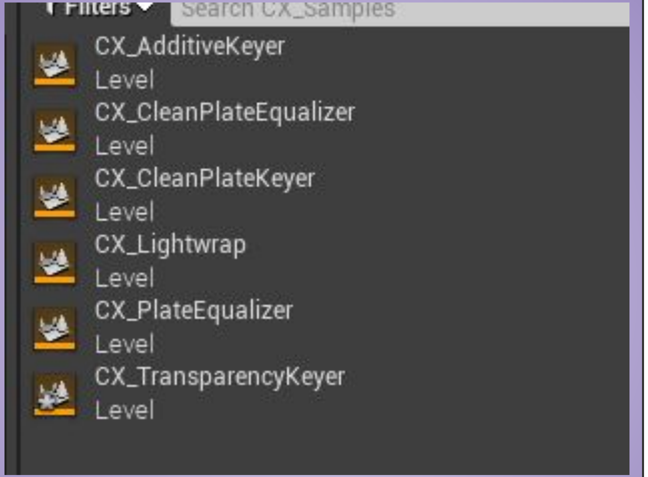


Editor

Invert	Compositing Element Material Pass
Enabled	☑
Pass Name	Invert
Material	CX_Invert
Input Elements	
PrePass	PrePass
Render Scale	1.0
Post Process Pass	0 Array ele
Intermediate	☑

Editor

CX - Samples



Filters
Search CX_Samples

CX_AdditiveKeyer
Level

CX_CleanPlateEqualizer
Level

CX_CleanPlateKeyer
Level

CX_Lightwrap
Level

CX_PlateEqualizer
Level

CX_TransparencyKeyer
Level

@dpredie

Included Samples:

1. Additive Keyer

2. Screen Equalizers
   a. Clean Plate Equalizer (if Clean Plate image is provided)
   b. Plate Equalizer (Clean Plate estimated)

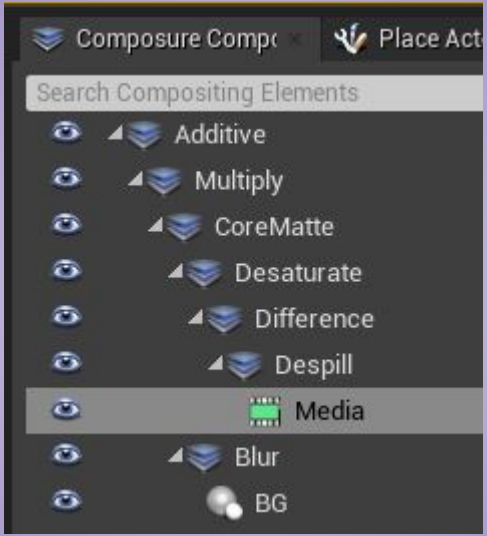
3. Clean Plate Keyer

4. Transparency Keyer (based on 2.b)

5. Lightwrap

Compositing in Composure

CX - Additive Keyer



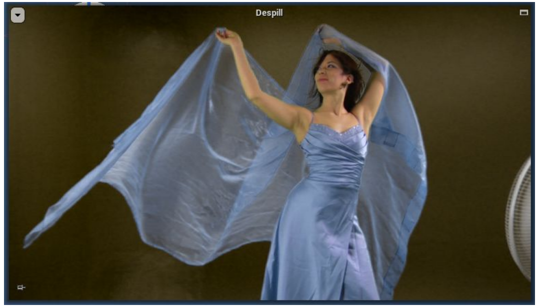
Composure Compo | Place Act

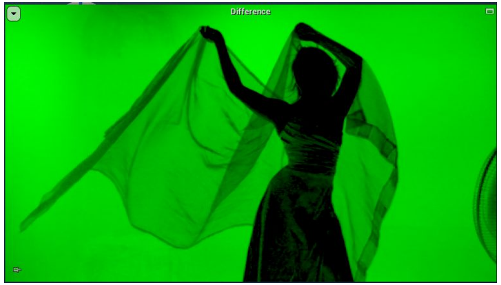
Search Compositing Elements

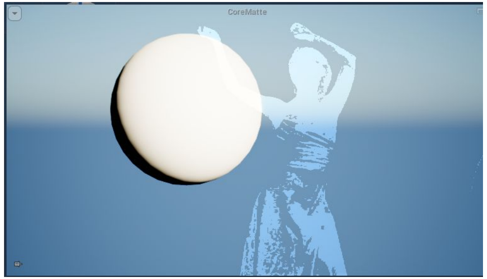
- Additive
  - Multiply
    - CoreMatte
      - Desaturate
        - Difference
          - Despill
            - Media
      - Blur
        - BG

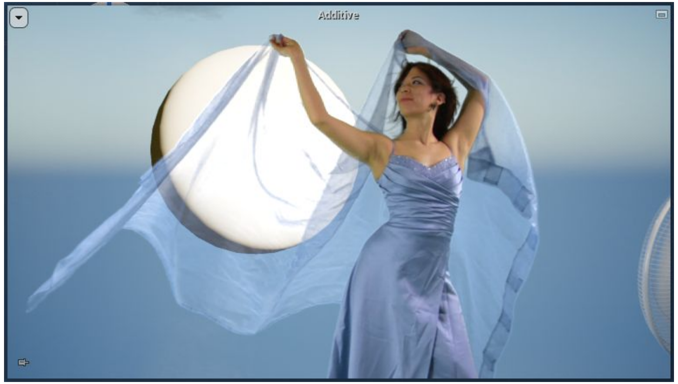
@dpredie

Keying without Alpha Channel, also known as Straight Dry Key/Despill Based Keyer







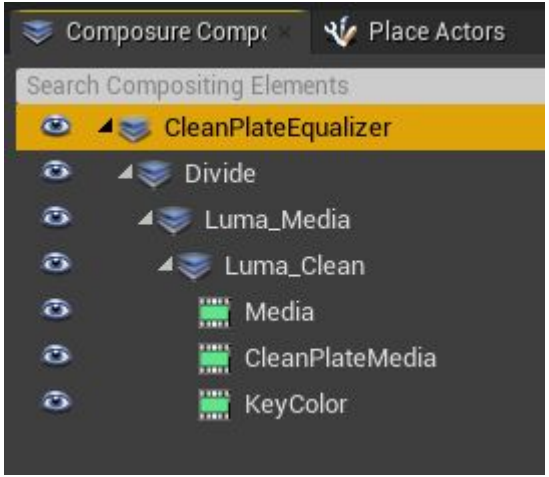


Compositing in Composure

@dpredie

CX - Clean Plate Equalizer

1. Provide Clean Plate Image & Equalize uneven background (Luma Based)

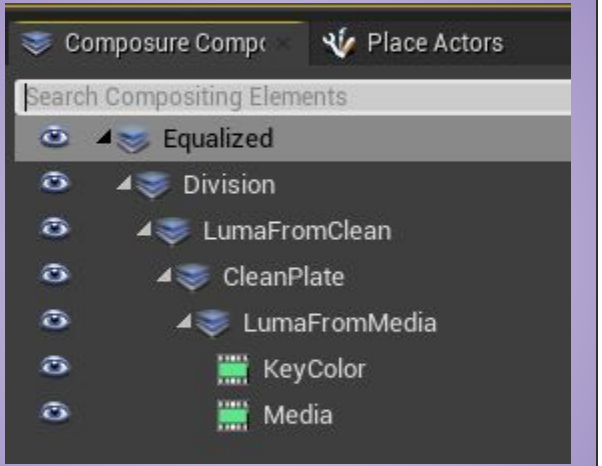


- CleanPlateEqualizer
  - Divide
    - Luma_Media
      - Luma_Clean
        - Media
        - CleanPlateMedia
        - KeyColor

Compositing in Composure

CX - Plate Equalizer

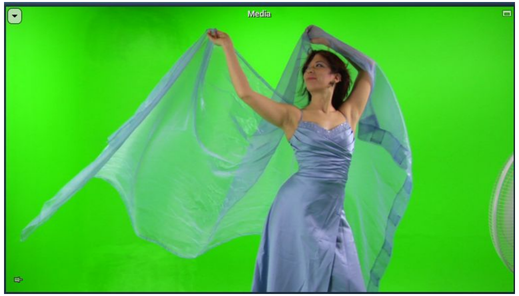
2. Estimate Clean Plate Image & Equalize uneven background (Luma Based)

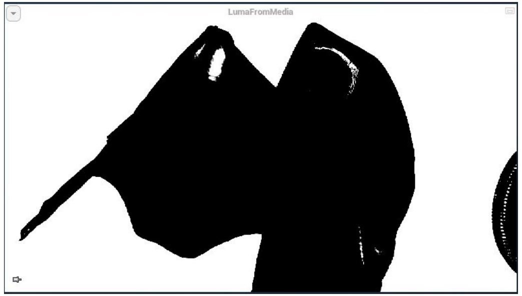


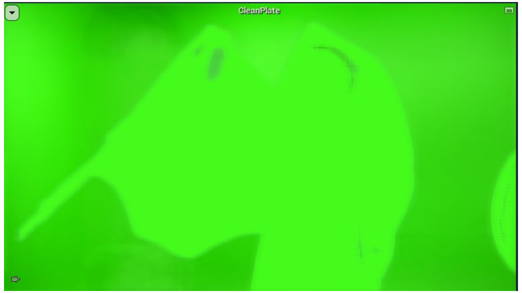
Composure Compo Place Actors

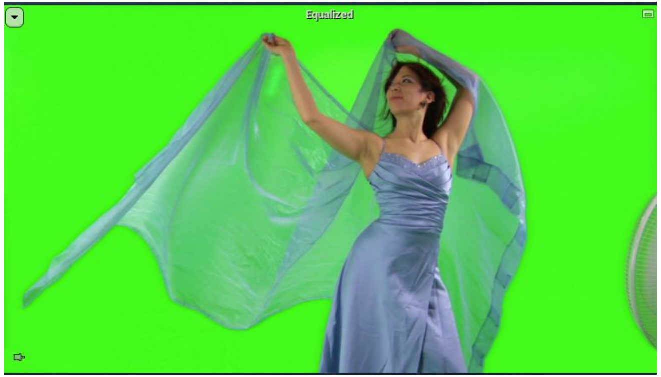
Search Compositing Elements

- Equalized
  - Division
    - LumaFromClean
      - CleanPlate
        - LumaFromMedia
          - KeyColor
          - Media







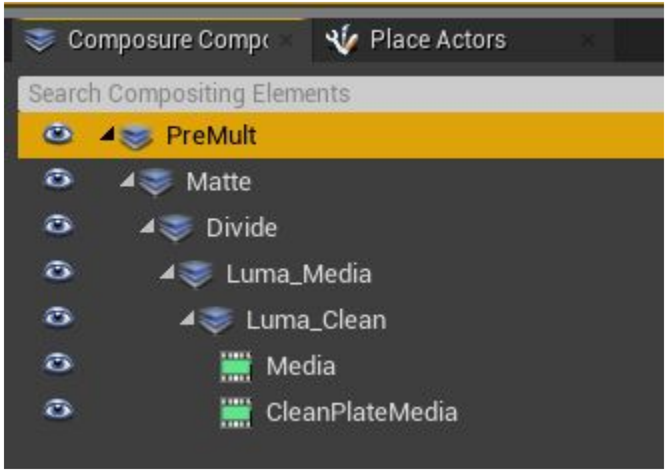


@dpredie

Compositing in Composure

CX - Clean Plate Keyer

Luma difference between provided Clean Plate & Media



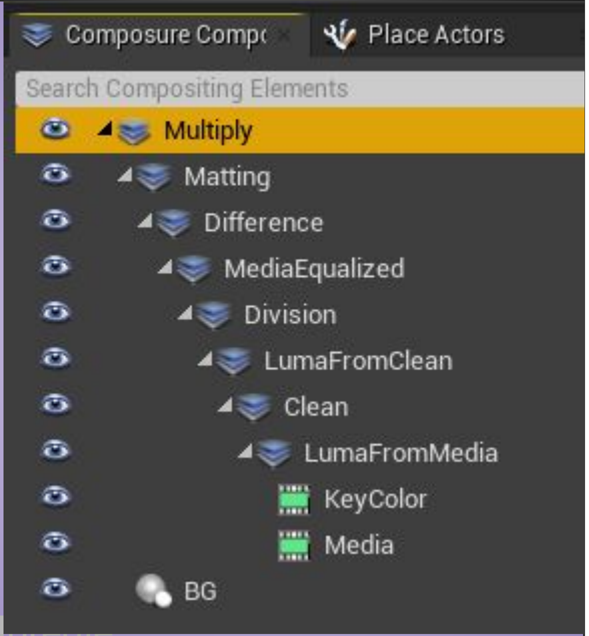
- PreMult
  - Matte
    - Divide
      - Luma_Media
        - Luma_Clean
          - Media
          - CleanPlateMedia

@dpredie

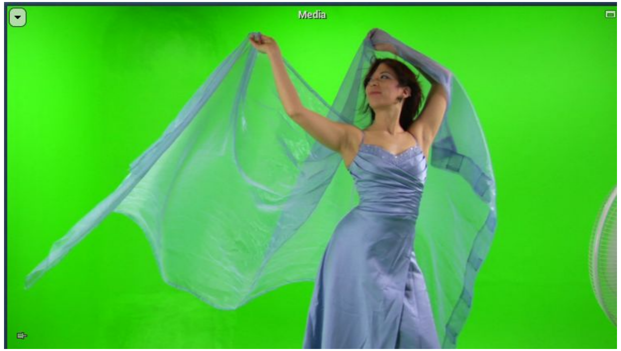
Compositing in Composure

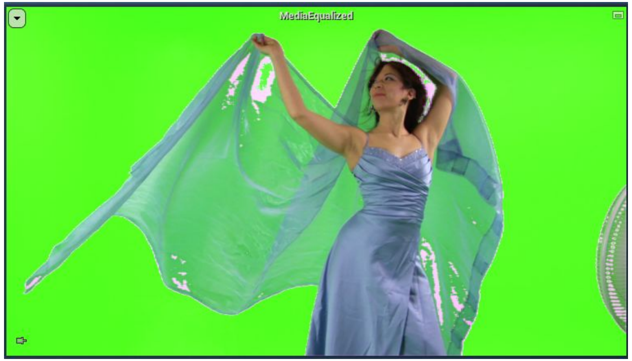
CX - Transparency Keyer

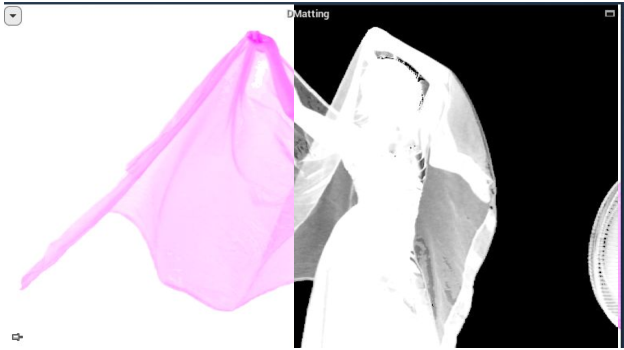
Difference Based Keyer using 2.b. Equalized Plate

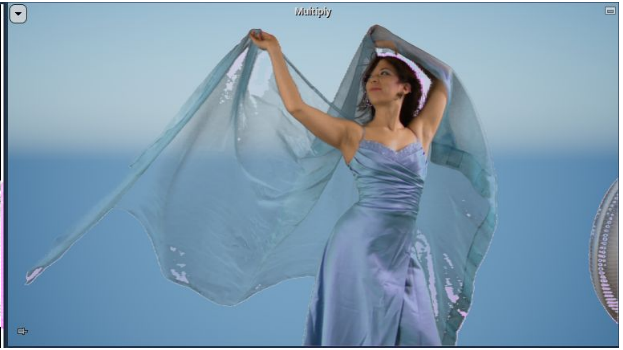


- Multiply
  - Matting
    - Difference
      - MediaEqualized
        - Division
          - LumaFromClean
            - Clean
              - LumaFromMedia
                - KeyColor
                - Media
- BG





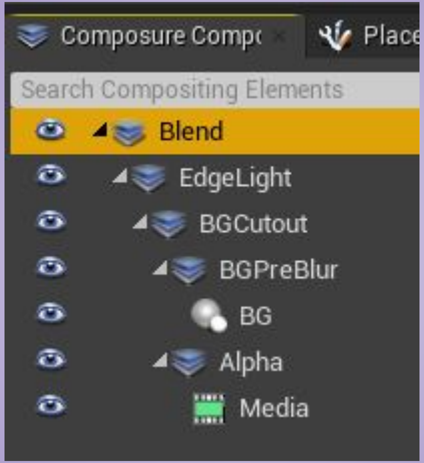




Compositing in Composure

CX - Lightwrap

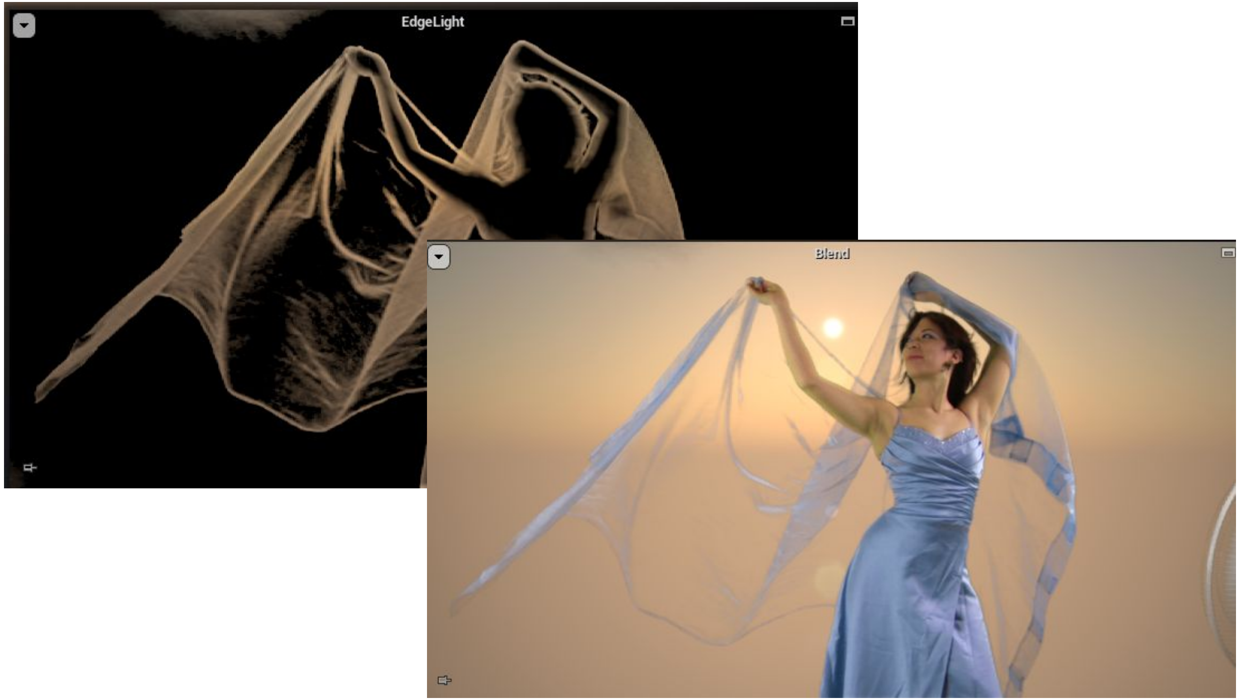
Modular implementation using CX - using epic’s M_SinglePasscolorDiffKeyer, but any previous keyer that generate alpha can be used



Composure Compo Place

Search Compositing Elements

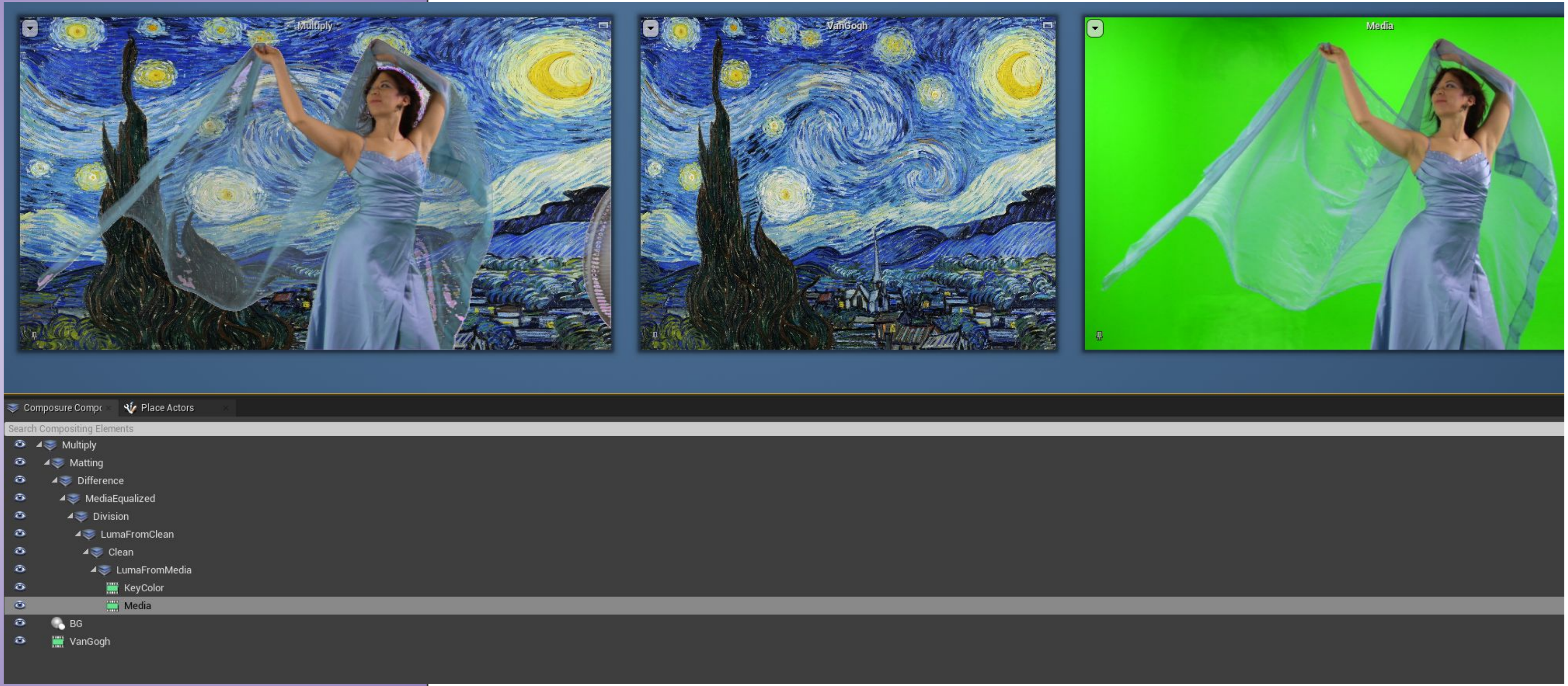
- Blend
  - EdgeLight
    - BGCutout
      - BGPreBlur
        - BG
      - Alpha
        - Media



@dpredie

What's Possible

UE can become a real-time software keyer - not necessarily using UE Environments. Any Camera/Video/Image feed can be fed to Composure layers



Search Compositing Elements

```
Multiply
  Matting
    Difference
      MediaEqualized
        Division
          LumaFromClean
            Clean
              LumaFromMedia
                KeyColor
                Media
BG
VanGogh
```

Credits

- Johan Folke for general Lightwrap algorithm
- MihranStepanyan for Plate Equalizer algorithm
  - https://www.youtube.com/c/MihranStepanyan/videos
- Clement Gharini for Straight Dry Key algorithm
  - https://www.youtube.com/watch?v=FRNFfPN8WjI