

CX

Composure X

Realtime Compositing Workflow



@dpredie

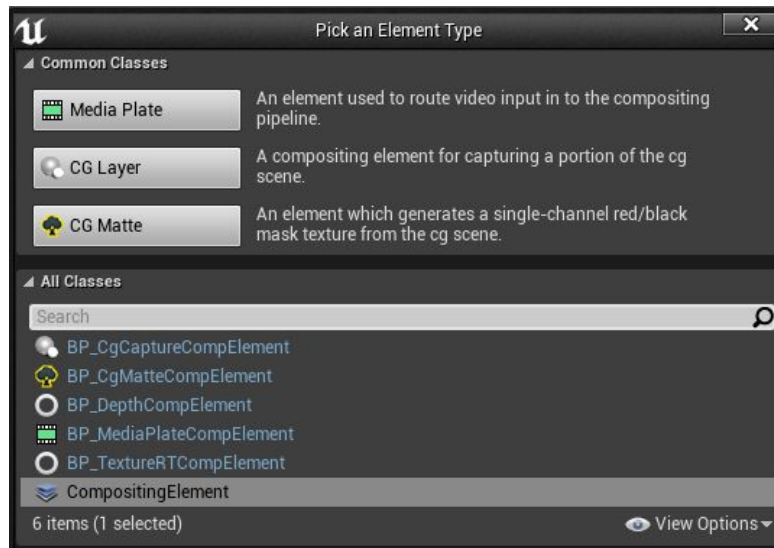
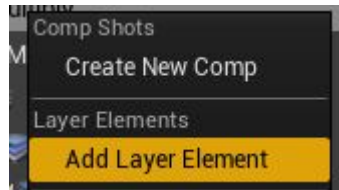


Building real-time compositing tools?



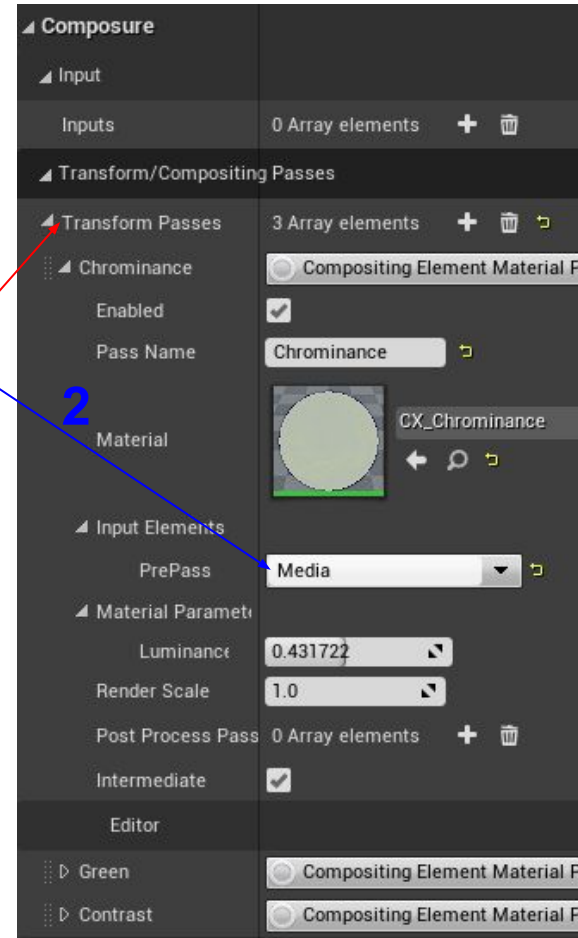
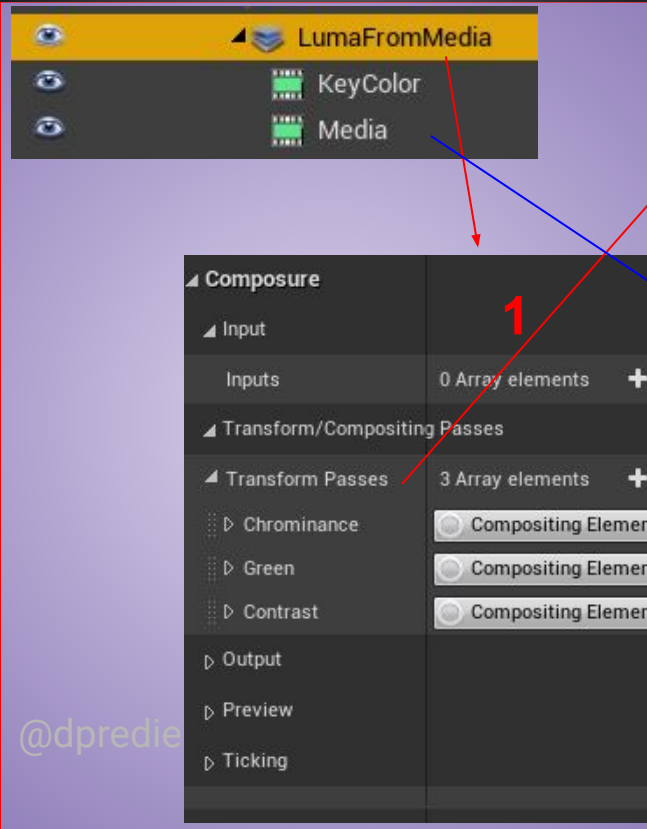
Layers or Passes?

Layers:



Layers or Passes?

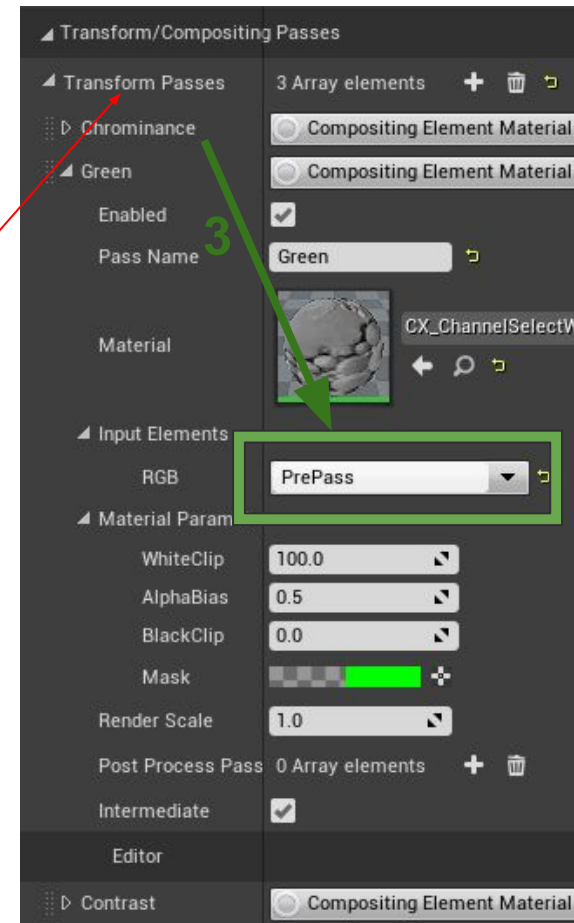
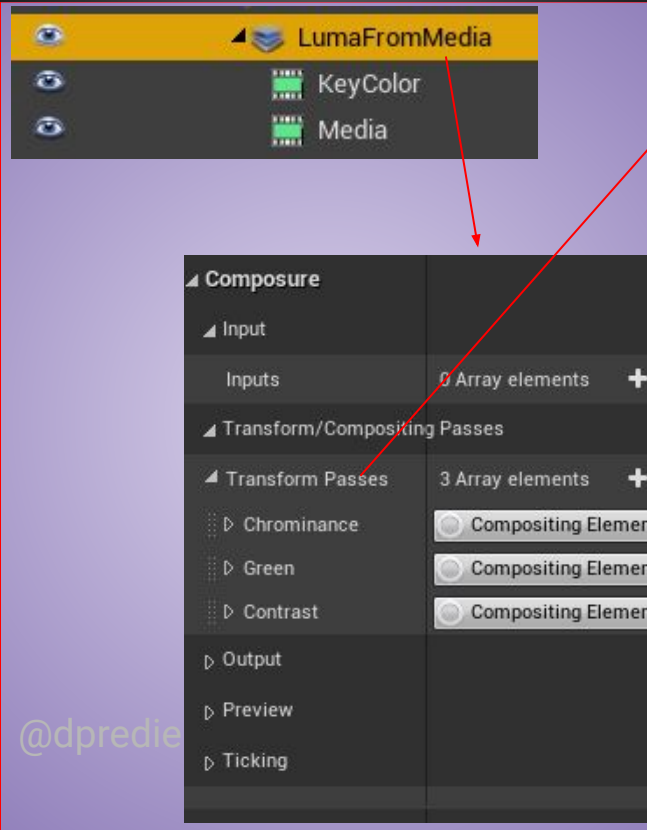
Passes:



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

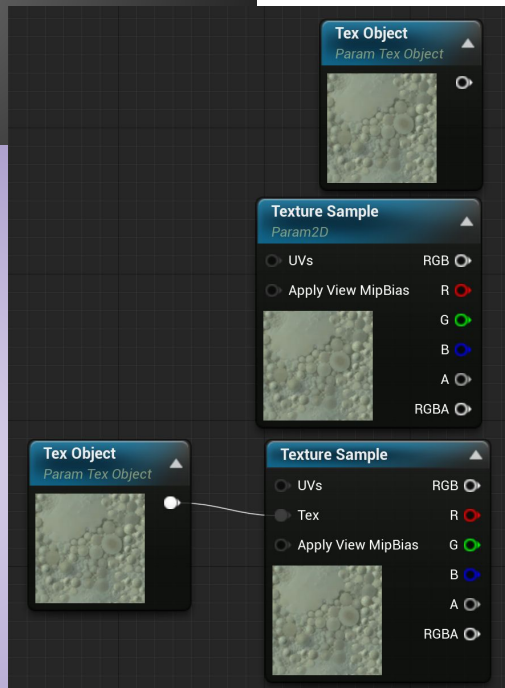
Layers or Passes?

Passes:



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:

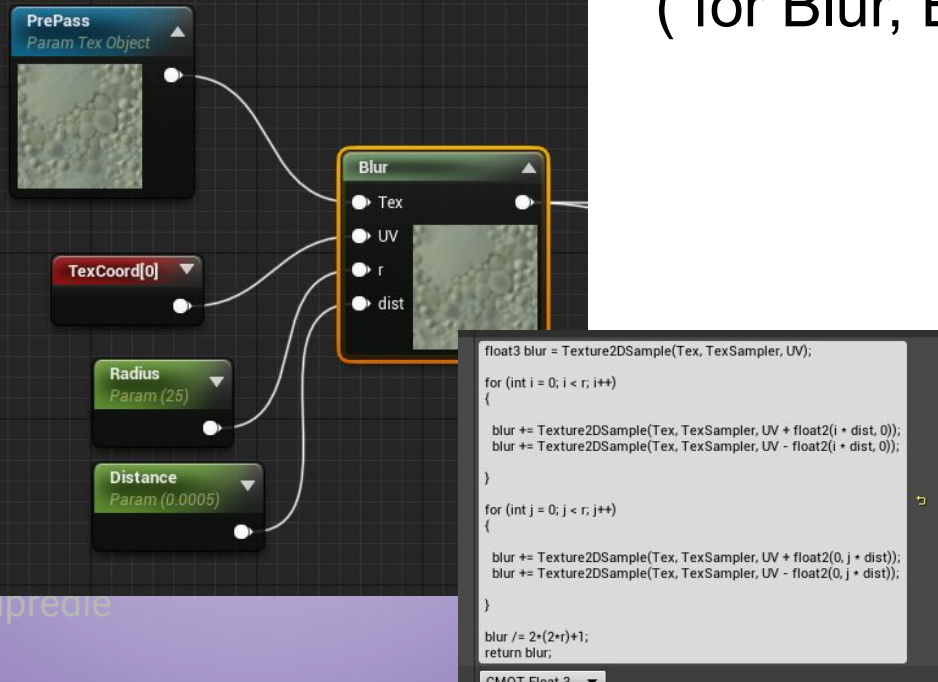
single pixel's color (cannot access neighboring pixel for manipulation)

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).



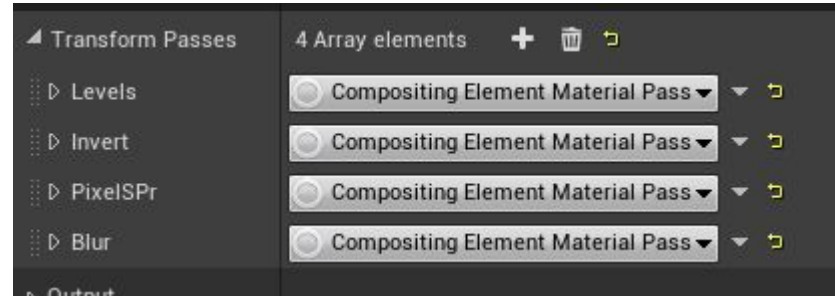
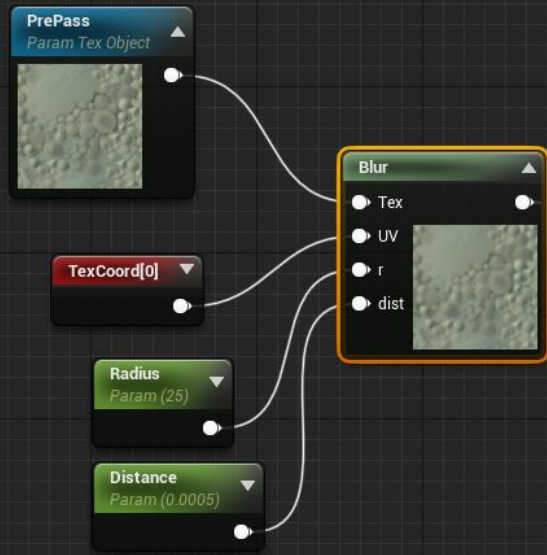
result of these mathematical nodes are Texture Sample, so you cannot perform multiple Tex Object manipulation in 1 Material BP

Why Build Multi Layers?

Tex Object

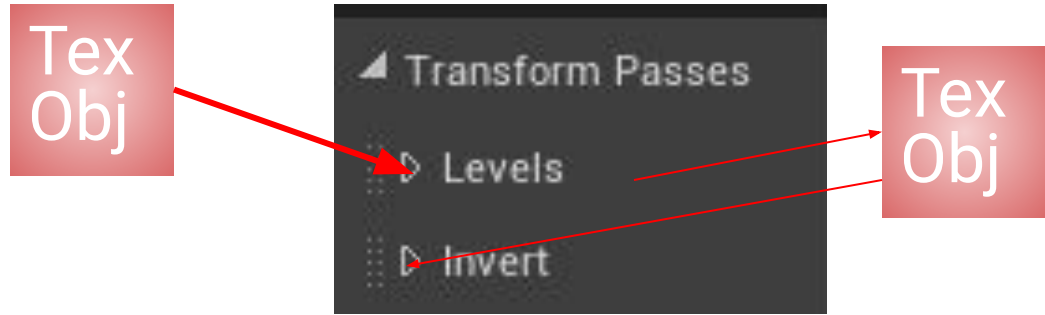
For compositing workflow you need to do multiple manipulation passes (preblur → Erode → blur, etc), which is not possible within single BP

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



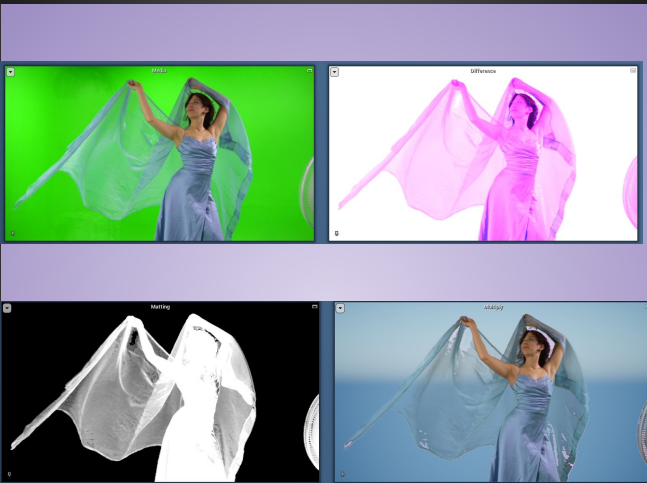
Why Build Multi Layers?

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

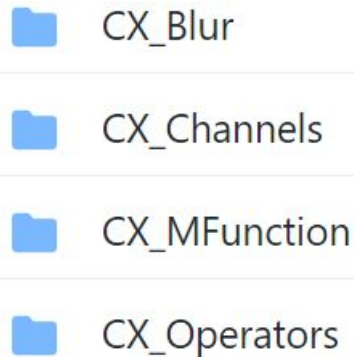
Once finalized, *you can build the compositor in fewer materials to optimize performance* (fewer renders to Tex Obj)

Compositing in Composure

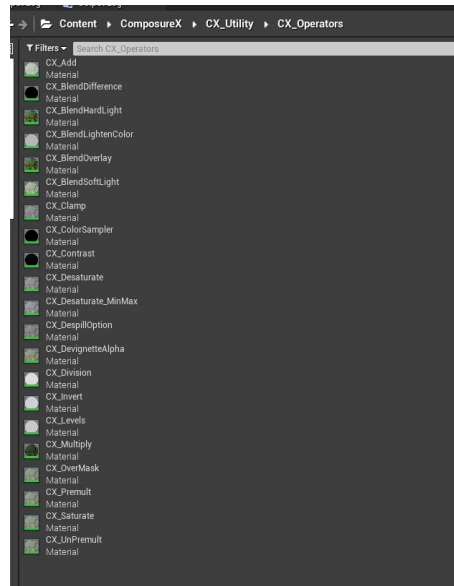
CX - Workflow

<https://github.com/dpredie/ComposureX>

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



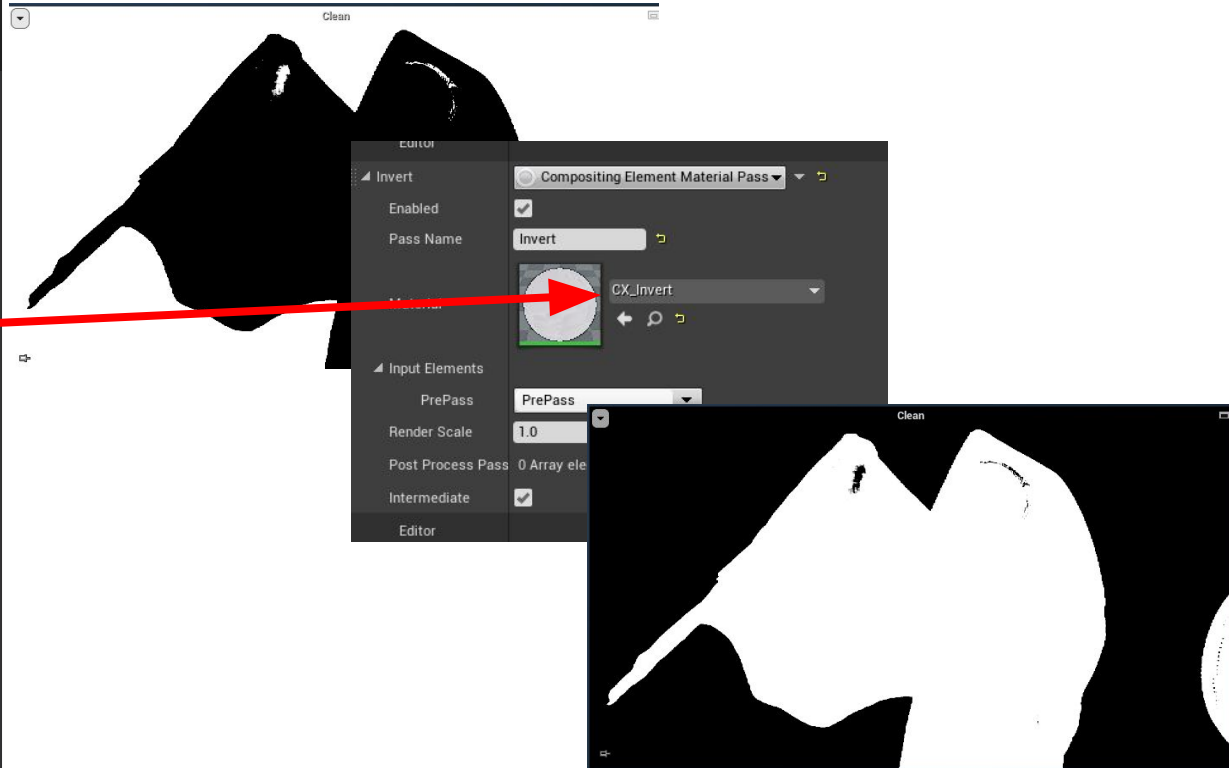
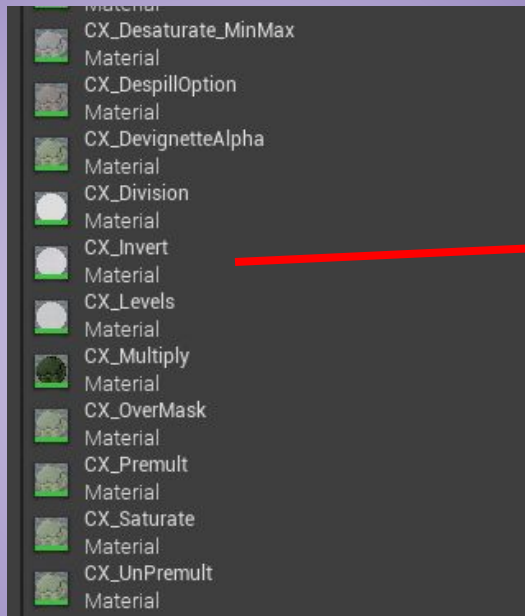
- ❏ CX_Add.uasset
- ❏ CX_BlendDifference.uasset
- ❏ CX_BlendHardLight.uasset
- ❏ CX_BlendLightenColor.uasset
- ❏ CX_BlendOverlay.uasset
- ❏ CX_BlendSoftLight.uasset
- ❏ CX_Clamp.uasset
- ❏ CX_ColorSampler.uasset
- ❏ CX_Contrast.uasset
- ❏ CX_Desaturate.uasset
- ❏ CX_Desaturate_MinMax.uasset
- ❏ CX_DespillOption.uasset
- ❏ 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



Compositing in Composure

CX - Utilities

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



Compositing in Composure

CX - Samples

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

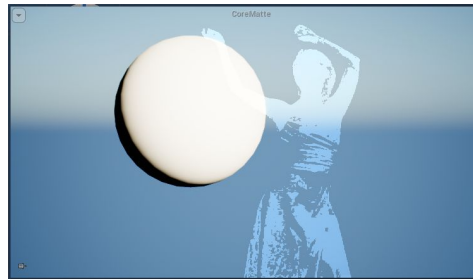
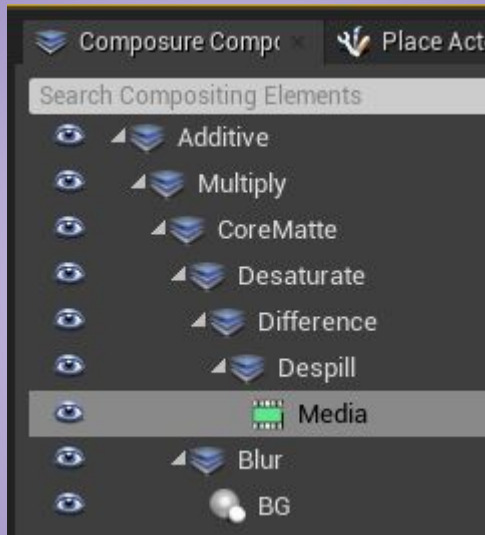
Filters Search CX_Samples

-  CX_AdditiveKeyer
Level
-  CX_CleanPlateEqualizer
Level
-  CX_CleanPlateKeyer
Level
-  CX_Lightwrap
Level
-  CX_PlateEqualizer
Level
-  CX_TransparencyKeyer
Level

Compositing in Composure

CX - Additive Keyer

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



Compositing in Composure

CX - Clean Plate Equalizer

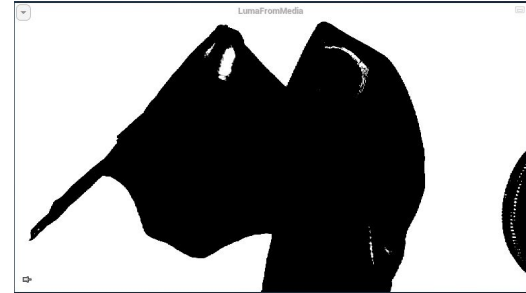
1. Provide Clean Plate Image & Equalize uneven background (Luma Based)



Compositing in Composure

CX - Plate Equalizer

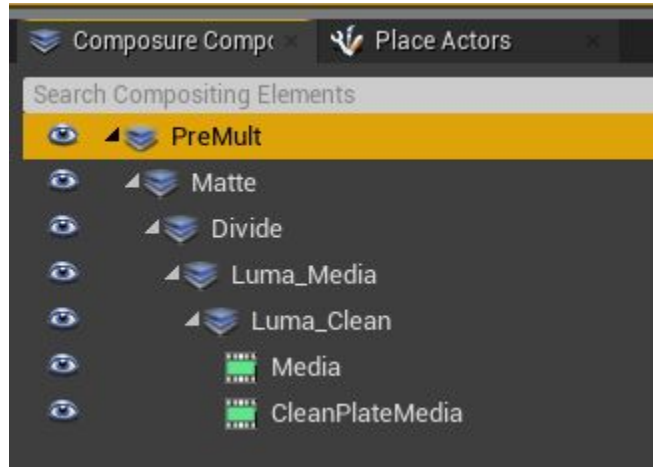
2. Estimate Clean Plate Image & Equalize uneven background (Luma Based)



Compositing in Composure

CX - Clean Plate Keyer

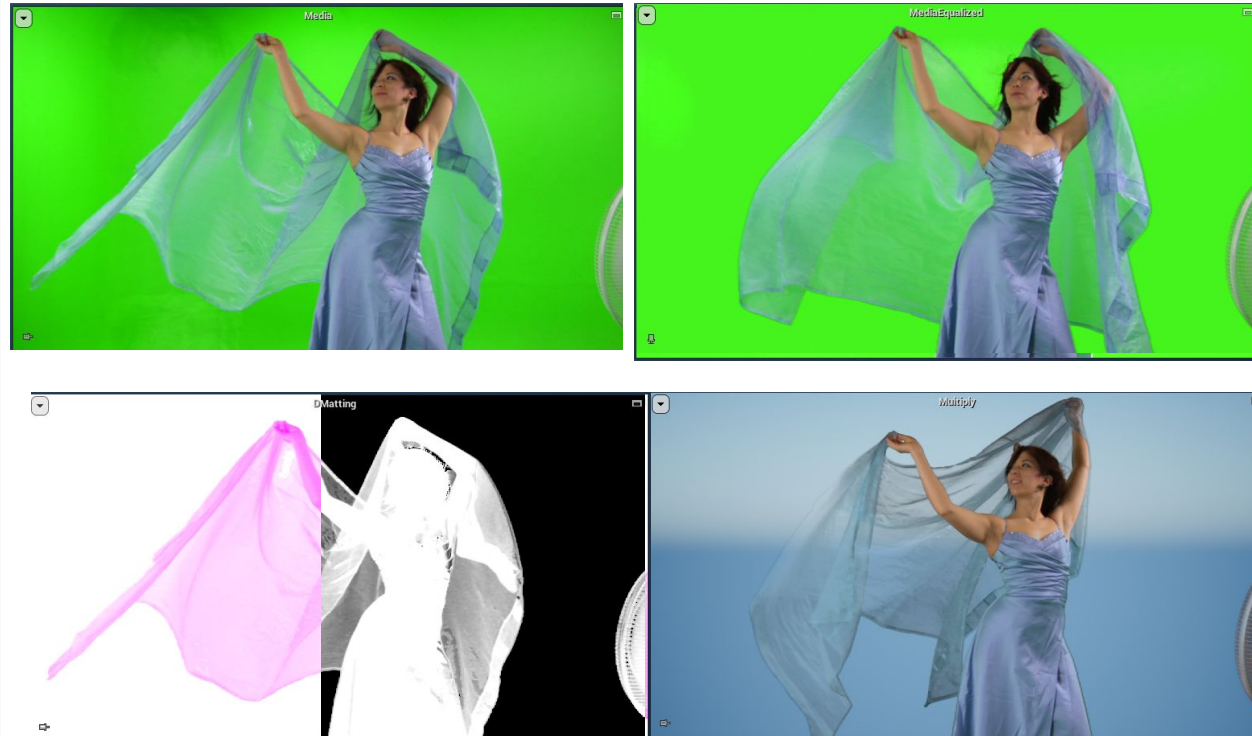
Luma difference between provided Clean Plate & Media



Compositing in Composure

CX - Transparency Keyer

Difference Based Keyer using 2.b. Equalized Plate



Composure Comp Place Actors

Search Compositing Elements

Multiply

Matting

Difference

MediaEqualized

Division

LumaFromClean

Clean

LumaFromMedia

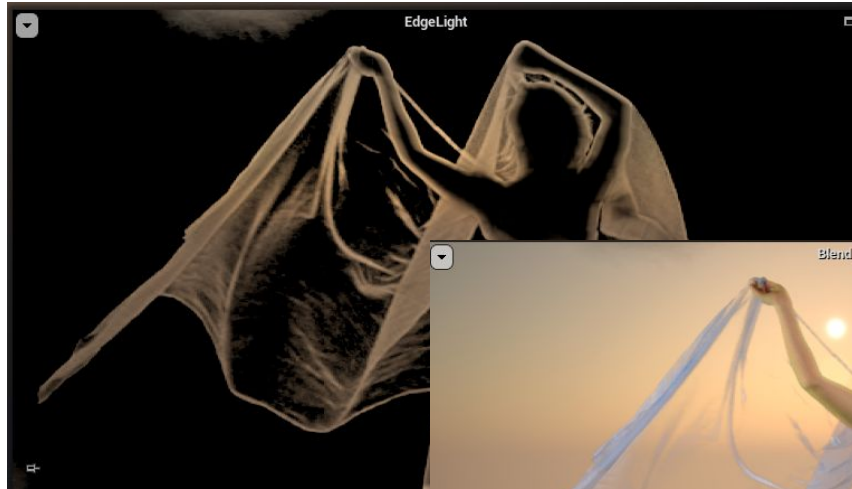
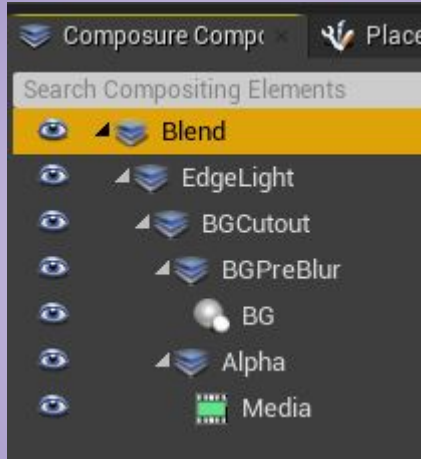
KeyColor

Media

BG

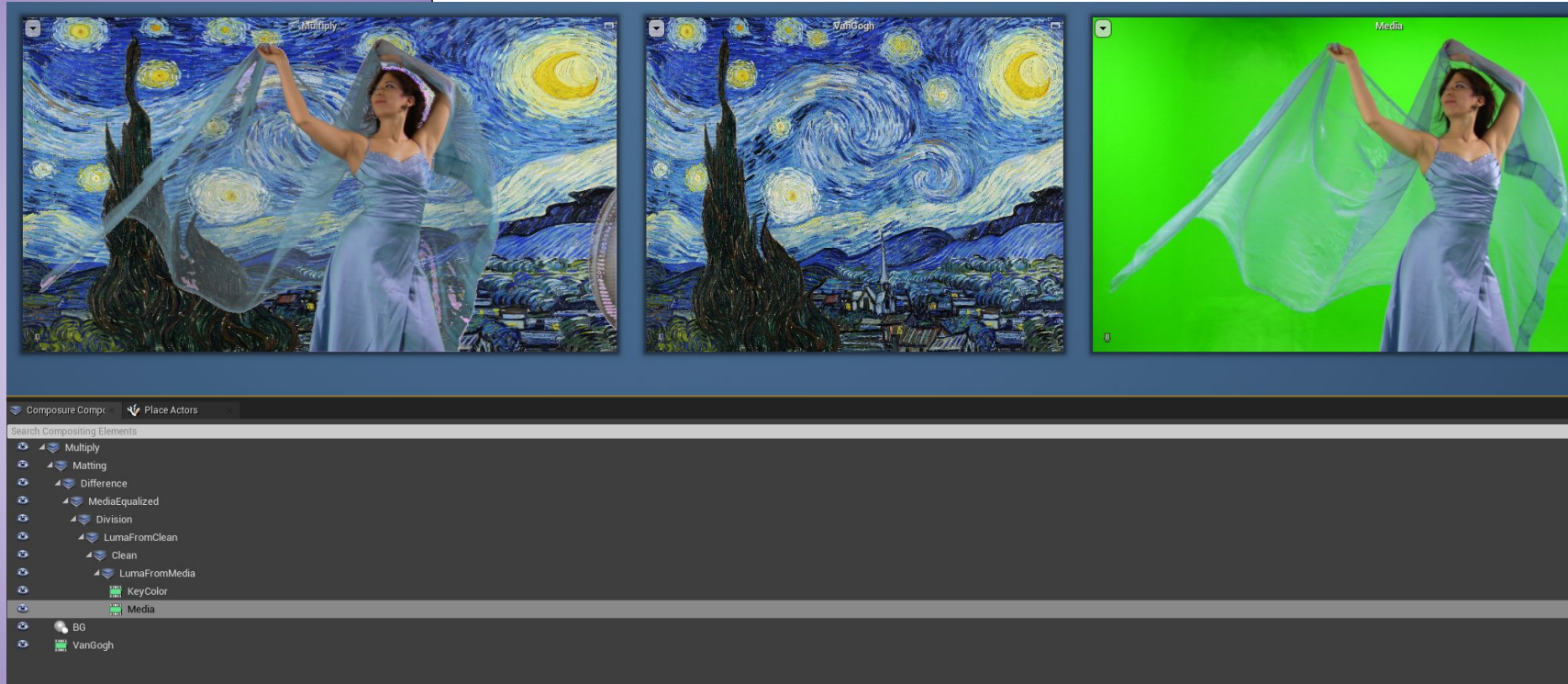
Compositing in Composure CX - Lightwrap

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



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



Credits

- 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>