Natron 2 : Integration with 3rd party tools

informatics mathematics

Inria

FOSS compositing: Natron

- Developed in a research institute: INRIA
- 1 developer full-time, 1 computer vision scientist, 1 internship developer, 1 community manager, 2 VFX artists/teachers
- Contract ends with INRIA in 2016
- Plan is to make a consortium with industrial members supporting the project
- Natron is already used among some major world-wide studios
- Focus on experimented users

Over the last year

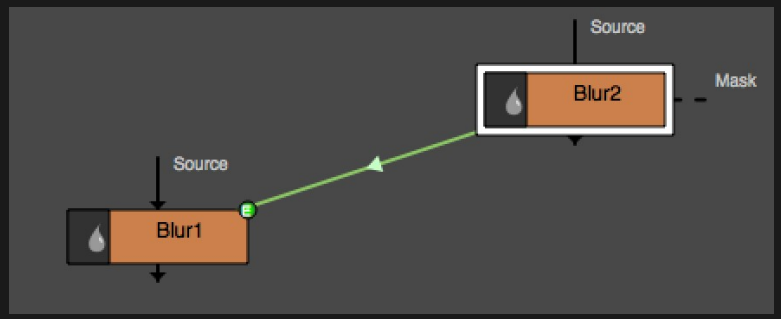
1. Natron 1 was presented at SIGGRAPH 2014
2. A lot of requests for interaction with 3D softwares
3. In January 2015 Natron received the award for Best Innovation at the Paris Images Digital Summit

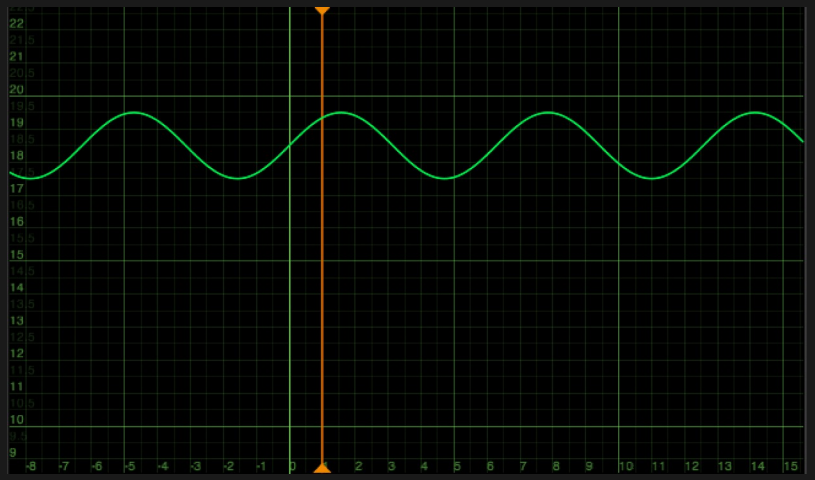
Market

- Competitors are The Foundry Nuke (company to be sold in July this year), Adobe After Effects, Fusion 7 by BMD and BColor by BUF software.
- Natron relies on OpenFX which is an industrial standard for visual effects plug-ins
- Natron aims to become a stable standard for compositing
- Natron is now taught in a few schools instead of Nuke

Parameters expressions

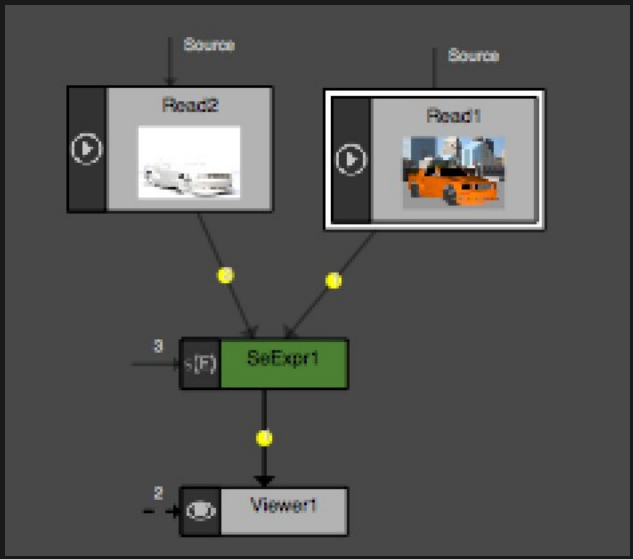
thisGroup.Blur2.size.get()[dimension] + sin(frame)





curve(((frame - 0) % (7 - 0 + 1)) + 0)

Se·Expr()



(Over operator): Cs + Cs2 * (1 - As)

Application to interactive lighting

Lpics: a Hybrid Hardware-Accelerated Relighting Engine for Computer Cinematography

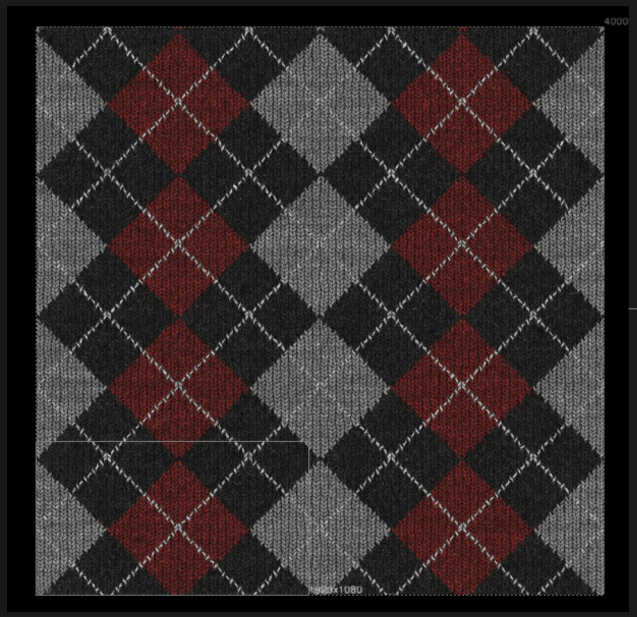
Fabio Pellacini* Kiril Vidimče† Aaron Lefohn† Alex Mohr† Mark Leone† John Warren†
Pixar Animation Studios

Lpics render ≈ 0.1s

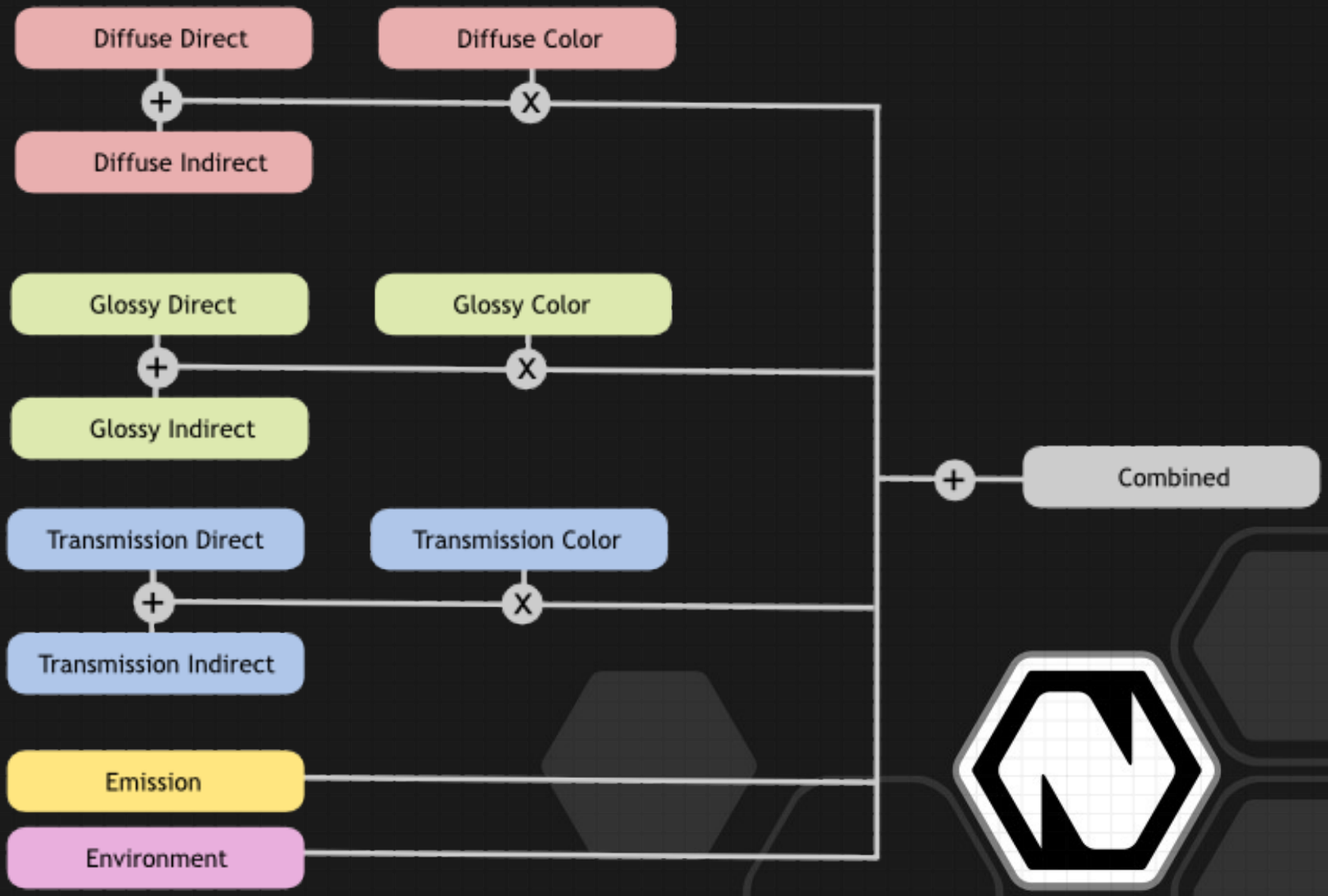
Final render ≈ 2000s

Figure 1: Images rendered by Lpics relighting engine versus software renderer.
Times reported are the time a lighting artist must wait for feedback after moving one light.

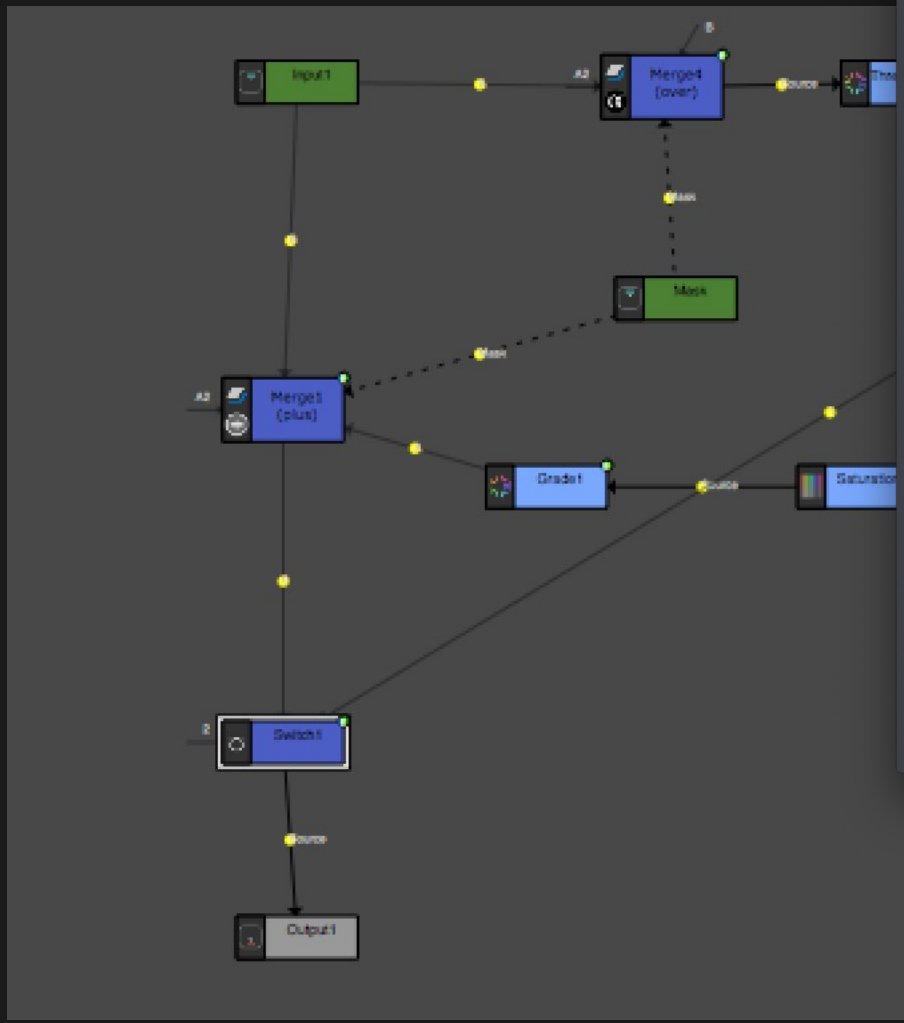
Texture mapping



Multi-layered



PyPlugs



Glow.py

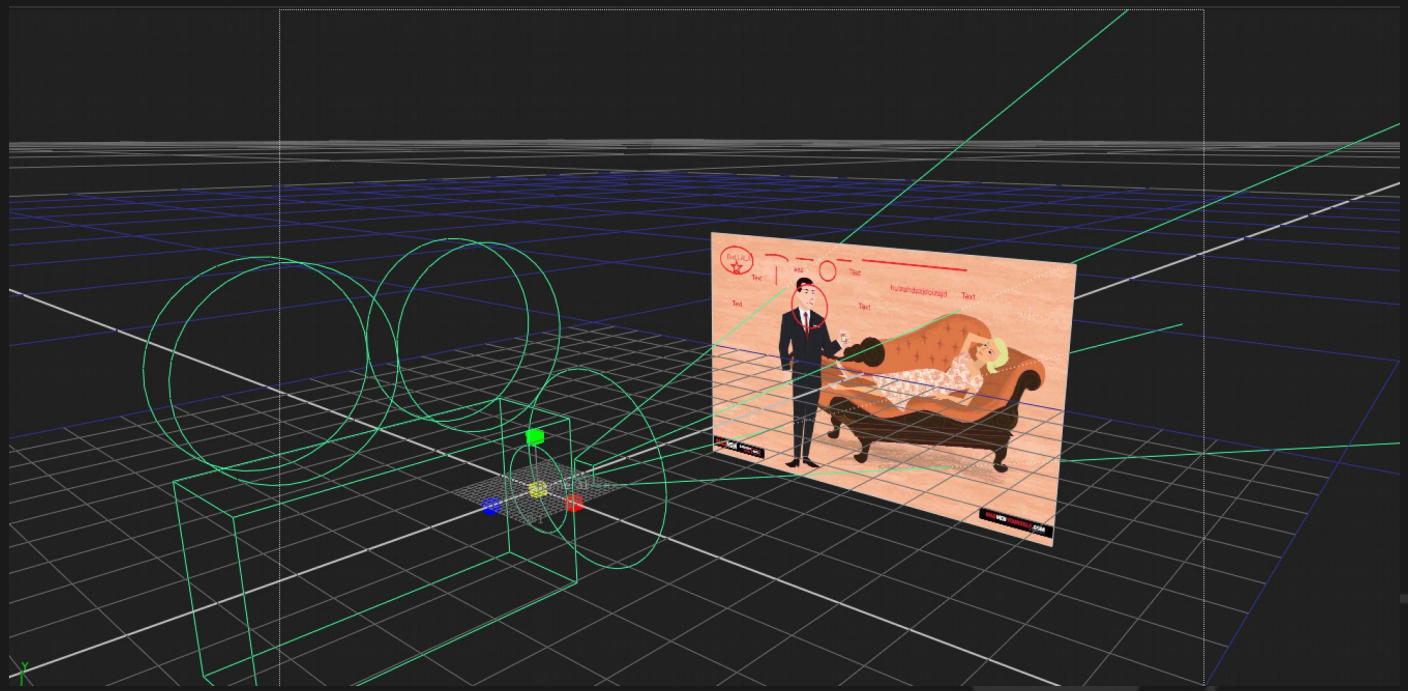
Glow.py 〉 No Selection

```python
# -*- coding: utf-8 -*-
#This file was automatically generated by Natron.
#Note that Viewers are never exported

import NatronEngine

def getPluginID():
    return "glow"

def getLabel():
    return "Glow"

def getVersion():
    return 1

def getGrouping():
    return "Filter"

def createInstance(app,group):

    #Create all nodes in the group
    lastNode = app.createNode("fr.inria.built-in.Output", 1, group)
    lastNode.setScriptName("Output1")
    lastNode.setLabel("Output1")
    lastNode.setPosition(-14.7337, 742.058)
    lastNode.setSize(104, 37)
    lastNode.setColor(0.6, 0.6, 0.6)
    groupOutput1 = lastNode

    param = lastNode.getParam("Output_layer_name")
    if param is not None:
        param.setVisible(False)
        del param

    param = lastNode.getParam("highDefUpstream")
    if param is not None:
        param.setVisible(False)
        del param

    del lastNode



    lastNode = app.createNode("fr.inria.built-in.Input", 1, group)
    lastNode.setScriptName("Input1")
    lastNode.setLabel("Input1")
    lastNode.setPosition(-14.7337, 7.54915)
    lastNode.setSize(104, 37)
    lastNode.setColor(0.300008, 0.500008, 0.2)
    groupInput1 = lastNode

    param = lastNode.getParam("Output_layer_name")
    if param is not None:
        param.setVisible(False)
        del param
```

In the future...

3D workspace:

- 3D cards for projected textures
- Camera position from directly from Blender
- Lights positions directly in Natron and talk-back via socket to Blender



In the future...

- Gmic integration: hundreds of effects that are also used in Gimp
- Bolster FOSS integration: Export to Shotcut/Open-shot
- Import from After Effects

Thanks!

- Any Questions?