REDUCE COGNITIVE OVERLOAD IN GRAPHICS DESIGN SOFTWARE

SUMMARY

This article investigates reducing Cognitive Overload in graphics design software by enabling intuitive user interfaces with a simplified digital compositing node graph, an **Open Pipeline**.

An Open Pipeline is to user interface of graphics design software like what Markdown is to HTML.

'In nova fert animus mutatas dicere formas corpora; di, coeptis (nam vos mutastis et illas) adspirate meis primaque ab origine mundi ad mea perpetuum deducite tempora carmen. ' – OVID

OUTLINE

COGNITIVE OVERLOAD IN GRAPHICS DESIGN SOFTWARE

Graphics design software such as photo editors, animators, or video editors deploys a combination of commonly used user interfaces such as Master-Detail panels, Dialog boxes, Properties Tree, Timeline, Applications Menu, Context Menu, and What-You-See-Is-What-You-Get (WYSIWYG) Canvas, in a **NON-STREAMLINED** manner. This non-streamlined user interface with too many, hard to find, or ambiguous options, can cause overstimulation and overwhelms the artist's cognitive working memory.

Imagine an artist performing the following photo edits:

1. Apply a Disc Blur to an image.
2. Apply a Sepia Tone to Disc Blur.
3. Blend the Sepia Tone with the original.

The artist's intent usually consists of a sequence of actions. Due to the way current graphics software is designed, it may involve the artist carrying out the task by selecting options in the Application menu, changing properties in a Side panel, bringing up an Object dialog box, and/or creating Layers to achieve the desired effect. This non-streamlined approach of user interface **disrupts the flow of thoughts of the artist**, and when exacerbated with an increasing number of steps results in **Cognitive Overload** and **loss of creative productivity**.

A PROPOSAL

It is proposed to streamline the user interface into a linear sequence of actions that can be compounded or stacked like a digital compositing node graph but with simplification to express how to achieve the desired outcome succinctly.

AN EXAMPLE

Imagine a graphics software having the following graphics components, hereupon refer to as a node, with each of them achieving an atomic graphics operation.

- Read Image
- Disc Blur
- Sepia Tone
- Blend

Each of these nodes is designed with a corresponding screen for changing properties.

- Read Image -> Properties Screen
- Disc Blur -> Properties Screen
- Sepia Filter -> Properties Screen
- Blend -> Properties Screen

STREAMLINE PROCESSES

An artist performing the above photo edits can follow his or her flow of thoughts by adding atomic graphics operation, abstracted into a node, into a list. The mapping of the artist's intent and the addition of nodes to a list are shown side by side below.

```
                                                    <-> Read Image
1. Apply a Disc Blur to an image.                   <-> Disc Blur
2. Apply a Sepia Tone to output of Disc Blur.       <-> Sepia Filter
3. Blend output of Sepia Tone with original image.  <-> Blend
```

The user interface of a photo editor can be generalized and simplified to be as follow:

```
List of actions
[
 .
 Empty
 .
]

+Add Node
```

And when an artist adds nodes, it becomes:

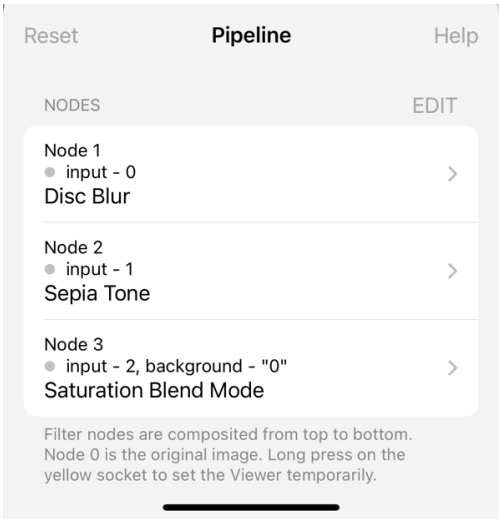
```
 List of actions
 [
  1. Read Image     |
  2. Disc Blur      |   Add nodes to replicate the thought process
  3. Sepia Tone     |
  4. Blend (1,3)    V
  ]

 +Add Node
```

The above can be simplified further to:

Pipeline
1. Read Image
2. Disc Blur
3. Sepia Tone
4. Blend (1,3)

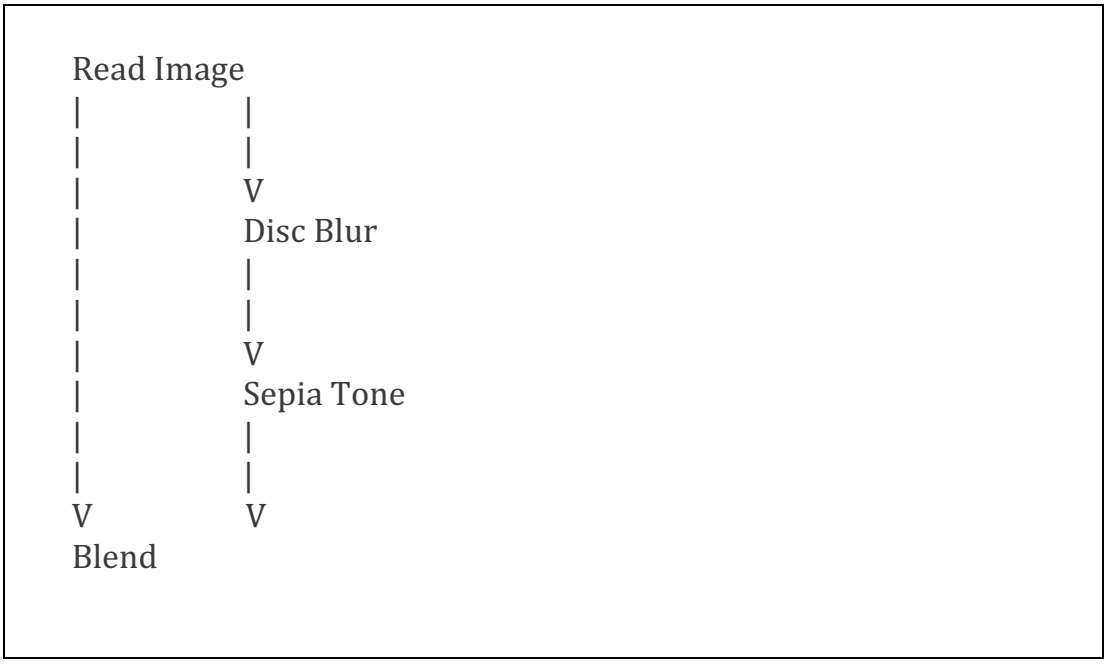
The focus of the artist is on what to do next from start to end by adding nodes to a graphics Pipeline. The user interface of the Pipeline makes it easy for the artist to keep his or her focus on what the next step should be. The Add Node capability could aid by displaying a list of all available and relevant next steps.



Sample User Interface

DIGITAL COMPOSITING NODE GRAPH

To someone familiar with Digital Compositing, they will immediately recognize that the list of steps, the Pipeline, above is "almost" identical to a digital compositing node graph. A digital compositing node graph is shown below:



WHAT REALLY CHANGED? A SIMPLIFICATION

A Pipeline is indeed like a digital compositing node graph with a subtle yet significant change. Instead of displaying a full node graph graphically, with two lines from the Read Image node and Sepia Tone node connecting to the Blend node. The Pipeline simplifies the node graph by just showing a list of steps and a spreadsheet-like function for the final Blend step.

4. Blend (2,3)

The above approach can be further expanded to support 2 or more nodes.

4. Blend with Mask (1,2,3)

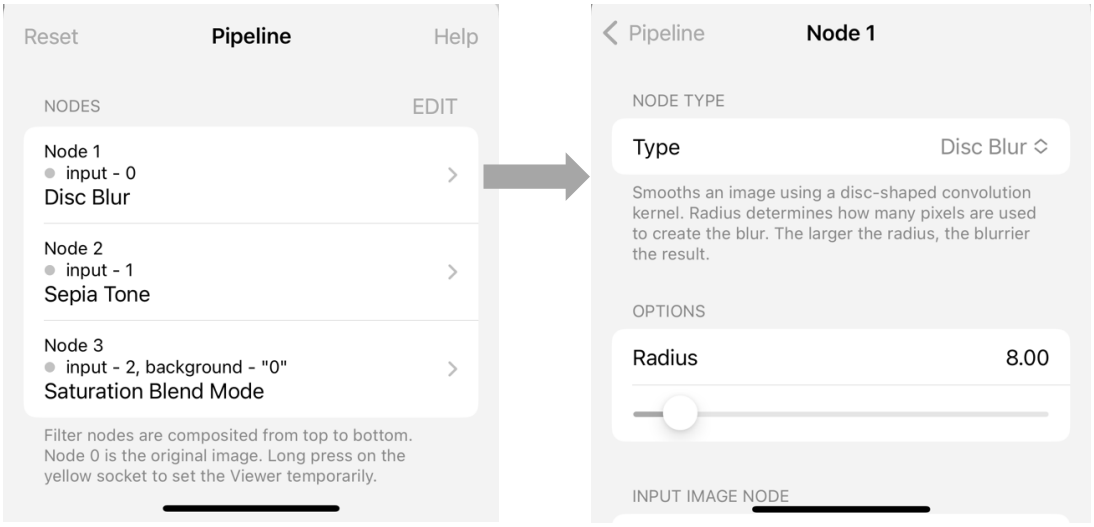
Node 1 and node 2 are the two nodes to be blended while node 3 is the mask node. We will leave the details of naming the nodes for another day.

This simplification of a digital compositing node graph to a Pipeline enables the intent of the artist to be conveyed as a succinct list of steps, following the artist's flow of thoughts from start to end, without a graphical map. The Pipeline also guides the artist into thinking of what to do next in a streamlined manner. Without using a traditional node graph, more information can be conveyed on a smaller amount of space in a linear manner, providing a larger contextual overview to the creative mind.

SIMPLE USER INTERFACE AND PROPERTIES SCREEN

A Pipeline, in software, maps directly to a list and is easy to implement. The list removes the need for a complicated graphical map that may require **panning** (scrolling and shifting in different directions) to identify the right node to work on. The properties screen of a node can also be brought up intuitively and naturally by drilling into the node.

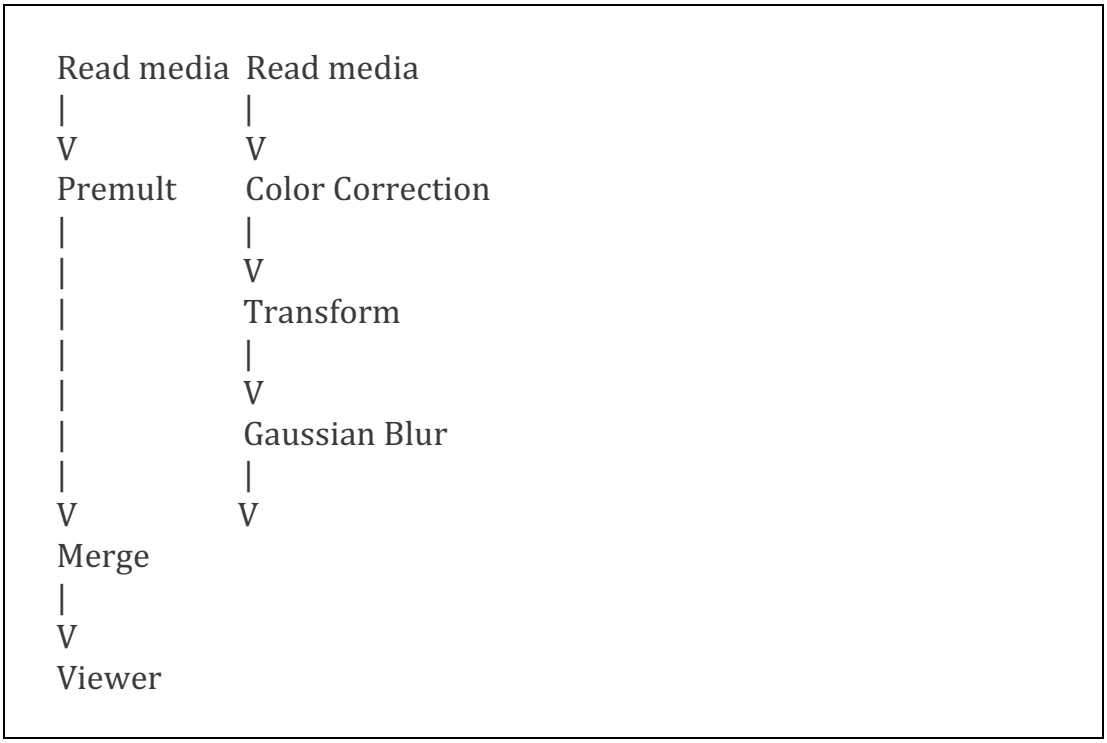
Pipeline
1. Read Image
2. Disc Blur ----> Drill into the Properties screen
3. Sepia Tone
4. Blend (2,3)



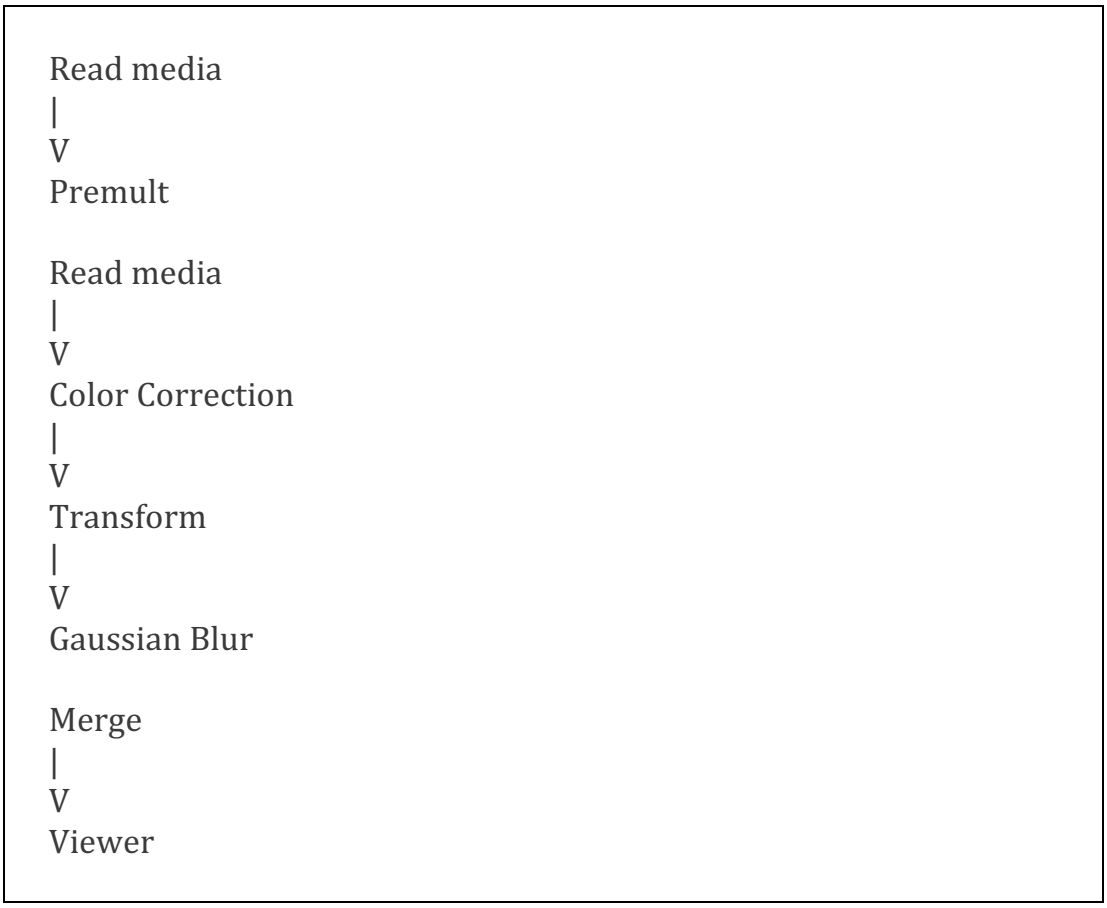
Another artist, loading a saved Pipeline, will be able to quickly understand the flow and intent of the original artist by following the steps from the start to the end.

NO LOSS OF INFORMATION

The important thing to note is the above simplification does not lose any information. The simplified node graph above, the Pipeline, in a mathematical sense, can be fully represented as a node graph (Directed Acyclic Graph) in any manner or form. For example, if we have the following node graph for video processing.



It can be flattened, streamlined, and represented as follows:



Note the break from the first series after the 'Premult' node and the second break after the 'Gaussian Blur' node.

```
Pipeline

1. Read media          >
2. Premult             >

3. Read media          >
4. Color Correction    >
5. Transform           >
6. Blur                >

7. Merge (2,6)         >
8. Viewer              >
```

The space between node 2 and node 3 represents the break, while the '>' is optional and is used to indicate the availability of a properties screen.

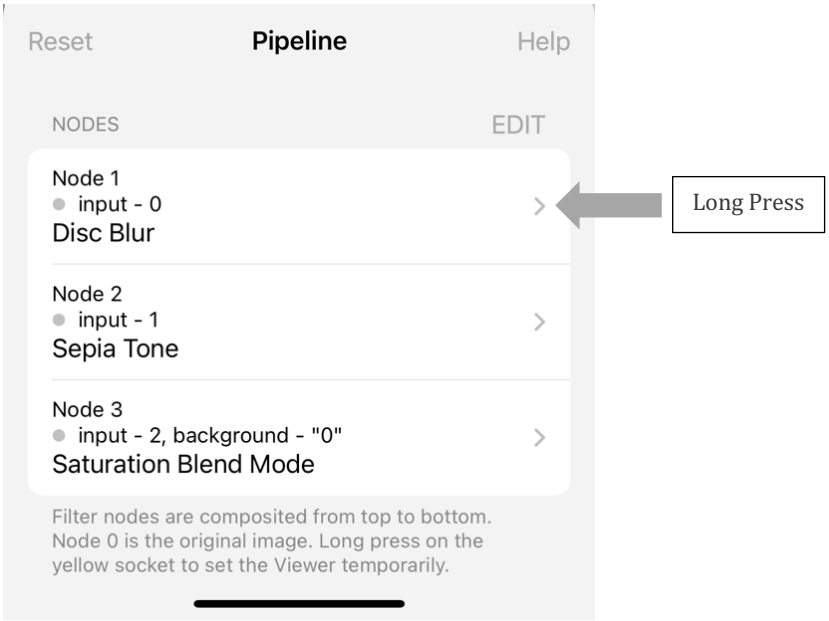
DIGITAL COMPOSITING PARADIGM ON A PIPELINE

The paradigm of node-based digital compositing where we can lay out steps in a progression and make modifications to an earlier step to change the outcome is completely kept. Each node can be easily reused by referencing it with a node number as in Step 7, and if required, used further in a later step, saving time and effort.

In the same way that a digital compositing node graph is broken into a series of nodes in a graph, a Pipeline breaks it into a series of actions or steps. A Pipeline, from another angle, is thus like a single-column spreadsheet, a metaphor that many computer users are quite familiar with.

- ***Viewing the output of each step: Viewer Cycling***

  The arrangement in a series of steps enables the output of each progression step to be viewed easily, for example, by long pressing on a node on the Pipeline. This enables an artist to step through the progression to reinforce his or her cognitive working memory of what has been performed.



- ***Connecting the steps: Auto-Chaining***

  Each node on the Pipeline that requires an input can be assumed to be taking the output of the previous node automatically. For example, the 'Premult' node in Step 2 is assumed to automatically take the output from 'Read Media' in Step 1.

  Pipeline

  1. Read media >
  2. Premult >

- ***Connecting the steps like a node graph: Reverse Compositing***

  The auto-chaining or automatic preceding node referencing behavior can also be altered by enabling the artist to make changes in the properties of a node. For example, in Step 7, the artist can choose from its properties screen any of the previous nodes, such as nodes 2 and 6, for blending.

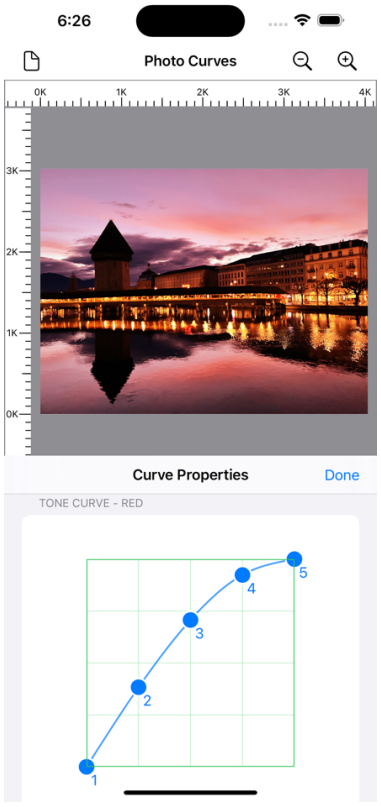
  7. Merge (2,6) >

MODULAR EXTENSIBILITY

A Pipeline, being almost identical to a digital compositing node graph, enables it to achieve the same modular extensibility in software development. Software developers can think of developing graphical effects, filters, animations, and 3D operations modularly as a node. Each node is accompanied by a properties user interface that does not interfere with other parts of the software.

A node and its user interface can thus be developed and tested independently while knowing ultimately that it can be easily integrated into a full node graph, aka the Pipeline, to achieve any complex graphics operation.

Such a design will benefit the open-source community greatly where many different developers can build an open-source Pipeline, the Open Pipeline, together in a distributed and collaborative manner.



Curves Node in Pipeline

REDUCE INFORMATION OVERLOAD

To start using a graphics application, artists usually need to learn the way of achieving it through documentation. Often, the documentation requires an artist to follow a series of screenshots, with an explanation of where and how to click or tap on different parts of the screen using a non-streamlined user interface offered by the graphics application. The non-streamlined user interface makes the information hard for an artist to remember and may require the artist to constantly refer to the documentation. Over time, this introduces an additional cost that, when accumulated, can lead to Information Overload.

Succinct Self-Describing Pipeline

The Open Pipeline reduces Information Overload as it succinctly describes what needs to be done and allows all necessary details to be shown in context nearby. This significantly benefits documentation: tutorials, articles, or books.

```
Open Pipeline
1. Disc Blur: radius = 8
2. Sepia Tone: intensity = 1
3. Blend (1,2)
```

The details of the radius for Disc Blur are shown as part of the node in the Pipeline. Moreover, besides being succinct, it is common for the above description to **NOT REQUIRE** supporting screenshots as it is self-explanatory.

FURTHER READINGS

- ***Open ^Graphics Pipeline***

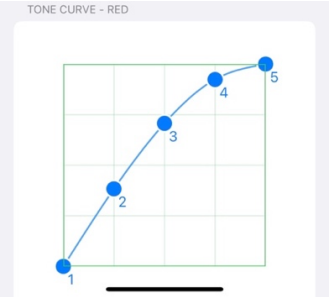
  Open ^Graphics Pipeline is fully open-source and can be found in a GitHub repo. We also have a community for showcasing each of the node projects developed independently.

- ***Digital Compositing with Open Pipeline Example***

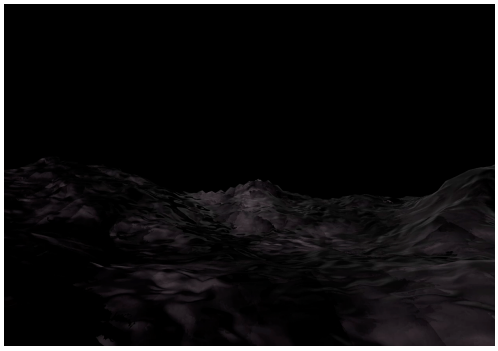
  A comprehensive example showing how Open Pipeline is used to digital compose a Photo Fire Animation Effect by mixing Shaders and Filters

- ***Modular Extensibility***

  Implementation of an open-source Photo Curve node and how its user interface can be easily integrated into Open Pipeline, achieving modular extensibility.



  Implementation of a Terrain Shader node with Fractal Brownian Motion (fBM) for integration into Open Pipeline.



  Implementation of a WYSIWYG Vector Draw node for integration into Open Pipeline.