



Ghostscript's ICC-based Color Architecture

Ralph Giles
Artifex Software Inc.
San Rafael CA



Outline

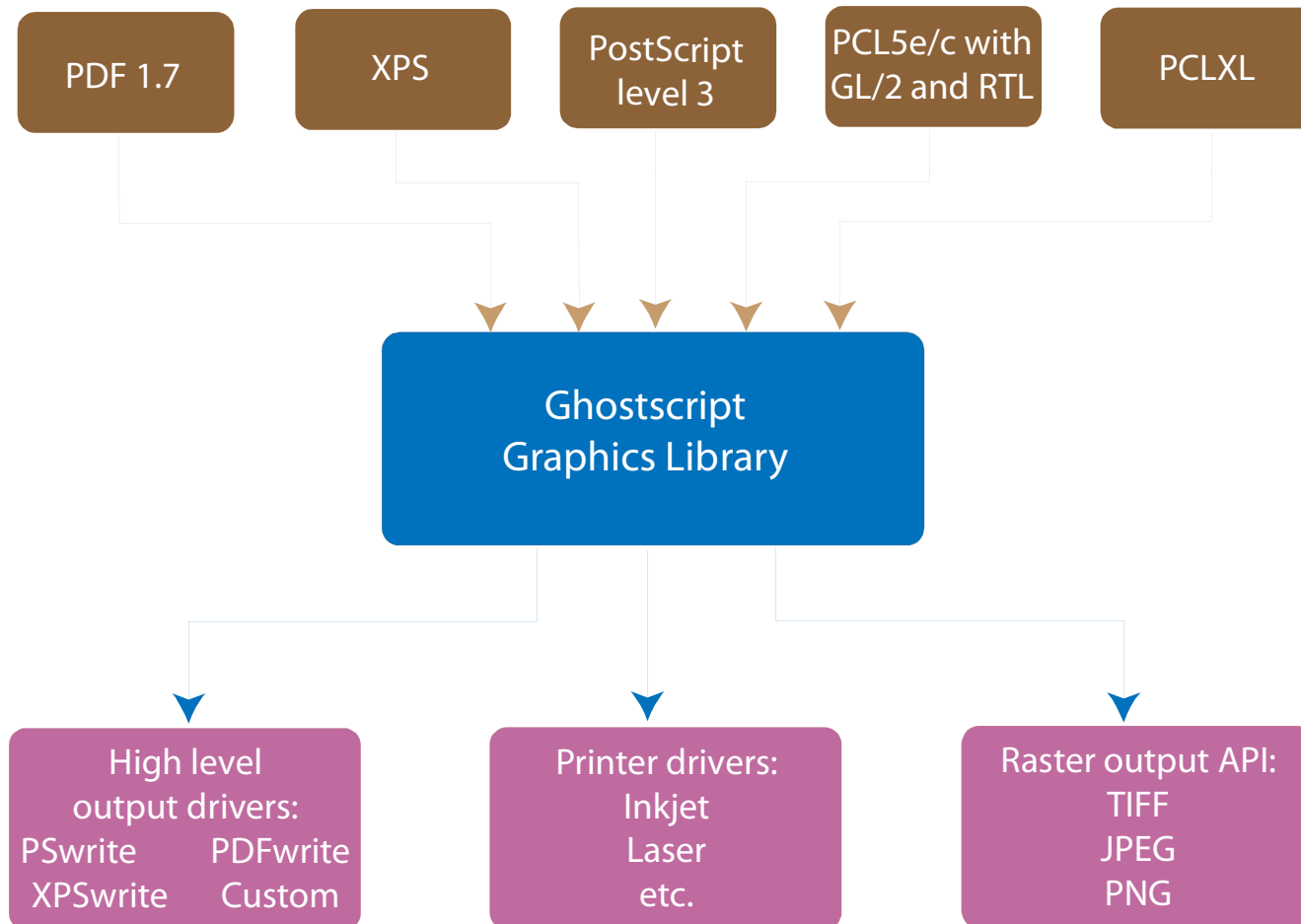
- Ghostscript Overview
- PDLs & Color Spaces
- Problems in Existing Ghostscript ICC Flow
- New Architecture



About Ghostscript

- Ghostscript is a document conversion and rendering engine.
- Essential component of the Linux printing pipeline.
- Dual GPL/Proprietary licensed. Artifex owns the copyright.
- Source and documentation available at www.ghostscript.com

Ghostscript Overview





PDLs and Color Spaces: Overview

PostScript Level 3 – DeviceGray, DeviceRGB, DeviceCMYK, Device independent color spaces 1, 3, and 4 component (CIEXYZ based), Separation, N-Device, Indexed, and Pattern.

PDF – Essentially same as PSL3, but adds ICCBased as input type and loses some PS CIE based spaces. Adds a LAB type. Only supports 1, 3 or 4 channel ICC profiles.

PCL – RGB based. Color assumed to be sRGB.

XPS, OpenXPS, SVG – All color defined by ICC profiles. XPS allows up to 8 channels.

Ghostscript supports ALL.



Ghostscript and ICC Color Spaces

Today, color management in prepress environments is handled completely through the use of ICC profiles.

Ghostscript supports all color spaces defined by PSL3 and PDF1.7 with the exception of support of V4 ICC profiles. Ghostscript supports ICC.1:1998-09 (vers 3.4).

The existing architecture is inefficient in its use of ICC profiles due to their late addition into the code. (Last color space added).

Primary Color Flows in Ghostscript



A Color Rendering Dictionary (CRD) is PostScript's method for defining a mapping from CIEXYZ to a device color. It can be defined with PS procedures (functions) and/or multi-dimensional look-up-tables.

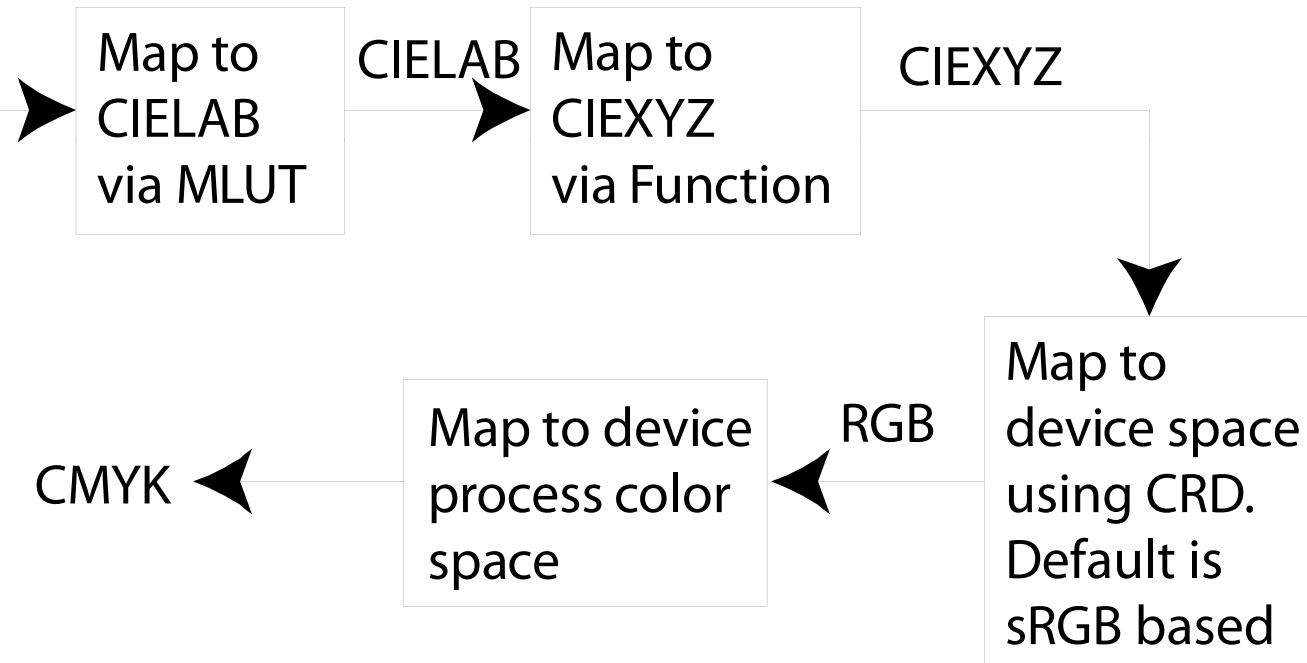
Current ICC Usage in Ghostscript. No Linking!

For each image pixel.....

RGB Image

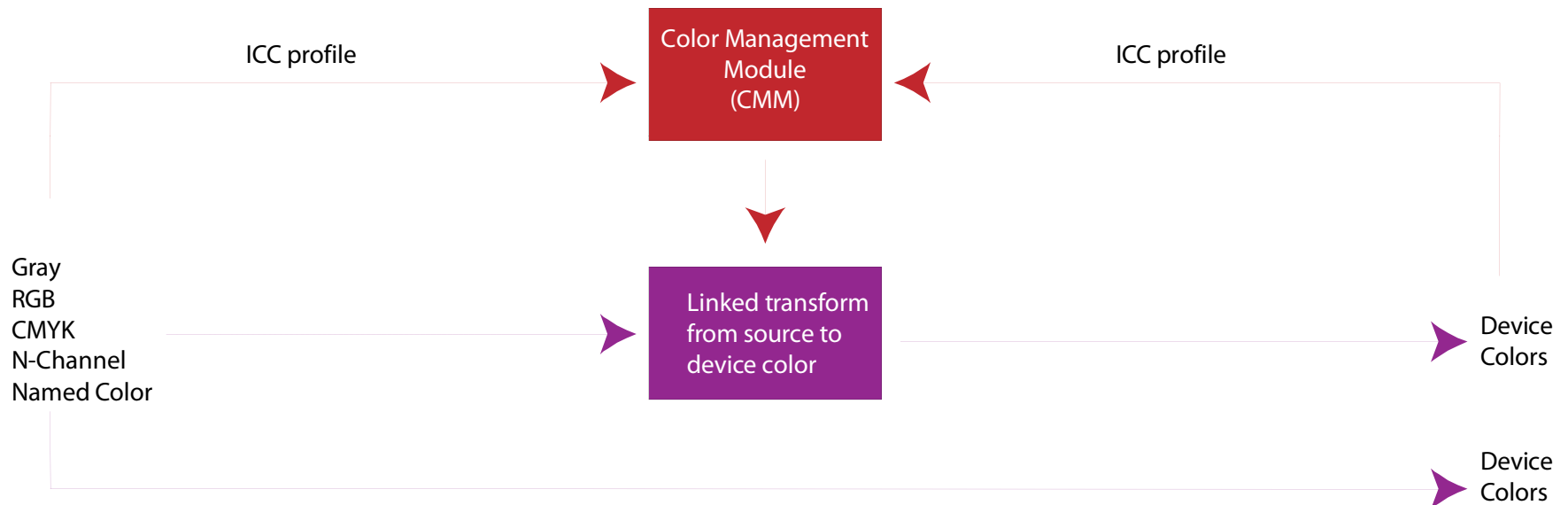


Scanner MLUT
ICC Profile
PCS is CIELAB



Note: If CRD is also an MLUT, we end up going through two MLUTS!

Desired Color Flows

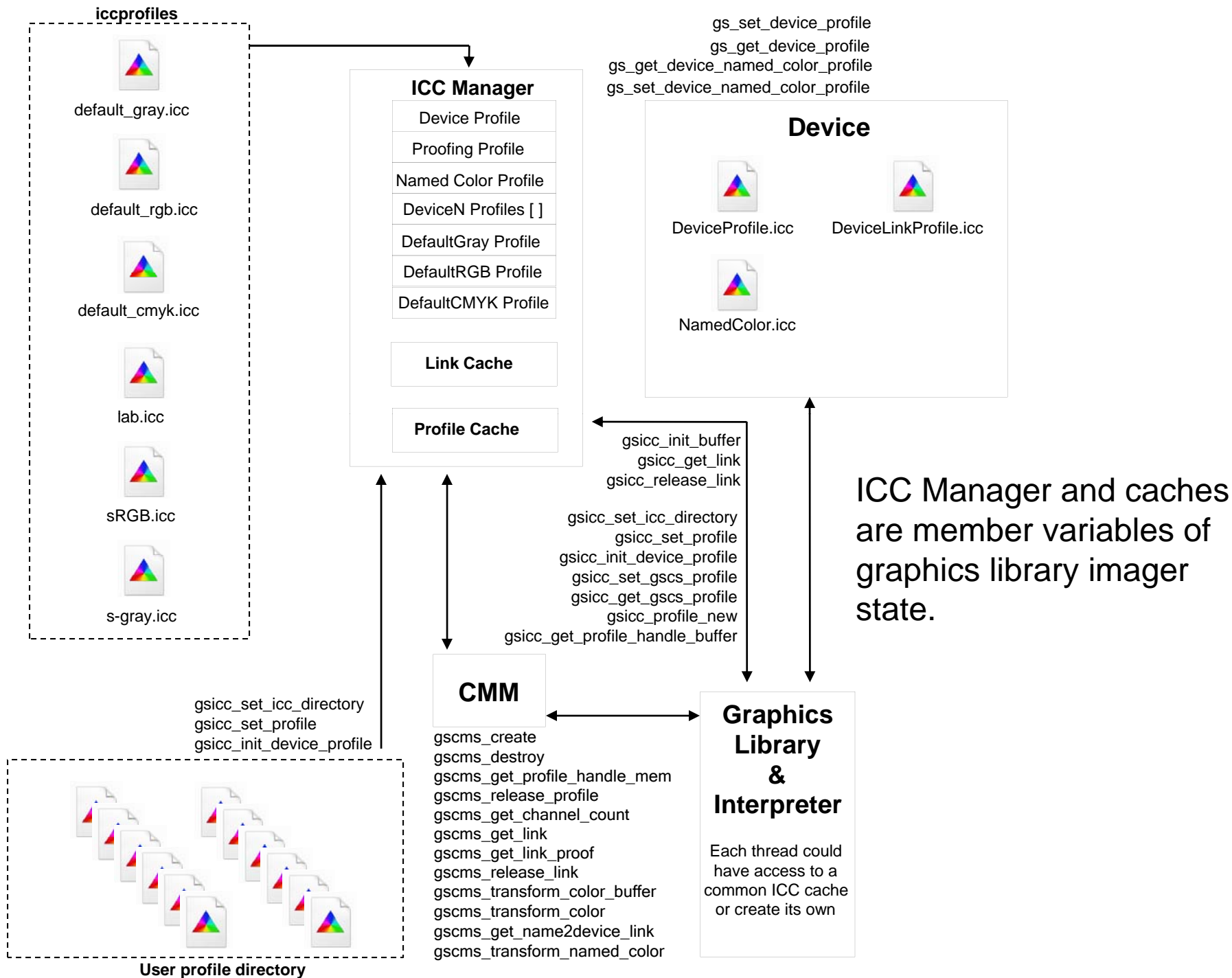


Even if it is ICC centric, Ghostscript will continue to support the color spaces of PDF and PostScript.



Goals of New Ghostscript Color Architecture

- Easy to interface different CMM with Ghostscript.
- Define all color spaces in terms of ICC profiles.
- Cache linked transformations and profiles.
- Easily accessed manager for ICC profiles.
- Devices communicate their ICC profiles and have their ICC profile set.
- Include object type (e.g. image, graphic, text) and rendering intent into the computation of the linked transform.
- Operate efficiently in a multithreaded environment.
- Color manage named colors with ICC named color profile or proprietary format.
- Color management of Device-N colors.





Typical Graphics Library Usage

- Graphics library will request link from link cache.
- Once link obtained, graphics library will apply link to buffers. Typical buffer may be a single scan line.
- When done, graphics library will notify cache.
- Ideal buffer transform case occurs in transparency code when transforming from blending color space during transparency group pop.



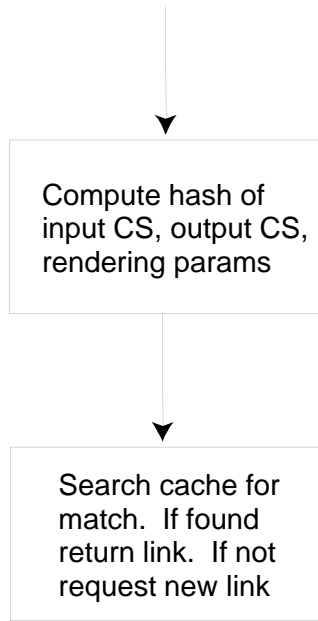
Link Cache

GRAPHICS LIBRARY

```
gsicc_get_link(* pis, *input_colorspace, *output_colorspace, *rendering_params,  
*memory, include_softproof)
```

Link Cache

Hash Code	Ref Count	Link Structure
Hash Code	Ref Count	Link Structure
Hash Code	Ref Count	Link Structure
Hash Code	Ref Count	Link Structure
	.	
	.	
	.	
	.	
Hash Code	Ref Count	Link Structure



Link entries are reference counted.

Links are only released if we are at maximum number (or memory), new request is made and a Ref Count is zero.



Named Colors

	Required	Optional
Pantone Uncoated Yellow	CIELAB	Device Value
Toyo Red	CIELAB	Device Value
Pantone Coated Green	CIELAB	Device Value
	.	
	.	
	.	
Toyo Coated Blue	CIELAB	Device Value

Missing from ICC profile is ability to use tint information. We provide opportunity for CMS to use. If it cannot, then alternate tint transform is used.

A look-up-table.

There is an ICC profile format for named colors.

In many applications, a custom format is used.

For some companies this is their value added.

```
int gscms_xform_named_color(gsicc_link_t *icclink, float tint_value,
const char* ColorName, gx_color_value device_values[] );
```



Conversion of PS and PDF Color Spaces

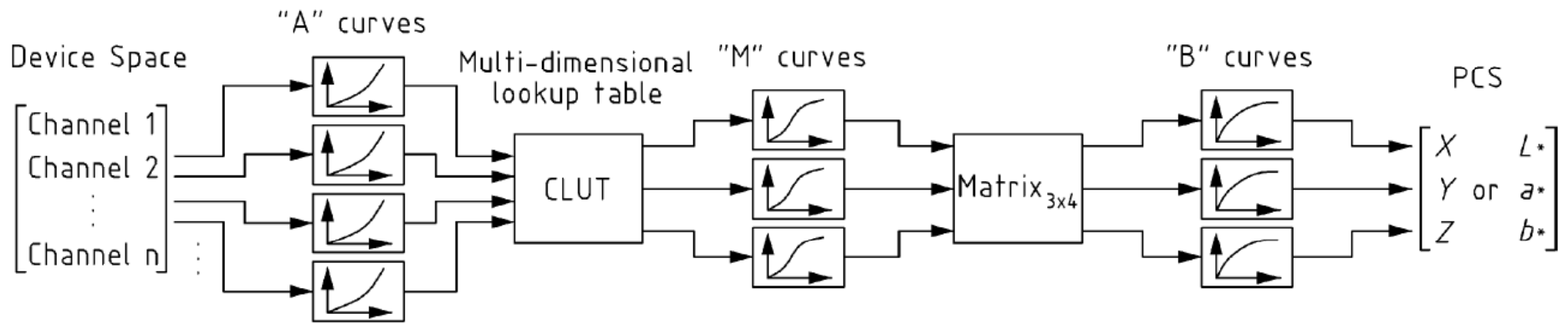
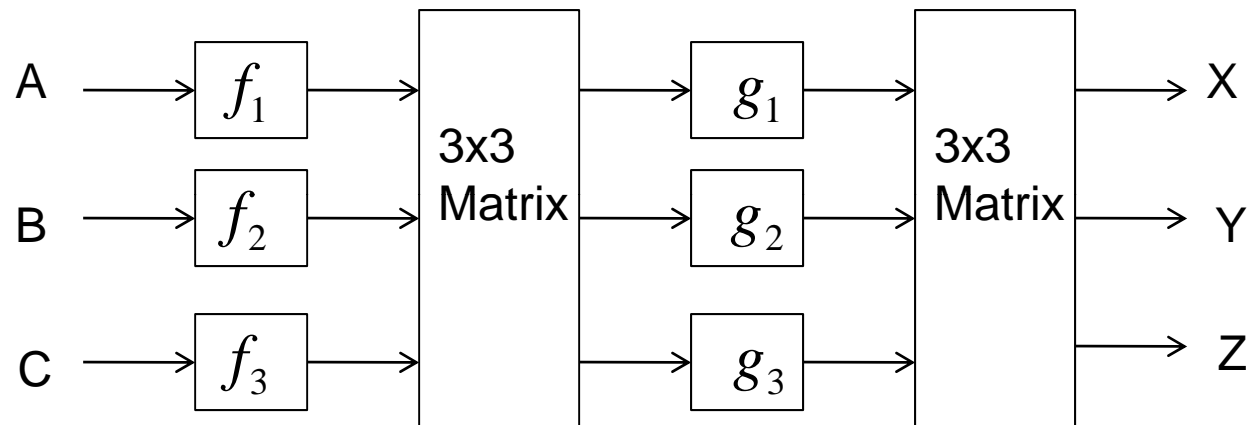
- PS and PDF CIE color spaces will be converted to ICC forms that the CMM can handle.
- If desired CMM could be provided with PS or PDF data in a private tag.
- PS mappings are all 1-way. Device to CIEXYZ or CIEXYZ to Device.
- Procedural mappings will be sampled. This is already done in Ghostscript's joint CIE cache for transforming CIE based colors.
- Because of the multiple matrix operations and procedural mappings, some PS color spaces that do not include MLUTs will give rise to ICC profiles that do include MLUTs.



Profile Cache

- Ghostscript creates ICC profiles from PDF and PS CIE colorspace definitions (e.g. CalRGB, CIEABC, CIEDEFG)
- To avoid repeated creations, these profiles are cached based upon a hash code that is dependent upon the source colorspace definition.
- Profiles are only released if we are at maximum number (or memory), new request is made and a Ref Count is zero.

Example PS CIEABC





Device N color spaces (PDF and PS)

For Device N output, very simple to provide capability for n-color ICC profile.

Many desire to have CM with CMYK and to pass additional spot colors unmolested.

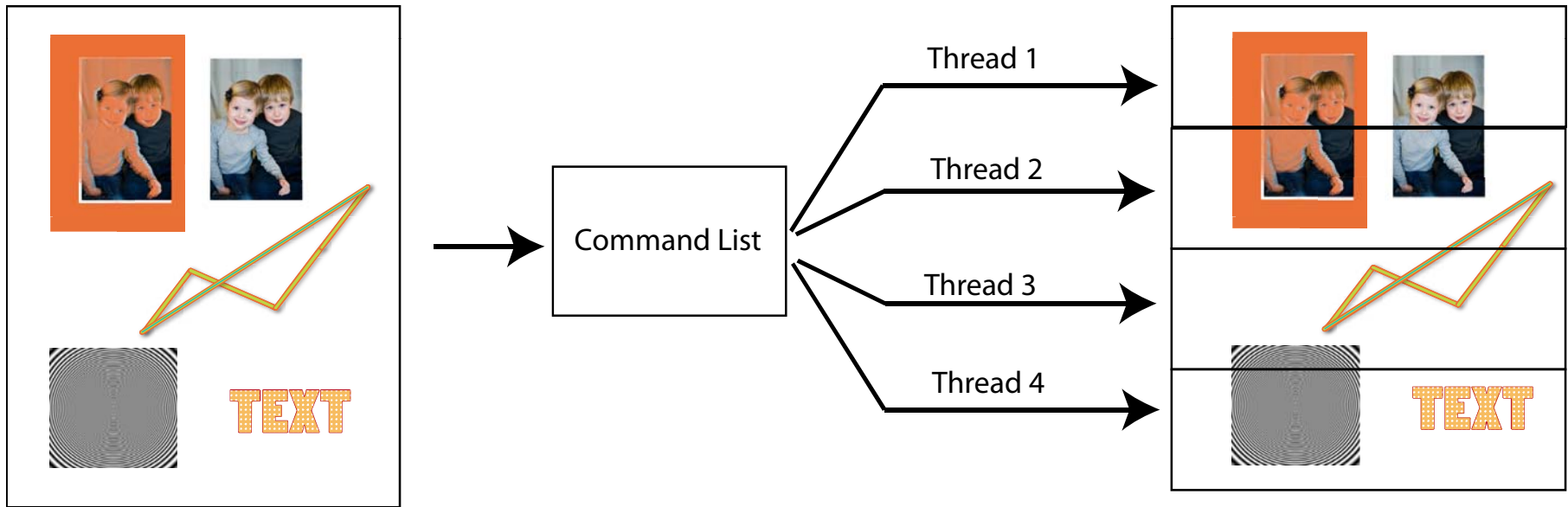
For DeviceN input color, XPS requires ICC profile. PDF and PS use an alternate tint transform. Ghostscript will provide capability to define N-color ICC profile for DeviceN input colors to replace the alternate tint transform if desired.



Ghostscript's Command List

- Ghostscript normally renders in immediate mode.
- Can also generate a command list for asynchronous processing.
- Subdividing a job into bands reduces peak memory usage.
- Threads can process bands simultaneously.

Ghostscript Multithreaded





Command List and ICC Profiles

- Due to table look-ups and interpolations color conversion is expensive.
- Would like to distribute the load in threaded rendering.
- ICC profiles are embedded in the command list.



Multi-Threaded Environment

- During command list read phase, each thread obtains the same initial imaging state that includes a pointer to the primary link cache.
- In a single page, it is very likely that similar links will be needed.
- This suggests sharing a common cache amongst the threads.
- Links are reference counted to ensure only unreferenced ones are removed.
- It may be necessary to have a lock and release feature on the cache.
- If desired, a thread can start its own private link cache. This is similar to how each thread has its own evolving imaging state.
- It will be up to the thread to destroy the cache when it completes.



Current Status

- ICC branch started in SVN.
http://svn.ghostscript.com/ghostscript/branches/icc_work
- Currently interfacing to littleCMS <http://www.littlecms.com/> Marti Maria.
- Code is almost complete. Reviewing regression differences.