DevCon 2016

A Spectral Workflow for Encoding & Analyzing Artwork with iccMAX

Ben Bodner, Roy S Berns
Munsell Color Science Laboratory, Rochester Institute of Technology

Overview

- Purpose
  - Accurate reproduction and analysis of artwork.
- Imaging Workflow
  - Spectral capture, processing, and display.
- iccMAX Implementation
  - Profile types and tags that mirror a MatLab code workflow.
- iccMAX Profile Generator
  - Graphical tool that helps create and debug profiles.

Purpose

- Why is spectral analysis important for art?
  - Produce digital archives that can be accurately reproduced.
  - Predict appearance under different light sources.
  - Determine component colorants, aiding in restoration and providing historical context.

620nm 720nm 720 – 620 nm

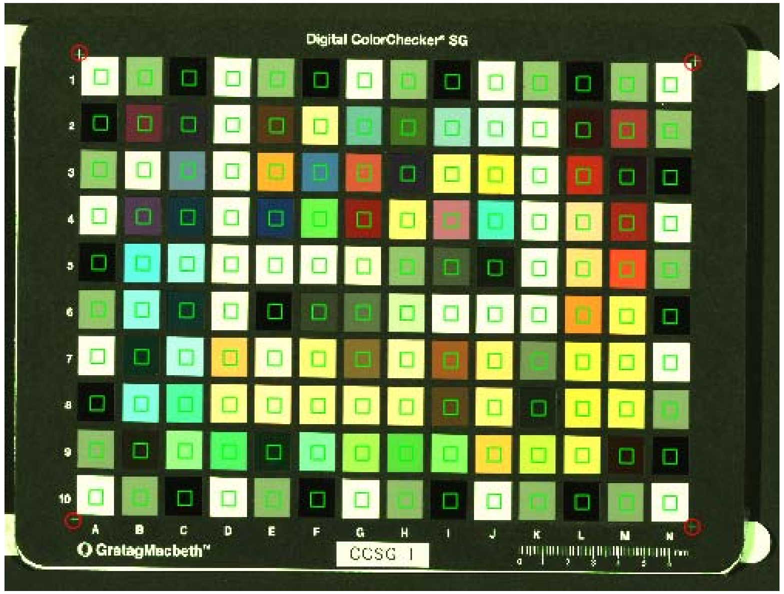
Purpose

- How would iccMAX help?
  - Current workflow requires custom software tools for analysis.
  - This makes data sharing and collaboration difficult.

Imaging Workflow

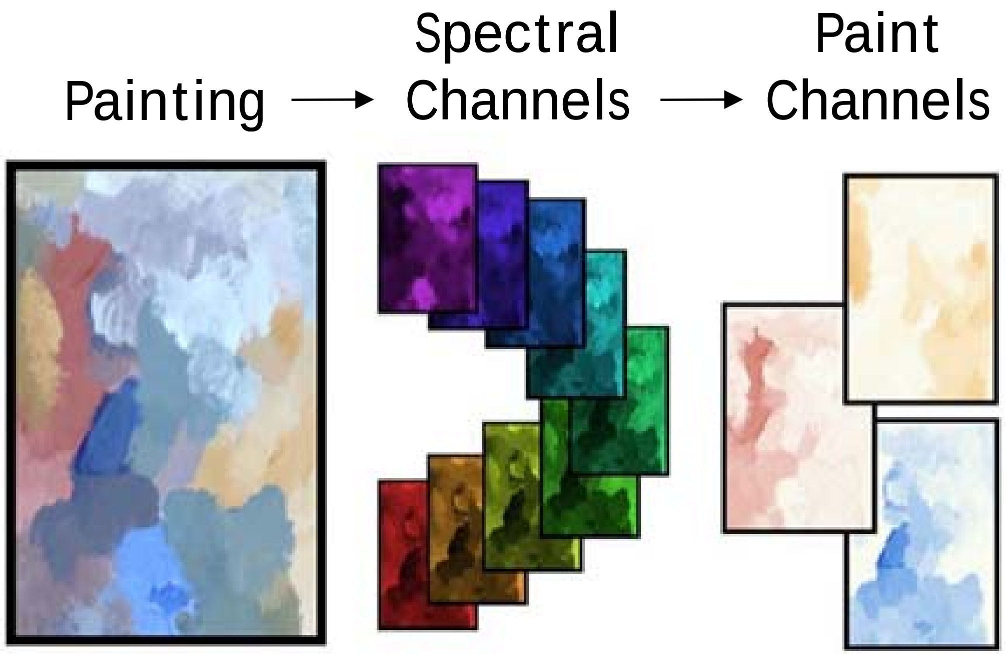
Imaging Workflow

- Multispectral capture
  - 6 channel camera (Dual-RGB)
  - Camera-to-Spectral & Camera-to-Tristimulus matrices are optimized using a color target (ColorCheckerSG or custom paint target)



Imaging Workflow

- Paint Concentration Estimation
  - Linear algorithm estimates amount of component paints at every pixel.
  - Result is a paint-channel image.



Imaging Workflow

- Paint Concentration Estimation
  - Complex linear algorithm involving spectral integration, inverse matrices, iterative operations and a lot of data.
  - iccMAX can handle it!

$$R_i = \frac{R_m - \kappa_{ins}\kappa_1}{(1-\kappa_1)(1-\kappa_2) + \kappa_2 R_m - \kappa_{ins}\kappa_1\kappa_2}$$

$$D_k = \frac{-2R_{i,a}{}^2}{s_a(1-R_{i,a}^2)}, D_s = \frac{R_{i,a}(1-R_{i,a})}{s_a(1+R_{i,a})}$$

$$\begin{bmatrix} X^P \\ Y^P \\ Z^P \end{bmatrix} = \begin{bmatrix} \bar{x} \\ \bar{y} \\ \bar{z} \end{bmatrix} E(D_k k + D_s s)$$

$$\begin{bmatrix} c_{p1} \\ c_{p2} \\ c_{p3} \end{bmatrix} = \begin{bmatrix} (X_{p1}^P - X_w^P) & (X_{p2}^P - X_w^P) & (X_{p3}^P - X_w^P) \\ (Y_{p1}^P - Y_w^P) & (Y_{p2}^P - Y_w^P) & (Y_{p3}^P - Y_w^P) \\ (Z_{p1}^P - Z_w^P) & (Z_{p2}^P - Z_w^P) & (Z_{p3}^P - Z_w^P) \end{bmatrix}^{-1} \begin{bmatrix} (X_a^P - X_w^P) \\ (Y_a^P - Y_w^P) \\ (Z_a^P - Z_w^P) \end{bmatrix}$$

$$\left(\frac{K}{S}\right)_m = \frac{c_{p1}k_{p1} + c_{p2}k_{p2} + c_{p3}k_{p3} + c_w k_w}{c_{p1}s_{p1} + c_{p2}s_{p2} + c_{p3}s_{p3} + c_w s_w}$$

$$R_i = 1 + \left(\frac{K}{S}\right) - \sqrt{\left(\frac{K}{S}\right)^2 + 2\left(\frac{K}{S}\right)}$$

$$R_m = \frac{((1-\kappa_1)(1-\kappa_2)R_i)}{(1-\kappa_2 R_i)} + \kappa_{ins}\kappa_1$$

$$\begin{bmatrix} \Delta c_{p1} \\ \Delta c_{p2} \\ \Delta c_{p3} \end{bmatrix} = \begin{bmatrix} (X_{p1}^P - X_w^P) & (X_{p2}^P - X_w^P) & (X_{p3}^P - X_w^P) \\ (Y_{p1}^P - Y_w^P) & (Y_{p2}^P - Y_w^P) & (Y_{p3}^P - Y_w^P) \\ (Z_{p1}^P - Z_w^P) & (Z_{p2}^P - Z_w^P) & (Z_{p3}^P - Z_w^P) \end{bmatrix}^{-1} \begin{bmatrix} X_a - X_m \\ Y_a - Y_m \\ Z_a - Z_m \end{bmatrix}$$

Imaging Workflow

- Displaying Paint Images
  - Absorption/Scattering to Reflectance.
  - Spectral Reflectance to Tristimulus or Display RGB.

$$\left(\frac{K}{S}\right)_m = \frac{c_{p1}k_{p1} + c_{p2}k_{p2} + c_{p3}k_{p3} + c_w k_w}{c_{p1}s_{p1} + c_{p2}s_{p2} + c_{p3}s_{p3} + c_w s_w}$$

$$R_i = 1 + \left(\frac{K}{S}\right) - \sqrt{\left(\frac{K}{S}\right)^2 + 2\left(\frac{K}{S}\right)}$$

$$R_m = \frac{((1-\kappa_1)(1-\kappa_2)R_i)}{(1-\kappa_2 R_i)} + \kappa_{ins}\kappa_1$$

iccMAX Implementation

iccMAX Implementation

- Multispectral Capture & Encoding
  - Input Device Profile (SCNR)
  - 6 Channel Data Color Space (6CLR)
  - Absolute Colorimetric PCS Transformation (A2B3, 6x3 matrix)
  - Absolute Reflectance Spectral PCS (D2B3, 6x36 matrix)
  - Matrices generated from non-linear optimization outside of iccMAX.

```xml
<ProfileDeviceClass>scnr</ProfileDeviceClass>
<DataColourSpace>6CLR</DataColourSpace>
<PCS>XYZ</PCS>
<SpectralPCS>rs0024</SpectralPCS>
```

iccMAX Implementation

- Paint Concentration Estimation
  - Material Identification Profile (MID) allows for custom encoding channels and transformation.
  - Device Channel Input transformed to Material Connection Space using iccMAX Programmable Calculator (A2M0).
  - Material Channels have custom names.

```
<ProfileDeviceClass>mid</ProfileDeviceClass>
<DataColourSpace>nc001C</DataColourSpace>
<MCS>mc0003</MCS>
...
<ArrayTags>
<utf8TextType><TextData>C_redOxd</TextData></utf8TextType>
<utf8TextType><TextData>C_ylwOcr</TextData></utf8TextType>
<utf8TextType><TextData>C_ultBlu</TextData></utf8TextType>
</ArrayTags>
```

iccMAX Implementation

- Paint Concentration Estimation
  - Paint, Illuminant, CMF spectral data are stored as Matrix Sub-elements.
  - Pseudo-iterative step written in Calculator Sub-element, called multiple times.
  - Calculator sub-element holds it’s own Matrix “sub-sub”-elements.
  - Use of calculator requires careful tracking of stack contents.

```
</SubElements>
<MainFunction>
{
in(0,28) %Rm
copy(28,1) .65 %Rm,Rm,0.65
smul(28) %Rm,.65*Rm
1 1 .03 .65 %Rm,.65*Rm,1,1,.03,.65
sub(2) mul sadd(28) div(28) %Rm/.65*R
tsav(0,28) %main = Ri, temp 0,28 = Ri
neg(28) 1 sadd(28) sq(28) %(1-Ri)^2
tget(0,28) 2 smul(28) %(1-Ri)^2,2*Ri
div(28) %ka
0 0 0 1 mtx(0) %ka,kw
sub(28) %ka-kw
tget(0,28) sq(28) 0 2 sub smul(28) %k
1 copy(1,27) tget(0,28) sq(28) sub(28
mul(28) %Dk*(ka-kw)
1 mtx(2) mul(28) mtx(3) %CMFx[Dk*(ka-
tput(28,3) % main stack is empty, tem
tget(0,28) 1 copy(1,27) %Ri, Sa =1
calc(4)%Matrix B
tget(28,3) solv(3,3) pop tput(31,3) %
out(0,3)
%tget(0,28) in(0,28) tget(31,3) %Ri,R
%calc(5) calc(5) calc(5) calc(5) calc
}
</MainFunction>
```

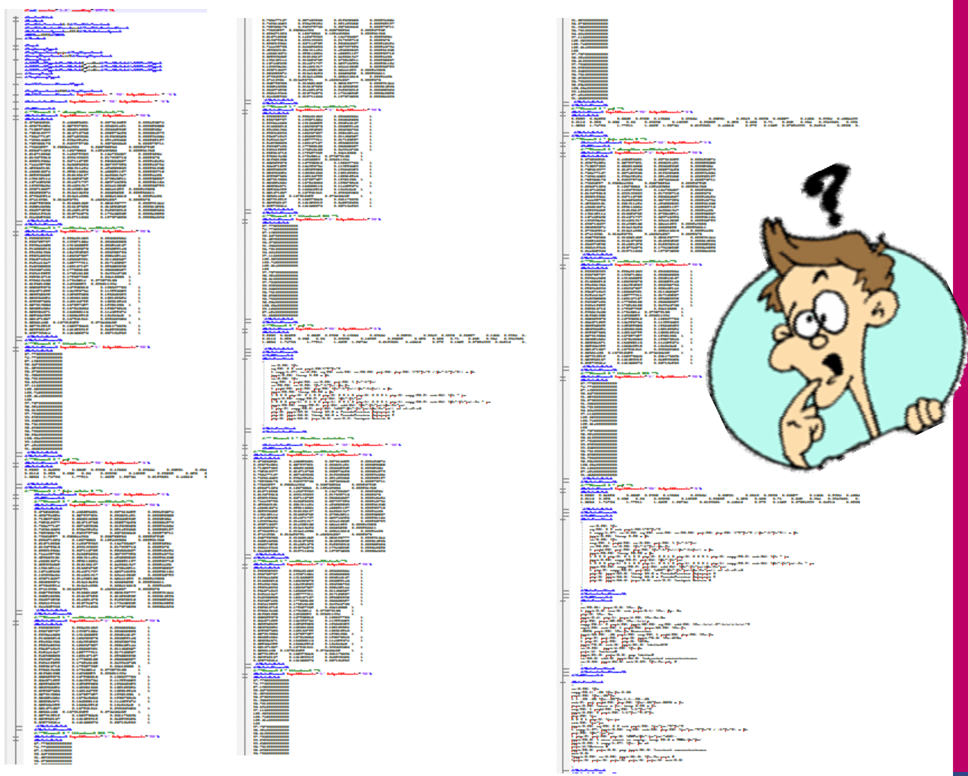
iccMAX Implementation

- Displaying Paint Images
  - Material Visualization Profile (MVIS) allows for transformation from MCS to Absolute Spectral (M2S3) or Colorimetric (M2B3) or Device (M2A0).
  - MID output channels and MVIS input channels have matching names.
  - MID profiles have MCS flag – MID channels are a subset of the single MVIS profile, which contains all paint channels.

iccMAX Graphical Tool

- What does it do?
  - MatLab GUI Auto-generates XML code for iccMAX profiles based on user input.
  - Why is this useful? →

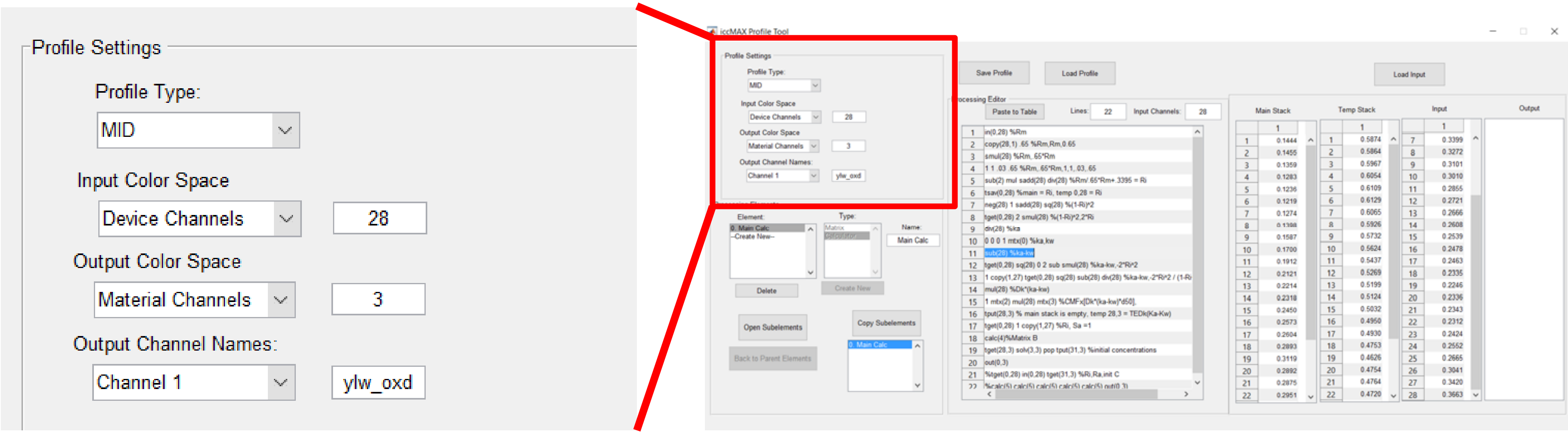
XML for One MID Profile



iccMAX Profile Generator

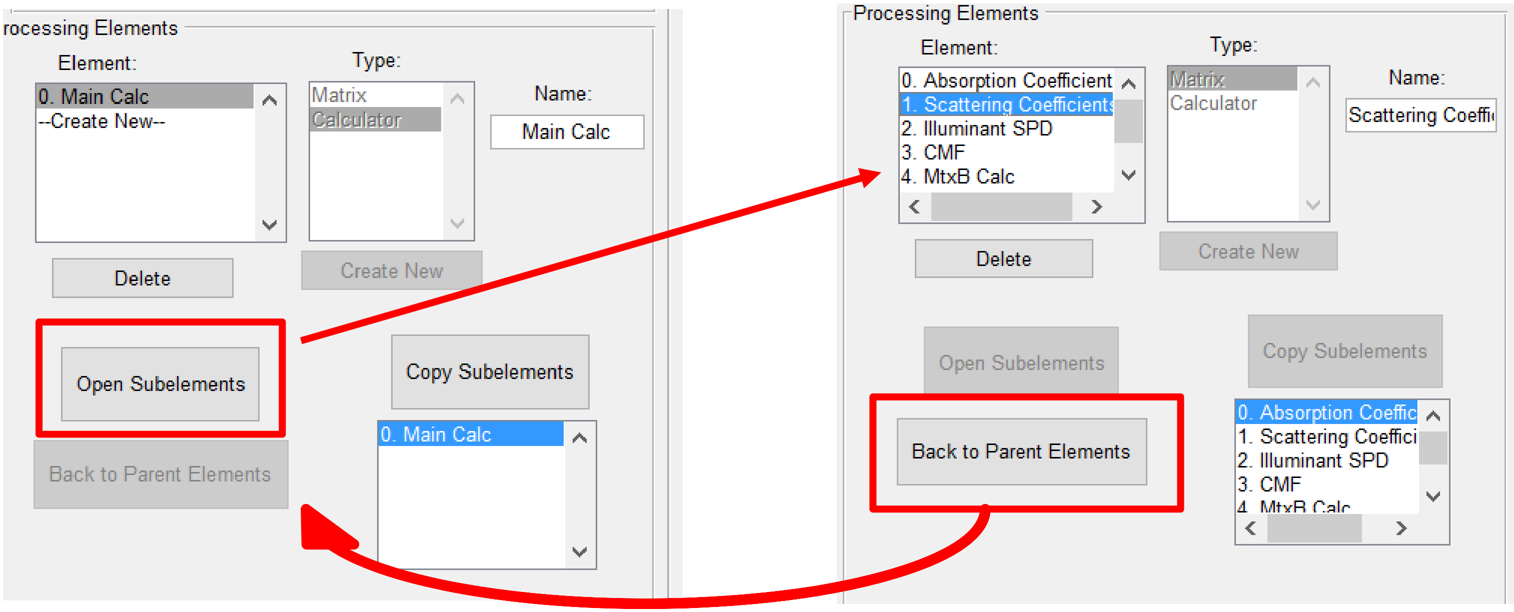
iccMAX Profile Generator

- Profile Selection
  - User selects profile type, input & output channels.



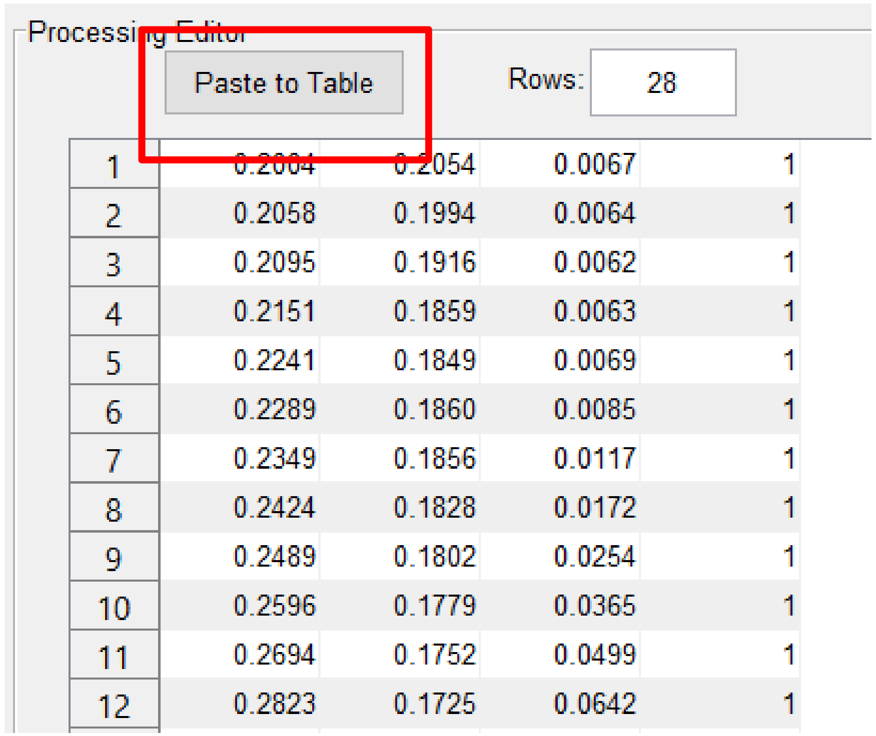
iccMAX Profile Generator

- iccMAX Processing
  - Processing elements and sub-elements can be added, deleted, and manipulated using a graphical layout.



iccMAX Profile Generator

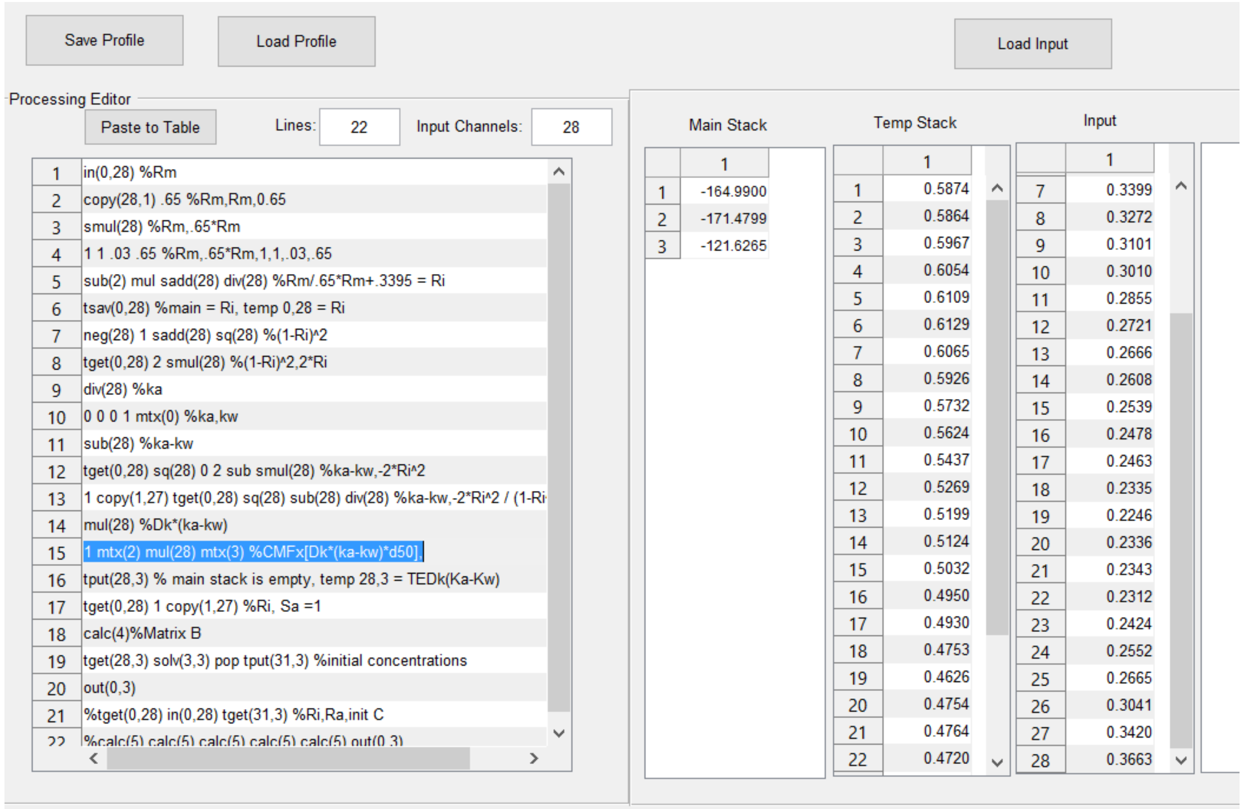
- iccMAX Processing
  - Both data and code can be copied from a text file, excel document, or MatLab variable and pasted into the tool.
  - Calculator code can be written directly in the tool.



1	0.2004	0.2054	0.0067	1
2	0.2058	0.1994	0.0064	1
3	0.2095	0.1916	0.0062	1
4	0.2151	0.1859	0.0063	1
5	0.2241	0.1849	0.0069	1
6	0.2289	0.1860	0.0085	1
7	0.2349	0.1856	0.0117	1
8	0.2424	0.1828	0.0172	1
9	0.2489	0.1802	0.0254	1
10	0.2596	0.1779	0.0365	1
11	0.2694	0.1752	0.0499	1
12	0.2823	0.1725	0.0642	1

iccMAX Profile Generator

- iccMAX Processing
  - Calculator code can be debugged in the tool.
  - Selecting a line of code shows the resulting calculator stacks.



iccMAX Profile Generator

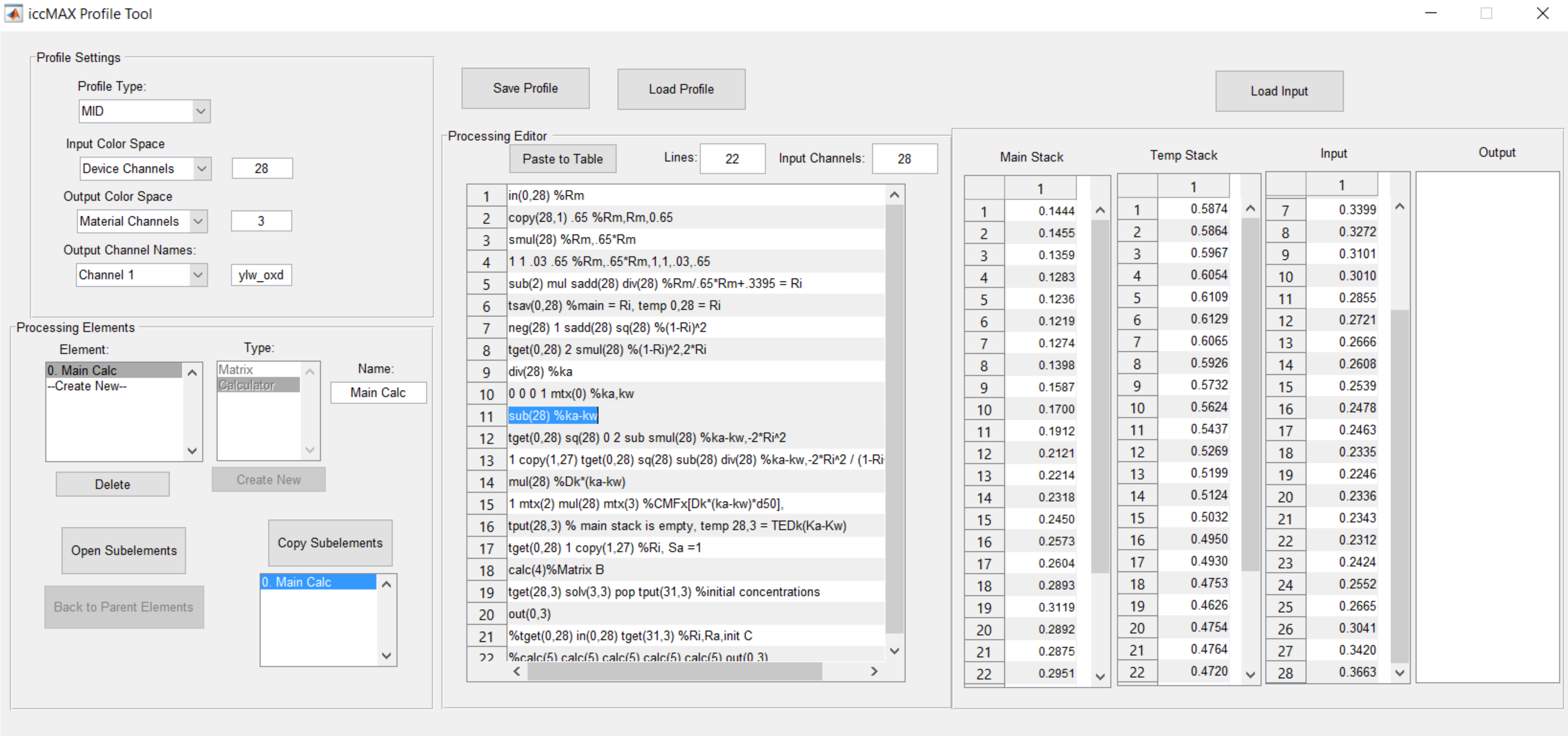
- ## Current Status
  - The MatLab GUI is far from complete. It lacks most profile types, processing elements, and calculator operations.
  - Collaboration on a more complete tool would be welcome!

Acknowledgements

- This research was supported by a grant from the Andrew W. Mellon Foundation.

- Thanks to Max Derhak for providing constant iccMAX support!

THANK YOU


iccMAX Profile Tool
Profile Settings
Profile Type:
MID
Input Color Space
Device Channels
28
Output Color Space
Material Channels
3
Output Channel Names:
Channel 1
ylw_oxd
Processing Elements
Element:
0. Main Calc
--Create New--
Type:
Matrix
Calculator
Name:
Main Calc
Delete
Create New
Open Subelements
Copy Subelements
Back to Parent Elements
0. Main Calc
Save Profile
Load Profile
Load Input
Processing Editor
Paste to Table
Lines:
22
Input Channels:
28
1 in(0,28) %Rm
2 copy(28,1) .65 %Rm,Rm,0.65
3 smul(28) %Rm,.65*Rm
4 1 1 .03 .65 %Rm,.65*Rm,1,1,.03,.65
5 sub(2) mul sadd(28) div(28) %Rm/.65*Rm+.3395 = Ri
6 tsav(0,28) %main = Ri, temp 0,28 = Ri
7 neg(28) 1 sadd(28) sq(28) %(1-Ri)^2
8 tget(0,28) 2 smul(28) %(1-Ri)^2,2*Ri
9 div(28) %ka
10 0 0 0 1 mtx(0) %ka,kw
11 sub(28) %ka-kw
12 tget(0,28) sq(28) 0 2 sub smul(28) %ka-kw,-2*Ri^2
13 1 copy(1,27) tget(0,28) sq(28) sub(28) div(28) %ka-kw,-2*Ri^2 / (1-Ri
14 mul(28) %Dk*(ka-kw)
15 1 mtx(2) mul(28) mtx(3) %CMFx[Dk*(ka-kw)*d50],
16 tput(28,3) % main stack is empty, temp 28,3 = TEDk(Ka-Kw)
17 tget(0,28) 1 copy(1,27) %Ri, Sa =1
18 calc(4)%Matrix B
19 tget(28,3) solv(3,3) pop tput(31,3) %initial concentrations
20 out(0,3)
21 %tget(0,28) in(0,28) tget(31,3) %Ri,Ra,init C
22 %calc(5) calc(5) calc(5) calc(5) calc(5) out(0,3)
Main Stack
1
1 0.1444
2 0.1455
3 0.1359
4 0.1283
5 0.1236
6 0.1219
7 0.1274
8 0.1398
9 0.1587
10 0.1700
11 0.1912
12 0.2121
13 0.2214
14 0.2318
15 0.2450
16 0.2573
17 0.2604
18 0.2893
19 0.3119
20 0.2892
21 0.2875
22 0.2951
Temp Stack
1
1 0.5874
2 0.5864
3 0.5967
4 0.6054
5 0.6109
6 0.6129
7 0.6065
8 0.5926
9 0.5732
10 0.5624
11 0.5437
12 0.5269
13 0.5199
14 0.5124
15 0.5032
16 0.4950
17 0.4930
18 0.4753
19 0.4626
20 0.4754
21 0.4764
22 0.4720
Input
1
7 0.3399
8 0.3272
9 0.3101
10 0.3010
11 0.2855
12 0.2721
13 0.2666
14 0.2608
15 0.2539
16 0.2478
17 0.2463
18 0.2335
19 0.2246
20 0.2336
21 0.2343
22 0.2312
23 0.2424
24 0.2552
25 0.2665
26 0.3041
27 0.3420
28 0.3663
Output