TAGA NextGen

# Efficient and Accurate Color Conversions for Multicolor Printing

**Dr. Hanno Hoffstadt, GMG**

TAGA NextGen

2026

# Motivation

- For multicolor printing (> 4 inks),
- color profiles **grow exponentially**
- and eventually **must lose resolution**

- GMG needed a profile format
- for **many colors**
- with **better accuracy**
- and still **high performance**

TAGA NextGen

2026

# Efficient and accurate color conversions for multicolor printing

- Color conversion tables
- Accuracy expectations

- **Subspaces**: store only what we need
- **Estimation**: beyond the stored subspaces
- **Correction**: increase accuracy of estimations

# Multicolor use case examples

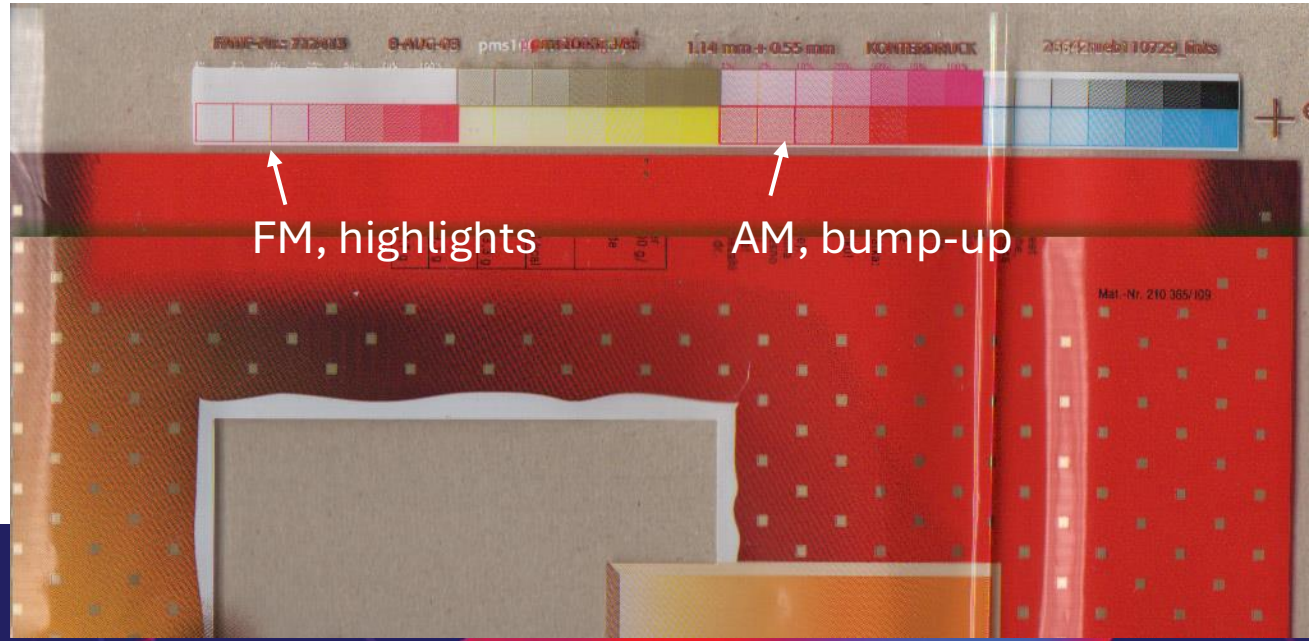
## CMYK + spot colors

- CMYK + pink, blue, brown, white
- solid white does not need proofing

- special case: ECG
- overprints are much more important

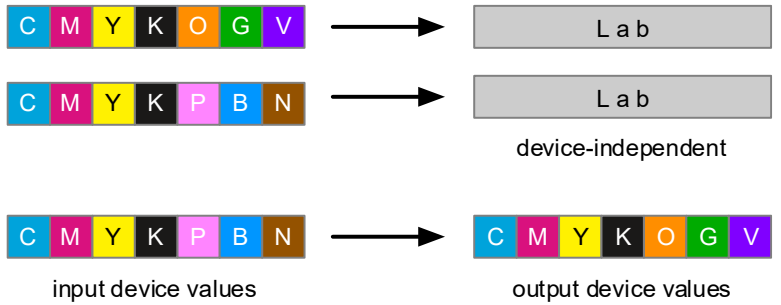

## Use of dual screens (2 channels/ink)

- CMYK + red, gold, white
- AM for flat areas
- FM for flexo gradients fading to 0%
- partial white may need proofing too



# Color conversion tables

- Interpolation table for ***n*** input colors
- Here: output is Lab, device-link, …




- Each input channel is sampled between 0 and 100%
- More sampling steps ***s*** → more accuracy → larger size

- All combinations must be stored: $s^n$
  (e. g. $17^4 = 83521$ for CMYK)

# Accuracy expectations

- We know CMYK well
- we expect **same accuracy** for multicolor
  …for proofing and for conversion
  (e. g. conventional to digital)

- 7+ channels: far **too many combinations**!

  $17^{7} = 17^{3} \times 17^{4}$

  → one 7-c profile ≈ 5000 CMYK profiles

# How we cope with multicolor

- Treat CMYK and spot colors separately
  (CMYK accurate; overprints guessed; no fingerprint needed)

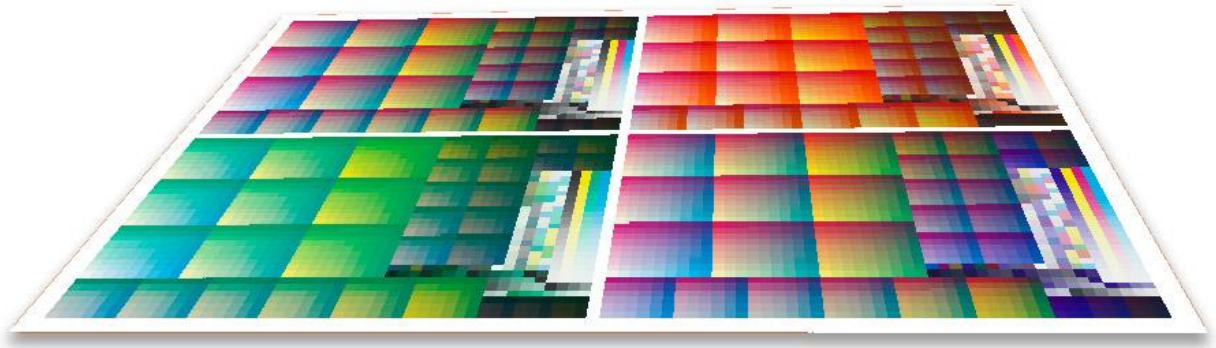
- Use n-color ICC profiles
  (all overprints in table, lower resolution, CMYK less accurate)

- Use multiple tables for **needed subsets**
  (for 4-color subsets, same accuracy as CMYK possible)

# Examples which use selected subsets

- Esko Equinox, …
  use e. g.

CMYK OMYK



CGYK CMYV

# 1. **Subspaces**: store what we need

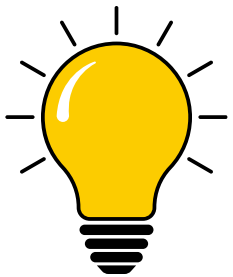
"Subspace" = subset of the *n*-color (*n*-dim.) device space

Separations: often **≤ 4 overprinting inks**

→ (some) accurate 4-c tables are enough

at most 35 such tables for 4 out of 7 colors: $\binom{7}{4} = 35$

Lower dimension → no size problem!



# Subspace tables for 4-c, 17 steps

- Full table would have $17^n$ entries
- Each subspace table: $17^4$ entries

- 7-color input: 35 tables (less than 1%)

- 7-color dual screen (**14 channels**): 1001 tables
- ≈ 84 million entries – large but possible
- 2 trillion times smaller than full table

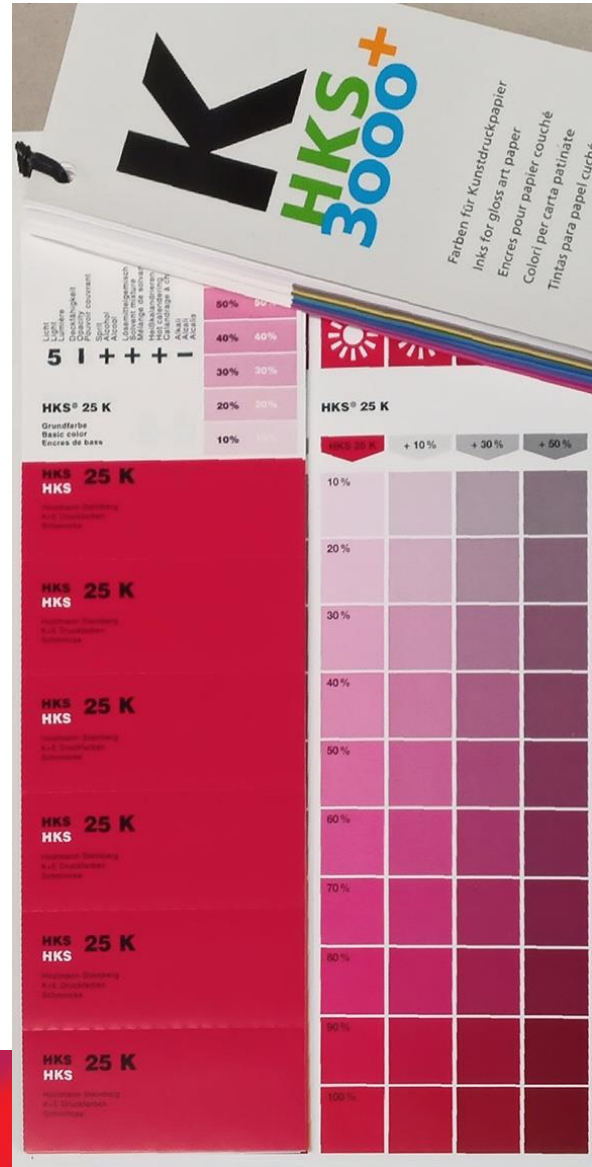
# Storing swatch books

Pantone Solid Coated swatch book

- store **1-c subspace table per ink** (≈2000)
  (2 steps per ink, 0% and 100%)

HKS 3000 swatch book (duplex spot + K)

- store **2-c subspace table per ink** (≈100)
  (11 steps per spot ink, 5 steps for K)

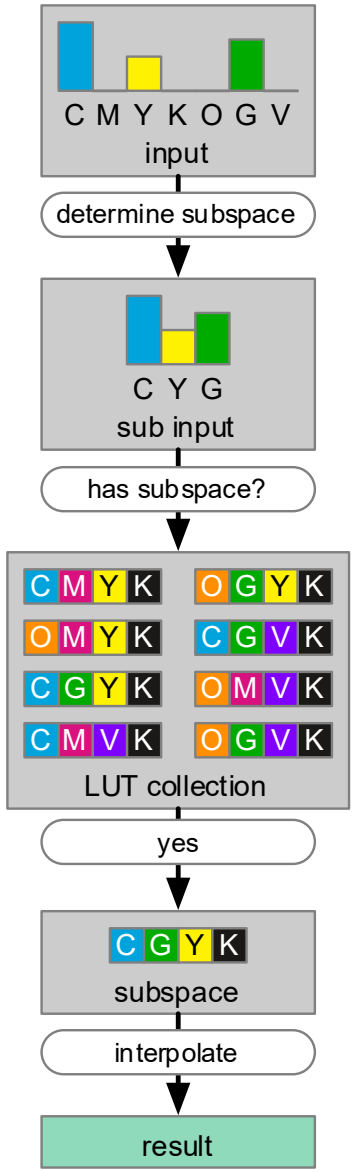


# Subspace look-up

- input: ***n*** device values (0-100%)
- determine subspace: channels > 0

- has subspace table?
  look-up (interpolate), done

- TO DO: what if there is no table?



# Overprints with more than 4 colors

Such separations exist:

- Inkjet: hide nozzle artifacts using as many inks as possible
- Conventional: register marks in all separations (***n*** x 100%)

- Could we store all subspaces up to 5, 6, 7 colors?

| subspace dim out of 7 colors | 4 | 5 | 6 | 7 = full |
|---|---|---|---|---|
| total table entries for 17 steps | 3 million | 30 million | 169 million | 410 million |

- Q: do we need full accuracy?

# Problem statement

- Enable conversion of up to ***n*** overprints
- Be as accurate as possible
- But avoid exponential size problem

Approach:

- **additional collections** for more overprints
- with **lower resolution** at **higher dimension**

# Multiple collections, 7-c example

Compare collections for 4-c to 7-c overprints

- total table entries (counting redundant entries)

| subspace dim out of 7 colors | 4 | 5 | 6 | 7 = full |
|---|---|---|---|---|
| 17 steps (every 6.25 %) | **3 million** | 30 million | 169 million | 410 million |
| 9 steps (every 12.5 %) | 229,635 | **1.2 million** | 3.7 million | 4.8 million |
| 7 steps (subset of 9) | 84,035 | 352,947 | 823,543 | **823,543** |
| or 5 steps (every 25 %) | 21,875 | 65,625 | 109,375 | **78,125** |

→ could e. g. use 3 collections for 4-c, 5-c, and 7-c subspaces with only 5 million entries

# How to use: estimate and correct

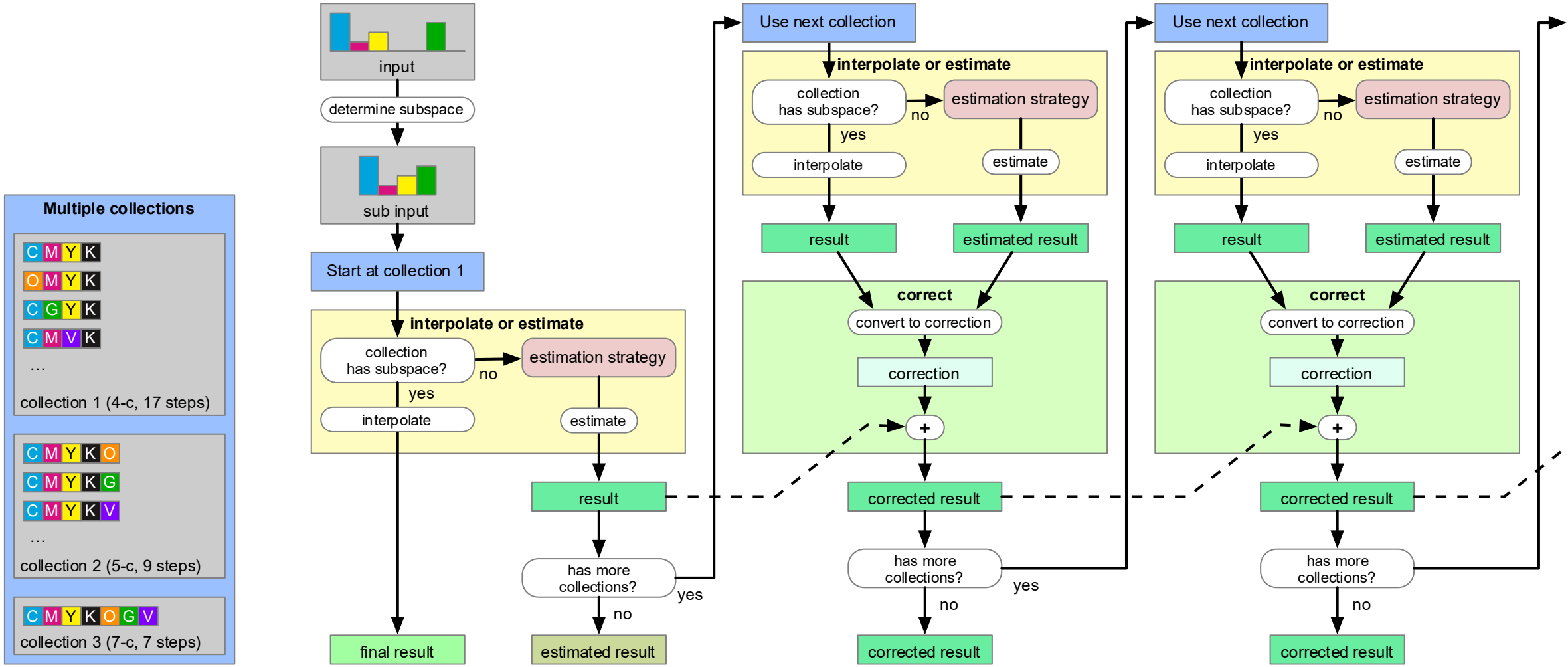
Start at collection with lower dimension (highest resolution)

- Has subspace? Then **look up** overprint, done
- Else **estimate** overprint

While collection for higher dimension exists:

- Has subspace? Use **lookup** to adjust the estimate, done
- Else **estimate** correction, adjust the estimate, repeat

# How to use: estimate and correct



# 2. **Estimation**: outside of subspaces

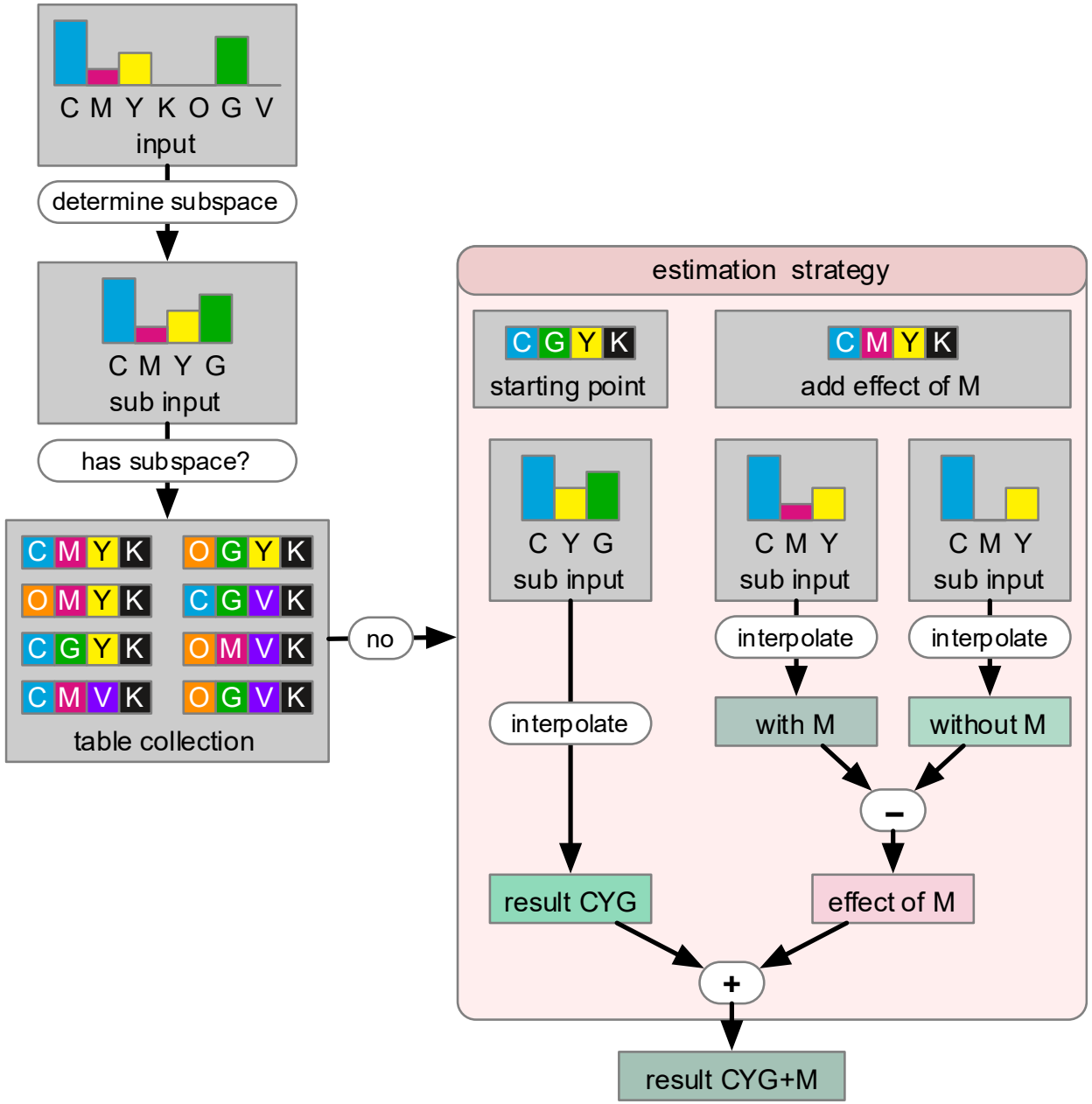
- Inspired by applications for designers
- CMYK is accurate (ICC profile)
- spot colors are like transparent overlays (simple estimate)

What could our color engine do?

- Idea: **combine data** from multiple tables
- like **blending modes**, for Lab and for device-link values

# Estimation example

- This collection has no table for the input (CMYG)
- Split the input into subsets:
- **CYG** – covers most of the input
- Lookup result for CYG in CGYK
- **CMY** – adds M to CY, in CMYK
- Get effect of M from 2 look-ups (with M, and with M=0)
- Apply its effect to the CYG result



# A fast estimation strategy

- Sort colors by value
- Use highest ink amounts as starting point
- Add effect of lower contributions ink by ink
- Done

- Other strategies are available
- Trade-off speed / smoothness

# 3. **Correction**: improve our estimates

- When we know **better values** than just estimated overprints
- from a few measured patches, or from a model

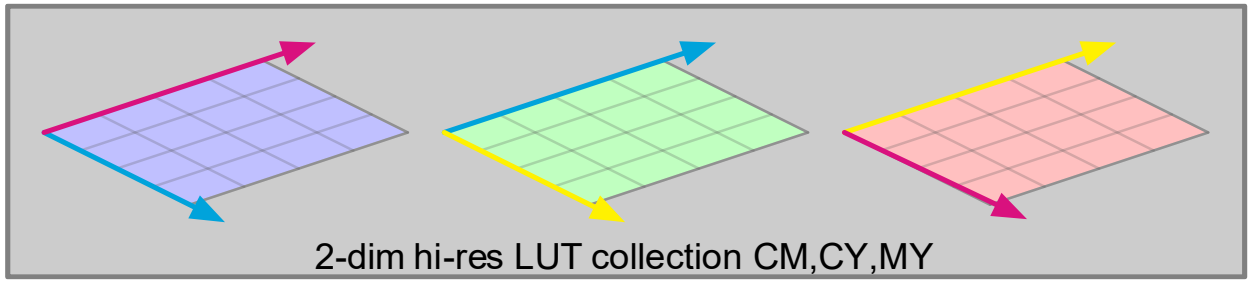
- Add a table collection of higher-dimensional tables

- remember: must use lower resolution to avoid the size problem

# Correction example: two table collections

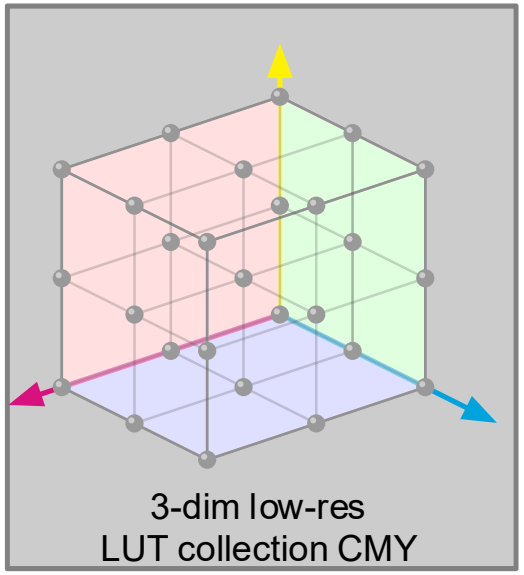
Collection with 2-dim subspaces

- High-resolution CM, CY, MY
- can estimate CMY




Collection with 3-dim subspace CMY

- contains **correct data** for CMY
- at a **lower resolution**
- correction = difference to estimate

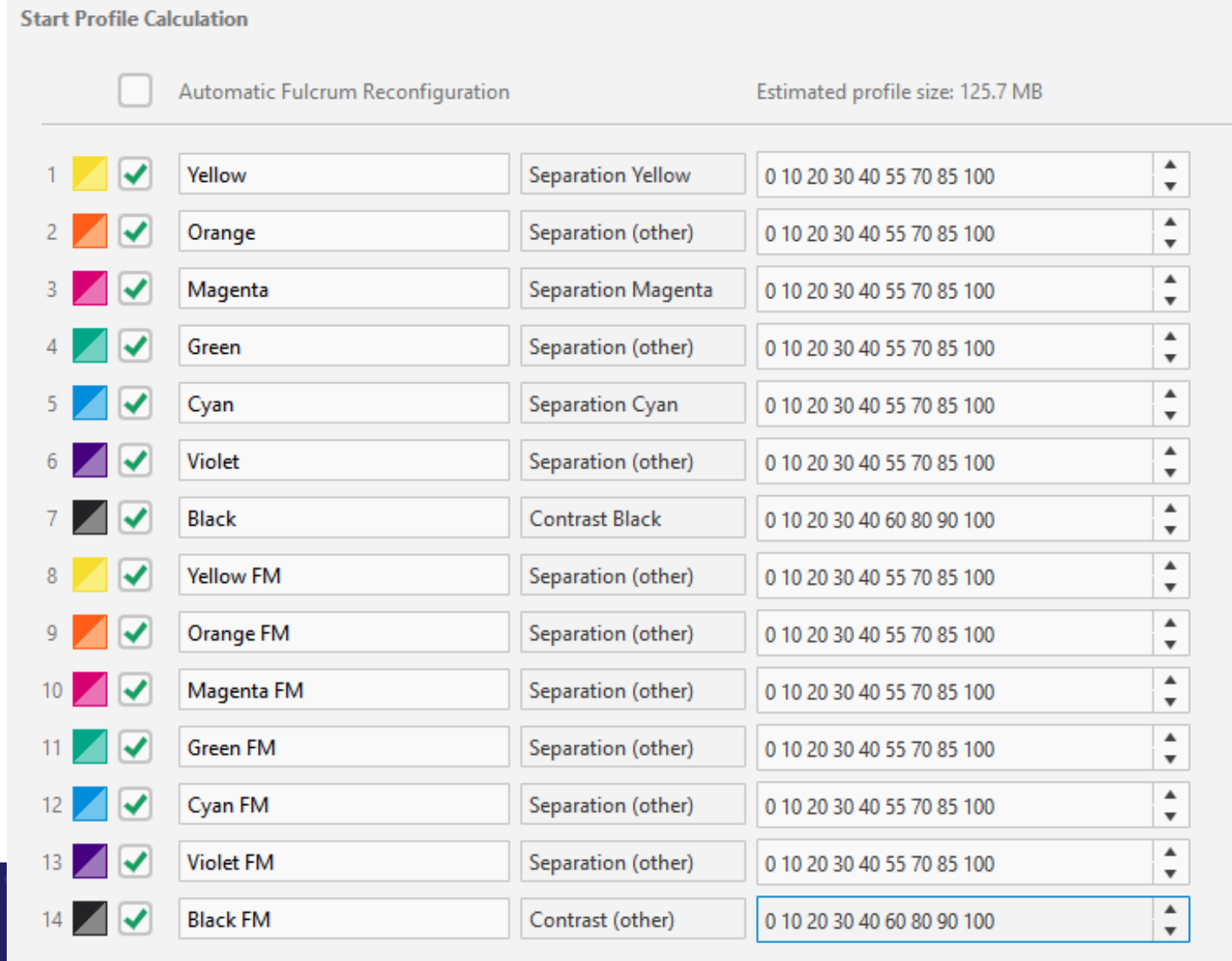


# GMG OpenColor example

- Flexo Dual Screen 7c

- Base: 1001 tables
- Correction: corners ($2^{14}$)
- Profile size **126 MB**

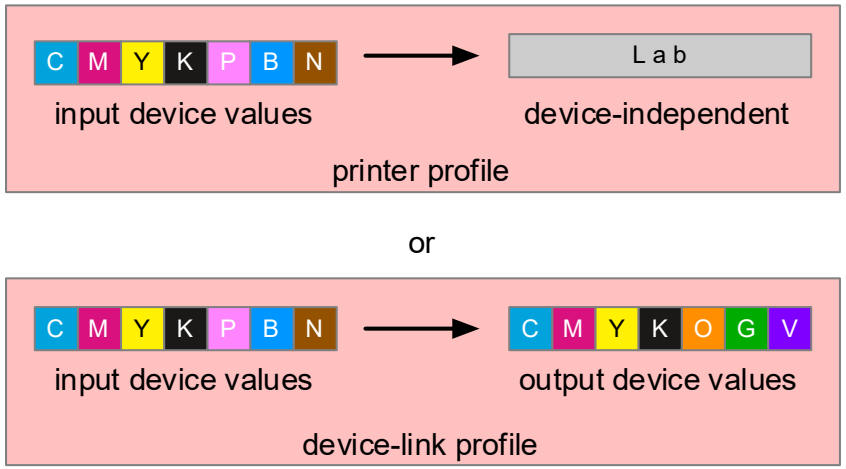
- Full-size profile would be **1 million times** larger



# Comparison with standard profiles

**Standard profiles and CMM**

Single table data + Interpolation math

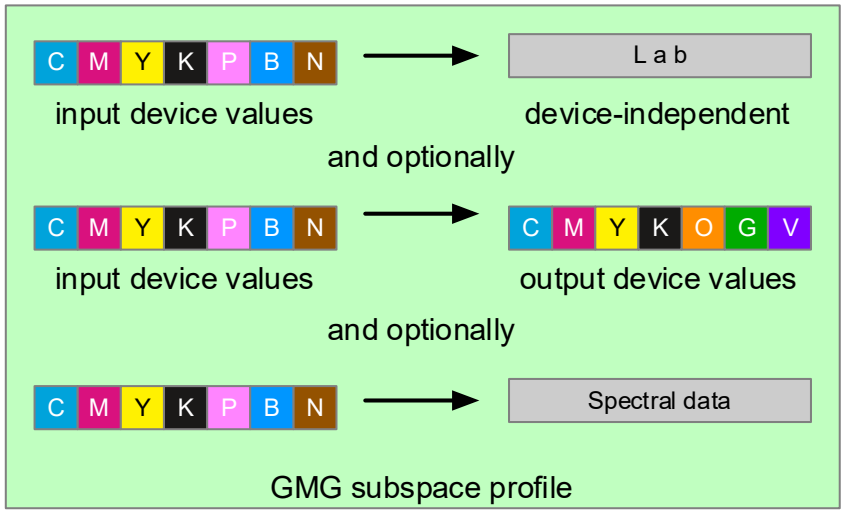



Exponential size increase

causes lower resolution
and must contain complete tables

**New GMG profiles and GMG subspace engine**

Multiple table data Different dimensions + Interpolation math + Color estimation + Color correction




Combinatorial size increase

allows higher resolution
and to select what we need

# Summary

- New GMG profile format and subspace engine
- High-resolution tables for needed subspaces (e. g. 4-color)
- Such conversions are fast
- Accuracy also for higher-dimensional conversions
  by optional correction layers

- Used in GMG OpenColor (currently: proofing)
- SDK available which integrates well into ICC workflows

# Efficient and accurate color conversions for multicolor printing

Hanno Hoffstadt, GMG

Thank you for your attention.

# Estimation strategies: some points

Choosing starting point and step sequence

- Adding an ink is like venturing out into uncharted territory
- Lower distance from safe ground is better

- Multiple strategies are available
- Choose trade-off between speed and smoothness

- Does also work with no overprint tables at all

# Correction example: convert to deltas