Constrained Spectral Uplifting

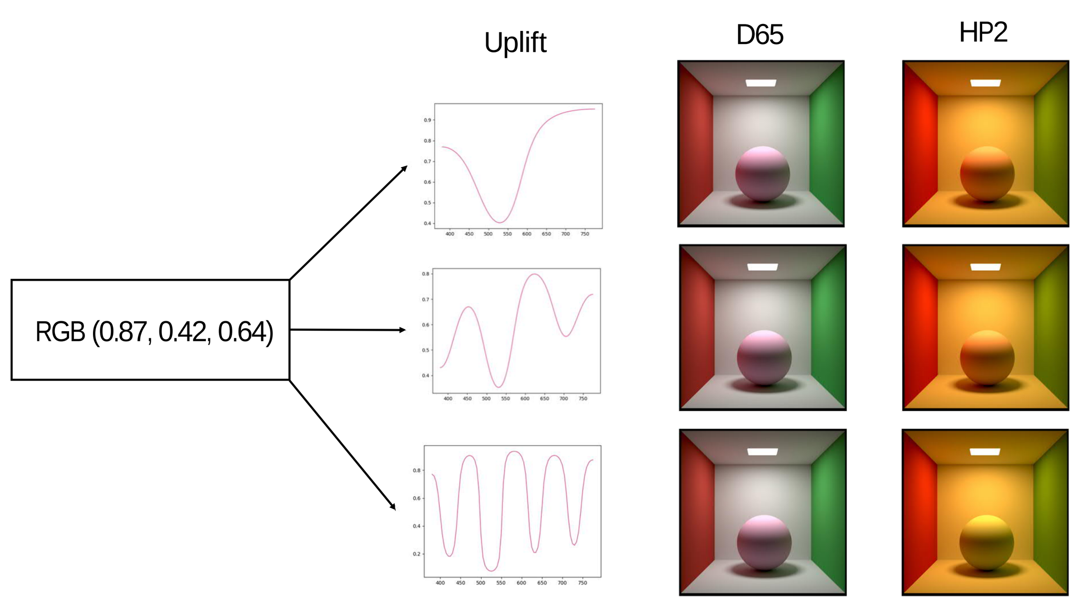
Mgr. Lucia Tódová (supervisor: doc. Alexander Wilkie, Dr.)

Faculty of Mathematics and Physics, Charles University

FACULTY OF MATHEMATICS AND PHYSICS Charles University

Problem

- spectral rendering has been becoming increasingly popular in the VFX and movie industry due to its ability to accurately simulate natural phenomena and predict object appearance under varying illuminating conditions
- a lot of already existing assets (materials, textures) are RGB-based — using them in a spectral renderer requires RGB->spectra conversion (*spectral uplifting*)
- RGB space is a finite subset of the visible gamut
- different uplifting techniques produce different results
  - none of the results may be the same as in nature
  - *metameric artifacts* may be visible under different illuminants

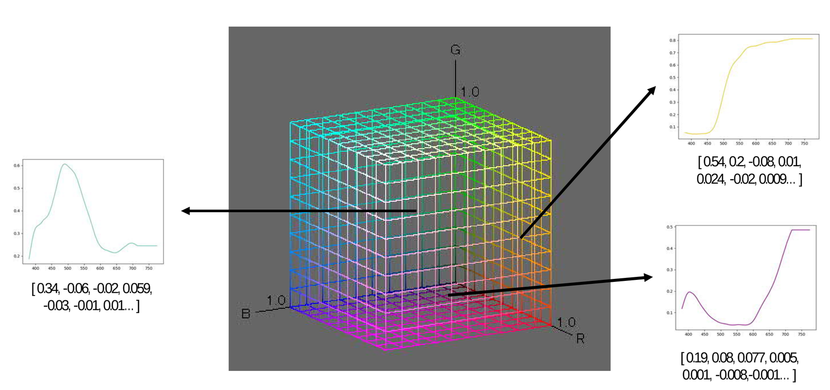


Solution

- *constrain* the uplifting process with a pre-defined RGB->spectra mappings
- preserve these mappings during uplifting

Uplifting system:

- RGB cube divided into evenly-placed voxels
- each lattice point contains a mapping to a spectral curve
  - each curve represented by trigonometric moments [2]
  - variable number of moments used depending on spectral shape complexity

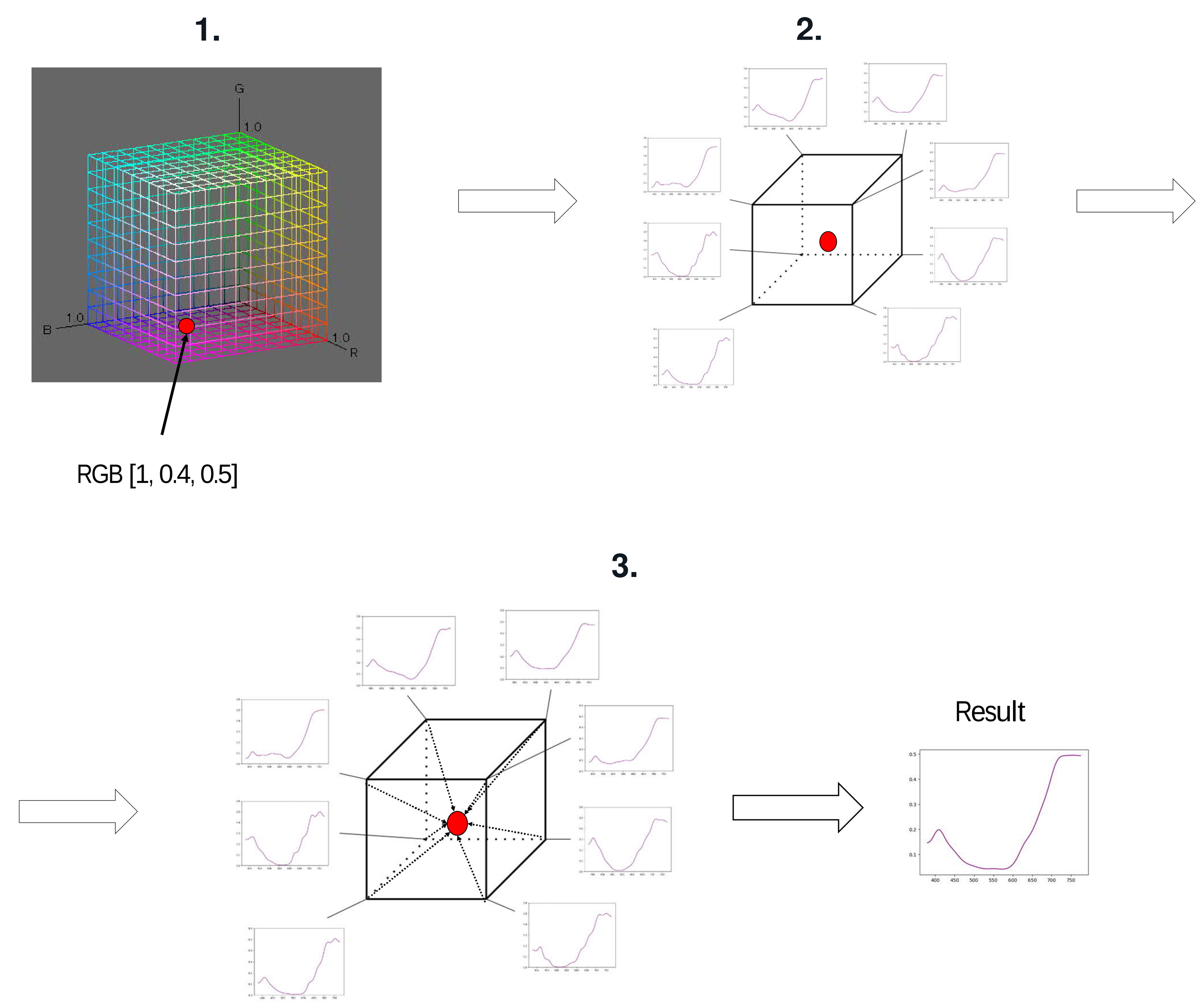


Creating an uplift cube:

1. insert representations of all input spectra to their closest lattice points within the cube
2. use CERES optimizer to perfectly match the lattice points' RGB values
3. to obtain spectra of non-constrained points, utilise neighbors' representations in CERES optimizer — cube iteratively grows from the positions of the constraints

Uplifting an arbitrary RGB

1. Find voxel the RGB value falls into
2. Get spectra of the voxel's 8 lattice points from their moment-based representations
3. Interpolate between spectra

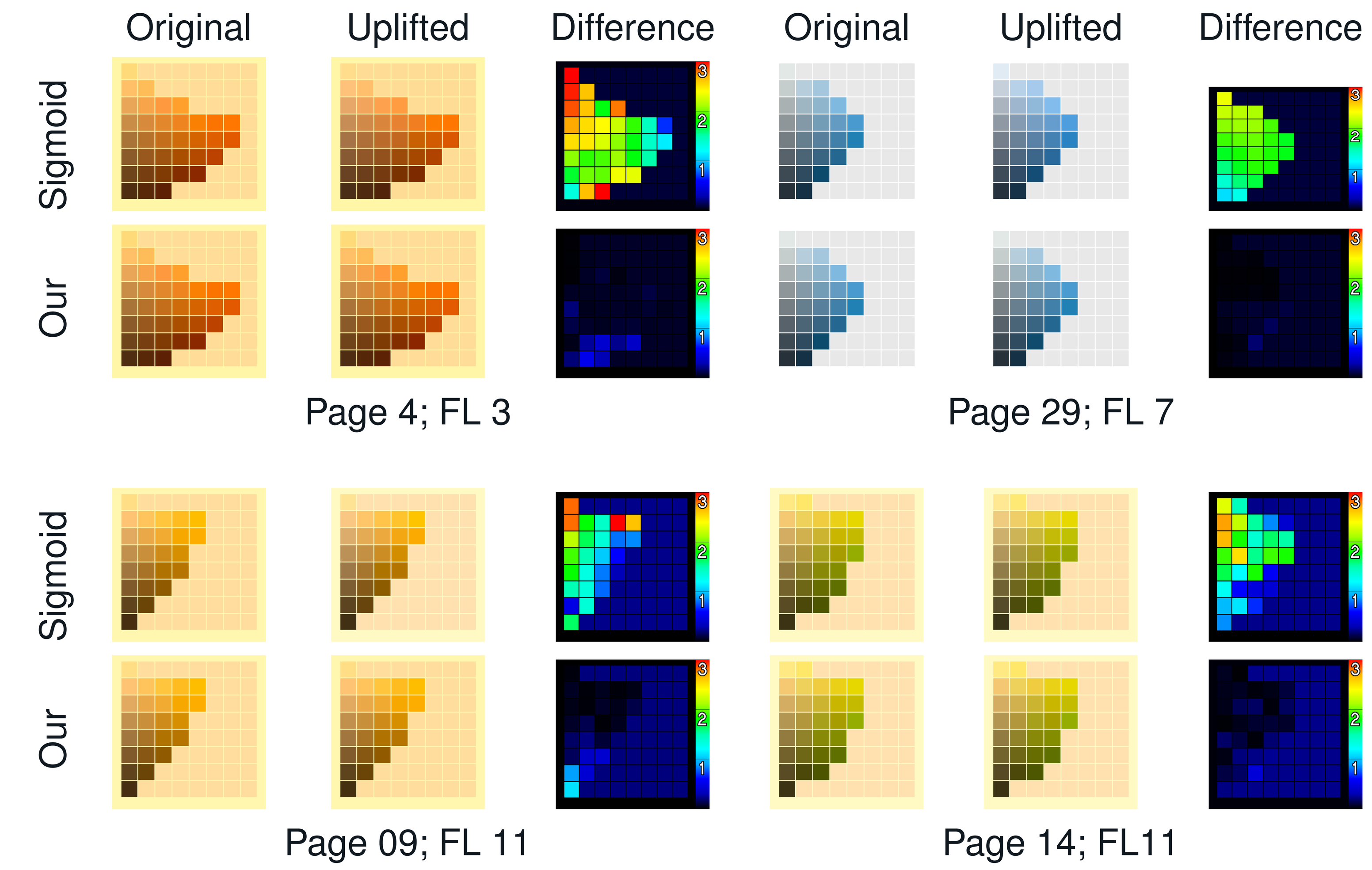


Results — constraint uplifting

- experiments consisted of uplifting RGB values of spectra that have been used as constraints
- testing scenarios included multiple CIE Illuminants and multiple entries from various colour atlases (Pantone, RAL, Macbeth Color Checker. . .)
- error measured by deviance from actual colour under the illuminants
- **Result:**
  - average error Delta E = 0.21 — negligible in terms of human colour perception

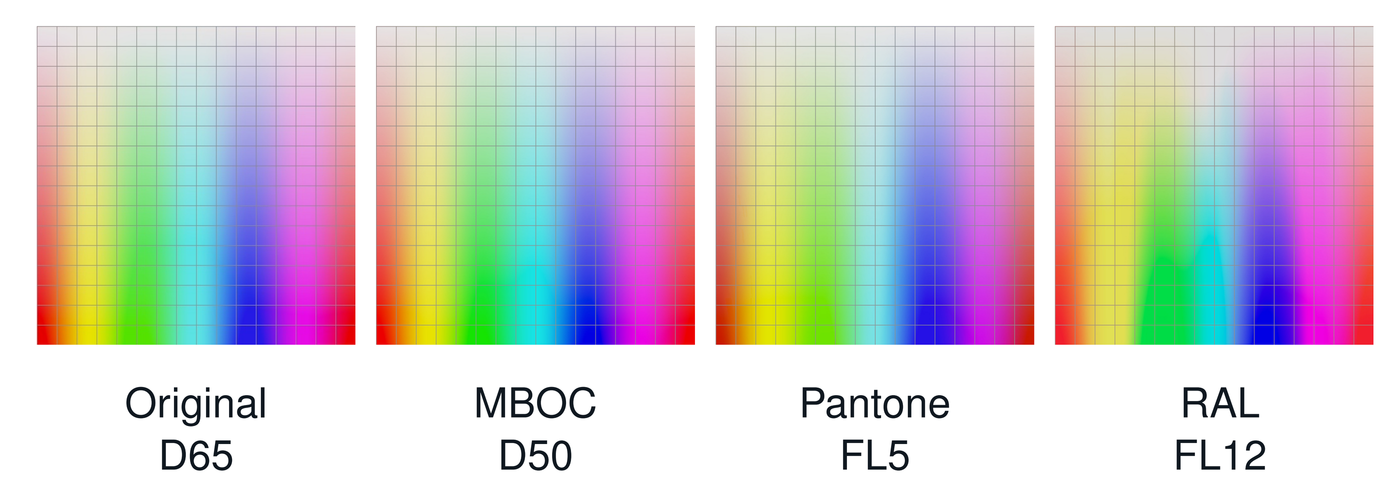
Comparison to sigmoid-based uplift by Jakob and Hanika [1]

- patches of Munsell Book of Colour used as constraints
- RGB values of constraints uplifted with our and sigmoid-based method respectively
- **Result:**
  - significant improvement of the current state of the art



Results — rest of the colour gamut

- uplifting of a highly saturated RGB texture in order to determine the performance of the non-constrained parts of the cube
- **Result**:
  - no significant artifacts
  - smooth transitions between individual metameric families



Future work

- Employ dynamic structure (kD-tree/octree) to improve memory overhead
- Remove sRGB gamut limitation by supporting wider colour gamuts (such as Adobe RGB)

References

[1] Wenzel Jakob and Johannes Hanika. "A Low-Dimensional Function Space for Efficient Spectral Upsampling". In: *Computer Graphics Forum*. Vol. 38. 2. Wiley Online Library. 2019, pp. 147–155.

[2] Christoph Peters et al. "Using moments to represent bounded signals for spectral rendering". In: *ACM Transactions on Graphics (TOG)* 38.4 (2019), pp. 1–14.