(Bi-)Spectral Rendering in Practice

Alexander Wilkie
Charles University in Prague

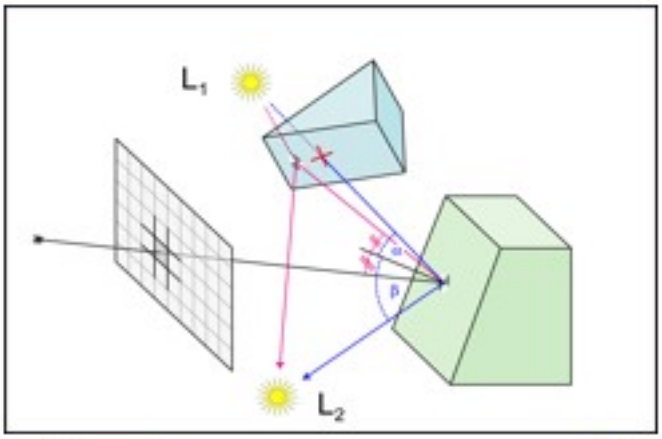
Andrea Weidlich
RTT AG

Overview

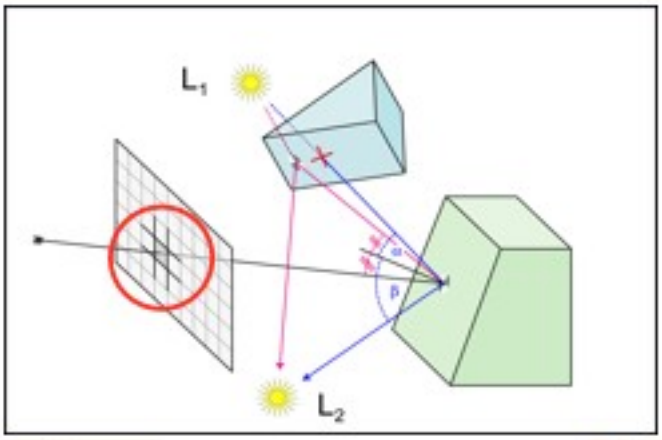
- Introduction
- Physics Background
- Why and Where We Need Spectral Rendering
- Acquiring Spectra and Colour Values
- Options for Describing Spectra
- Practical Aspects of Building and Debugging a Bi-spectral Hybrid Colourspace / Spectral Rendering System

Motivation

What do we compute during rendering?



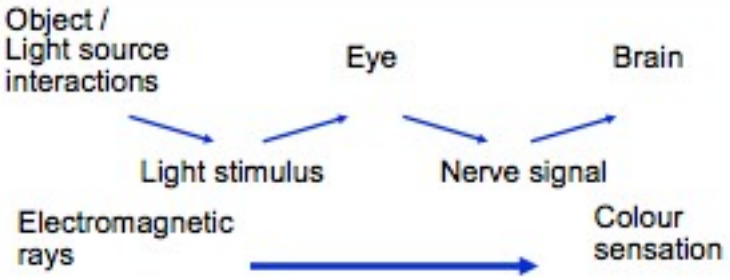
What do we compute during rendering?



What do we compute during rendering?

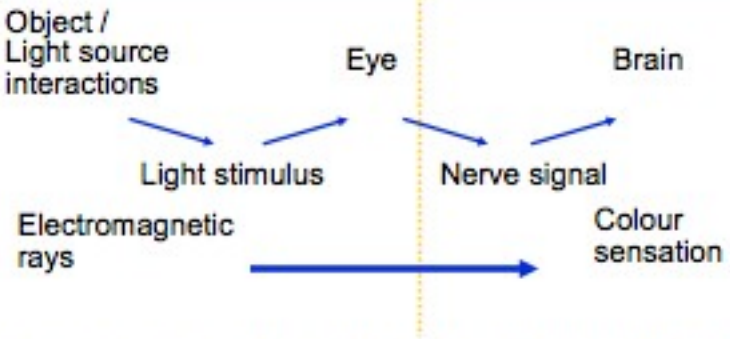
- What is the goal of image synthesis?
  - Computation of (realistic) images
- These are of course intended to be viewed
  - Practically all display devices take colour values as input
- So the desired end result of rendering computations are usually colour values!
  - But this does not necessarily mean that the calculations that lead to these colour values also need to be done using colours!

Colour - A Visual Sensation



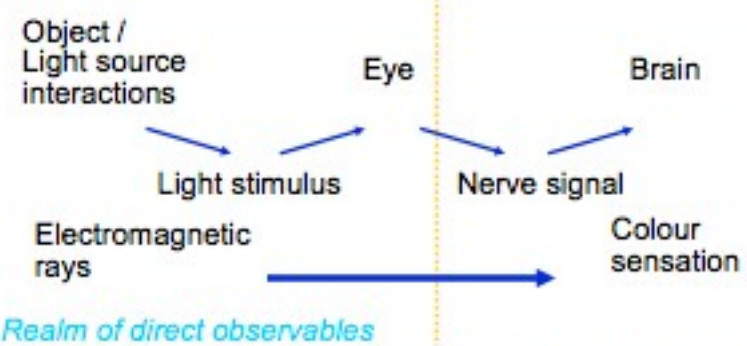
Rendering *computations* take place on the top left, but the display stage deals with colour values

Colour - A Visual Sensation



Rendering *computations* take place on the top left, but the display stage deals with colour values

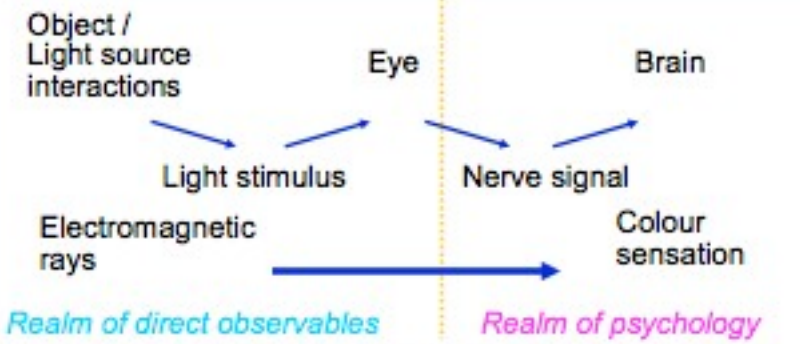
Colour - A Visual Sensation



Realm of direct observables

Rendering *computations* take place on the top left, but the display stage deals with colour values

Colour - A Visual Sensation

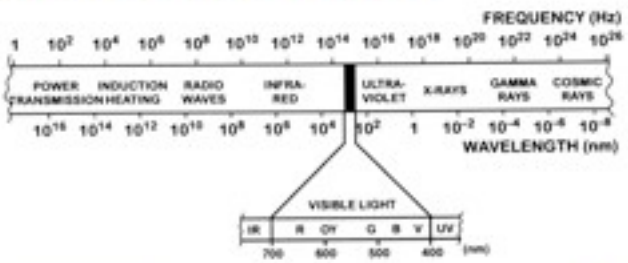


Rendering *computations* take place on the top left, but the display stage deals with colour values

Physics Background

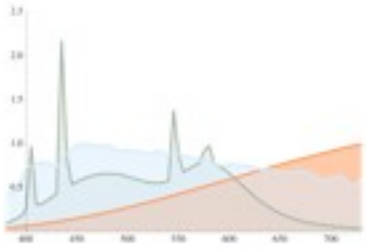
Light – Basic Properties

- Visible light is electromagnetic radiation in a particular region of the entire spectrum
- Distinguishing criterion: its frequency



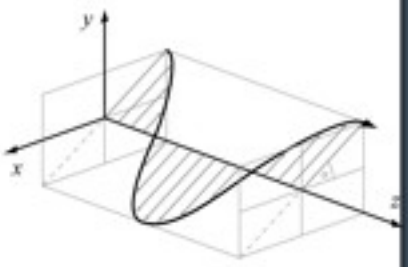
Light – Spectrum

- Visible light is electromagnetic radiation in a particular region of the entire spectrum
- Normally, a ray of light contains many different waves with individual frequencies
- The associated distribution of wavelength intensities per wavelength is referred to as the spectrum of a given ray or lightsource

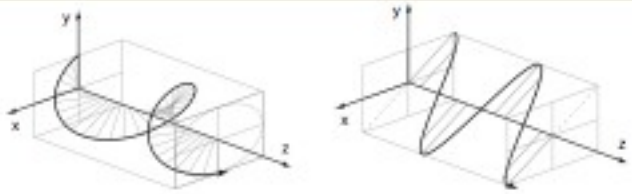


Light - Polarisation

- Light is a transversal wave
- Its frequency alone is sometimes not sufficient to describe a given wave train
- For coherent beams of light, phase information has to be maintained
- This is referred to as the *polarisation state* of a light ray



Waveforms vs. Perception



- Trace of the wave-vector for two photons
  - Individual photons / wave-trains are *always* polarised!
- These two photons have the same frequency
  - We perceive them as having the same colour!
- Only potential difference: interaction with objects

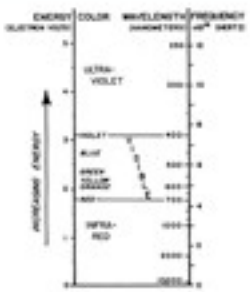
What are we modelling in a renderer?

- Light and reflectance are usually treated alike
- But from a rendering engineering perspective, the following facts are noteworthy:
  - Emission spectra can be spiky (i.e. contain high frequency components)
    - Resolutions as fine as 1nm can be necessary
  - Reflectance spectra are usually smooth
    - 20nm resolution are common
  - Fluorescence requires bi-spectral data
  - Polarisation requires additional data that is orthogonal to per-wavelength sampling

Emission Spectra: Blackbody, Excitation, LED

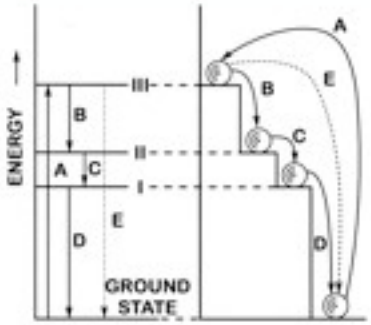
Energy Levels & Visible Range

- A given light frequency corresponds to a fixed wavelength and energy level
- With organic sensors, „seeing" is best done in the visible range of humans:
  - Higher ➔ energy destroys molecules (UV)
  - Lower ➔ focus & transparency problems (IR)

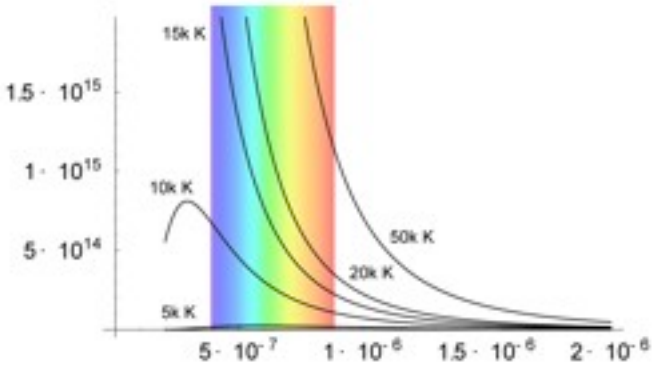


Quantum Properties of Matter

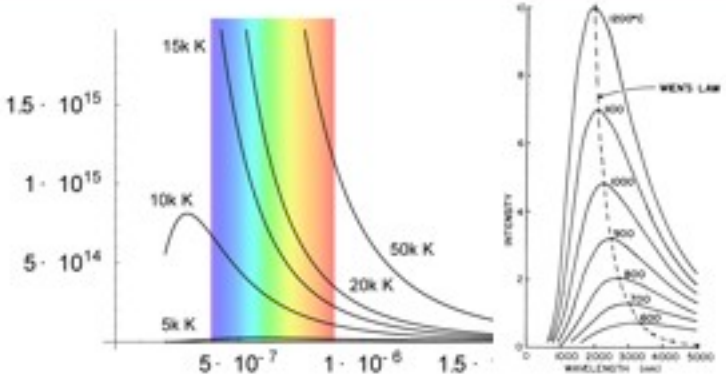
- Quantum systems such as molecules can only exist in certain discrete states
- Only certain transitions between these states are allowed
- The states directly correspond to energy levels, and to frequencies!



Blackbody Radiation #3

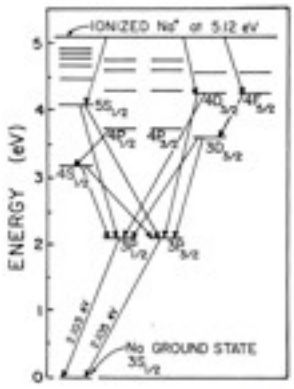


Blackbody Radiation #3

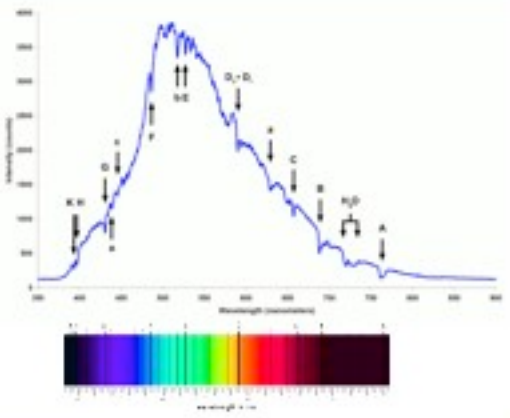


Gas Excitation – Sodium

- Several different excitation paths exist
- The existence of the two „twin" steps at the bottom is responsible for the double emission line spectrum of sodium lamps



Fraunhofer Lines in Solar Spectrum



Fluorescent Tube Spectrum

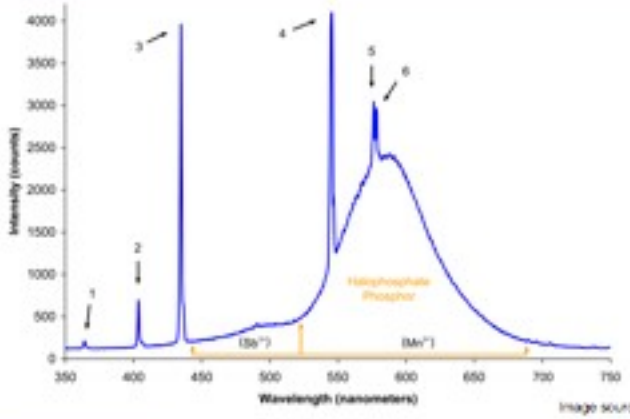


Image source: Wikipedia

Band Gap Emission - LED

- LEDs are semiconductor diodes with a band gap that causes a specific „tunnelling delta" for electrons that cross it

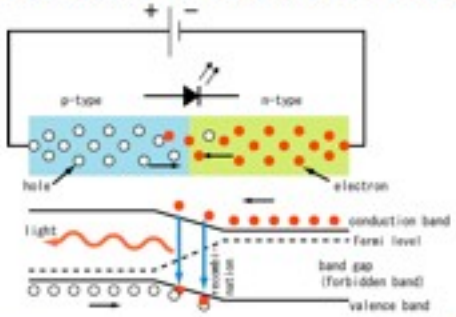


Image source: Wikipedia

White LED Spectrum

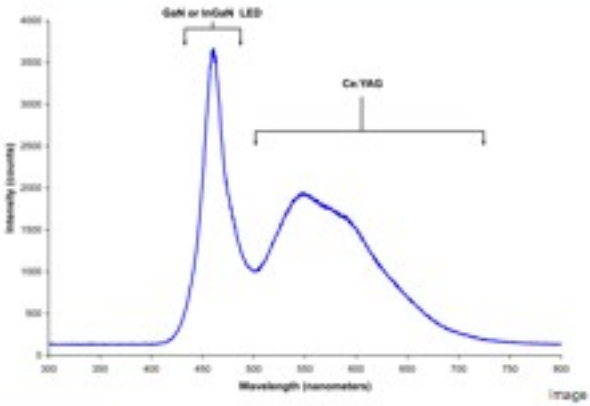
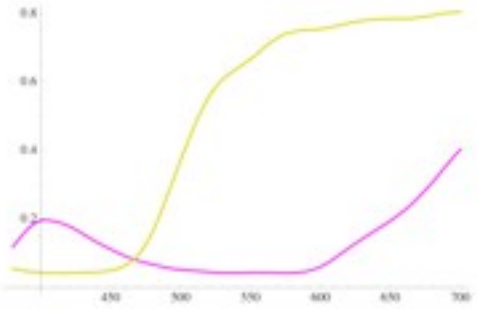


Image source: Wikipedia

Reflectance Spectra: Normal & Bi-Spectral

Organic Molecule Reflectance

- Smooth reflectance spectra
- Easy measurement, easy representation in a rendering system

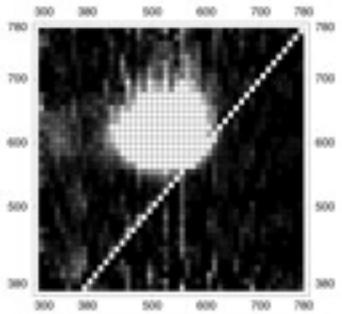


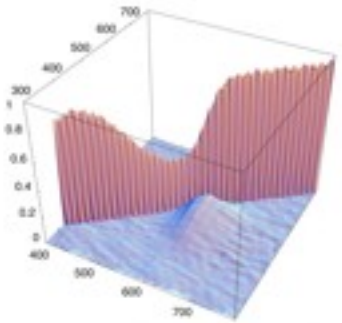
Stokes Shift

- The phenomenon of light being re-emitted at a different frequency than the one at which it was absorbed is called *Stokes Shift*

Image source: Wikipedia

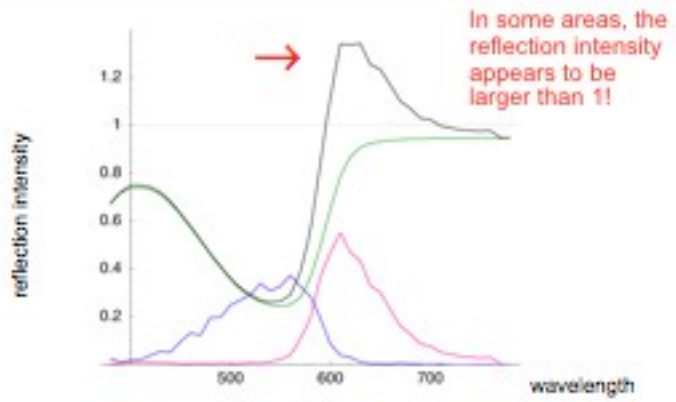
Re-radiation Example





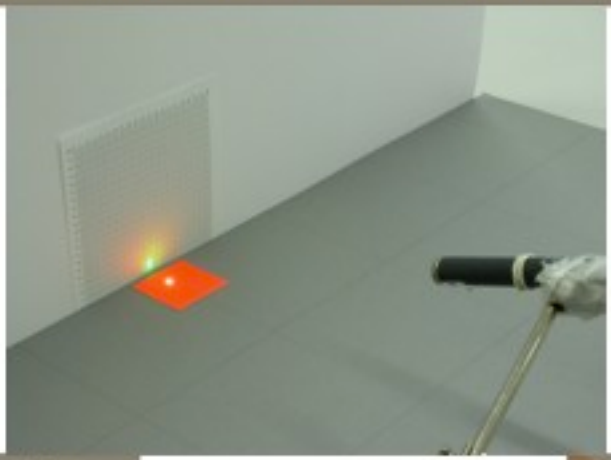
- Most common fluorescent pigments re-radiate at lower levels (here: pink 3M Post-It)

Fluorescence: Energy Transfer



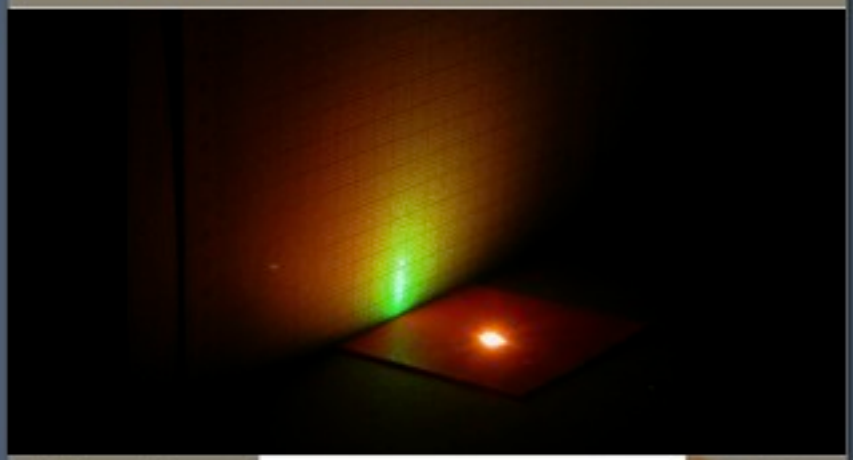
Fluorescence: Real-World Example

Fluorescent Reflection Experiment

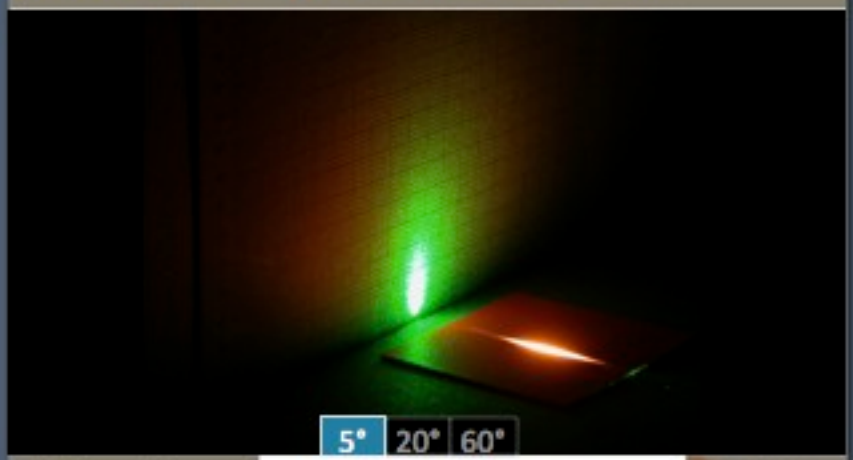


Plain vs. Fluorescent Sample

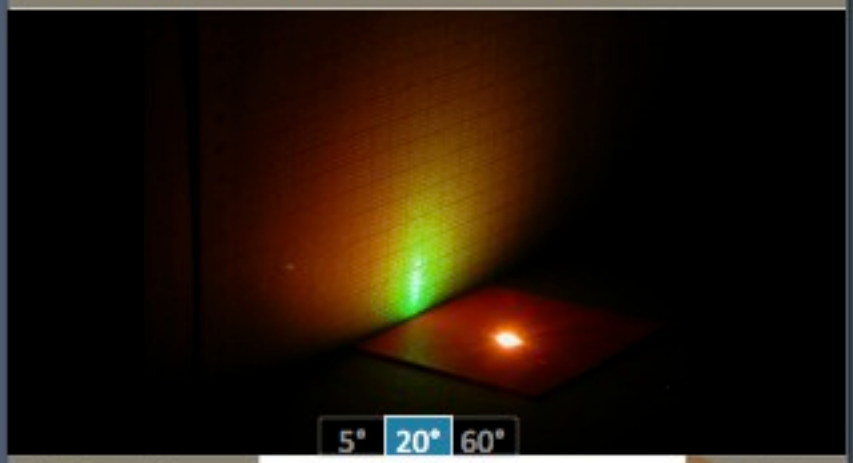
Plain vs. Fluorescent Sample



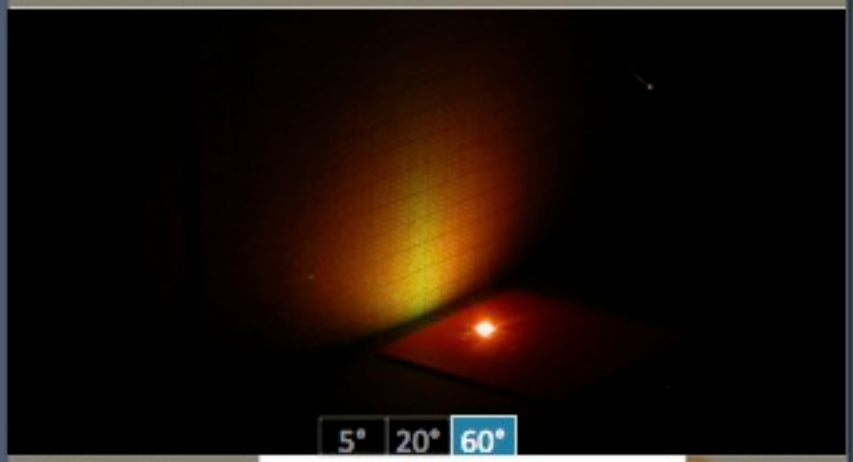
Fluorescent Sample @ 5°, 20°, 60°



Fluorescent Sample @ 5°, 20°, 60°

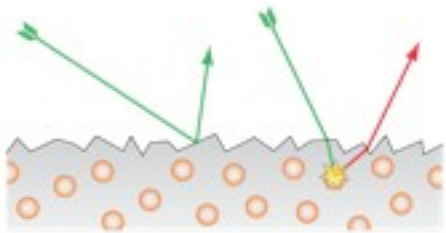


Fluorescent Sample @ 5°, 20°, 60°



Bi-Coloured Reflection Pattern

- Rays which are reflected by the substrate retain their colour
- Rays which interact with the colorant molecules undergo wavelength shift



Why and Where We Need Spectral Rendering

Shading Calculation Quantities

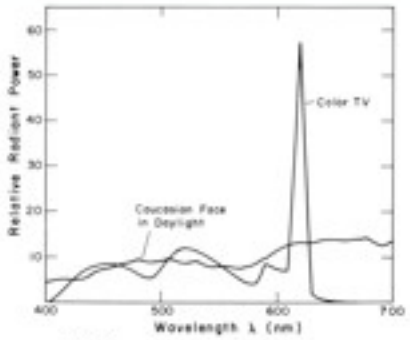
- Two main types of rendering engine exist:
- Conventional colour space based renderers
  - RGB space (majority)
  - CIE XYZ space
- Spectral Renderers
  - Very few products available
  - One example: Maxwell renderer
  - Few details of internal workings known

Why Spectral Rendering?

- Effects that e.g. require spectral representation of light:
  - Metamerism
  - Volume absorption
  - Dispersion (prisms, rainbows)
  - Interference and diffraction
  - Fluorescent materials and light sources
- RGB insufficient to accurately reproduce such effects

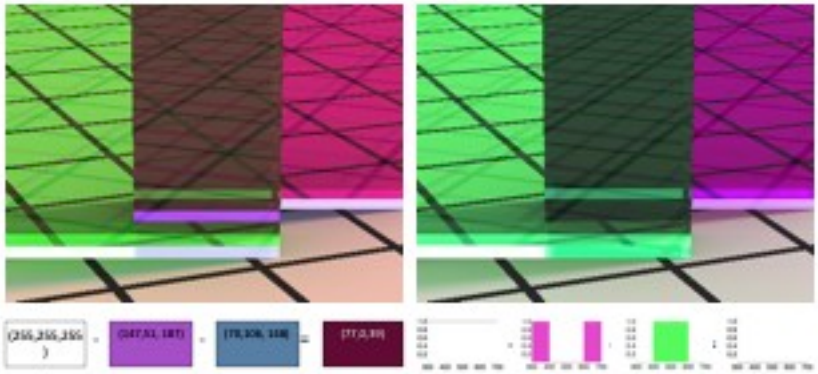
Metamerism

- Occurs frequently
- RGB monitors are a typical example
- One of the most interesting problems in the paint and pigment industry
- Makes prediction of object appearance by a colour space renderer impossible!



RGB vs. Spectral: Metamerism

RGB vs. Spectral: Absorption



Spectral vs. RGB

- RGB Rendering
  - +Fast, widely supported
  - –Limited accuracy (sharp spectra, different illuminants), some effects hard to support
- Spectral Rendering
  - +Accuracy, prediction of nature possible, can handle metamers and various optical effects
  - –High cost, Aliasing, data mixing, input data, more complicated to write

RGB (Tristimulus) Rendering

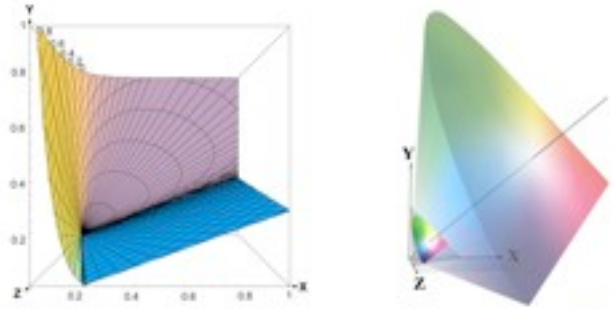
- Three wavelengths (corresponding to red, green, blue) define light source and material properties
- Process 3 samples separately throughout rendering calculation
- Device (or at least RGB space) dependent
  - –Larger RGB spaces better than smaller ones!
  - –Pre-filtering desirable
- RGB representation not ideal

CIE XYZ

- A tri-stimulus colour system derived from RGB and based on imaginary primaries referred to as) 1931
- All three are outside the human visual gamut
- Hence only positive XYZ values can occur
- Valid colours a subspace of the first octant - XYZ not closed under multiplication!

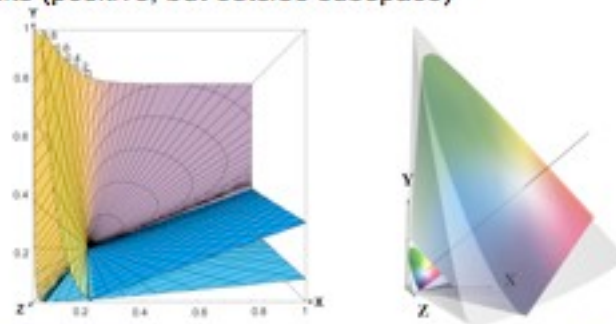
Usefulness of XYZ in 3D Graphics

- Valid colours a subspace of the first octant
- XYZ not closed under multiplication!



XYZ: Problem of Multiplication

- Component-wise XYZ multiplications can yield invalid results (positive, but outside subspace)



CIE XYZ vs. RGB

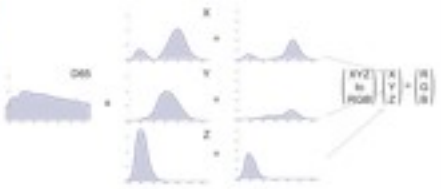
- Problem does not exist for RGB spaces:
  - They occupy the entire first octant, are closed under multiplication
- RGB cannot represent all visible colours
  - But XYZ is no alternative due to the discussed problems!
  - Older literature still recommends using XYZ!
- Correct choice of RGB space can make a significant difference!

Accurate RGB Rendering

- Observation of Greg Ward e.a. [Ward02]:
  - RGB rendering can be made fairly accurate!
- Key improvements:
  - Using large RGB spaces
  - Pre-filtering of reflectance for used illuminants
  - Direct lighting accurate, indirect light not!

Spectral Rendering: Steps

- Get spectrum
- Prepare spectrum
- Process spectral samples separately throughout rendering calculation
- Convert spectrum to final display colour using CIE color matching functions and standard transformations



Acquiring Spectra and Colour Values

Measurements

- In this part of the course, we only talk about individual spectral measurements
  - Result: single emission or reflectance spectrum
- Measurements of entire BRDFs are an orthogonal problem
  - Conventional gonio-reflectometers need a spectro-radiometer as one of their components
- BTF measurements would ideally also capture the reflection spectrum for each pixel of the surface that is being measured
  - Active research area

Radiometry vs. Photometry

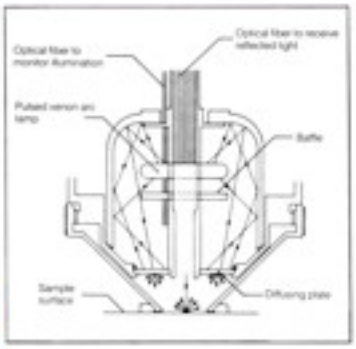
- **Radiometry** is the science and technology of the measurement of electromagnetic radiation over the whole spectrum of electromagnetic waves
- **Spectroradiometry** concerns itself with the spectral composition of electromagnetic radiation
- **Photometry** and spectrophotometry are the same in terms of the visible spectrum, and the average human observer

Side note: Chroma Meters

*(increasingly common devices, but they are **not** what we need for rendering)*

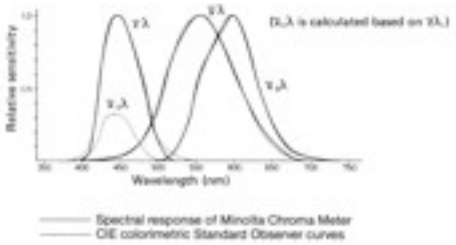
Chroma Meters 1

- Principle: a diffusor spreads light over three photocells which are covered by CIE XYZ-related filters
- Available in self-illuminating and passive versions
- Only records colour values which are specific to a given illuminant
- → *Not useful for rendering purposes, except under very special circumstances!*



Chroma Meters 2

- Cannot be used to predict appearance
- Still very useful for industrial quality control
- Good at finding colour discrepancies in production environments
- Much simpler and cheaper than spectrophotometers



Minolta ChromaMeter

- Typical device for industrial use
- Principle unchanged over past decades
- New devices are just more user-friendly
- Separate data processor and measurement head
- Head can be used separately and/or connected to a computer

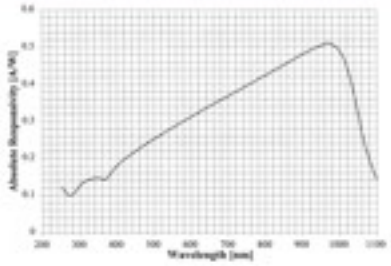
Spectro-Radiometers

Spectro-Radiometer Design

- Two (three) key components are needed:
  - A reliable broadband sensor for light with a known spectral response curve
  - A reliably controllable monochromator to scan the entire visible spectrum
  - (only for fluorescent reflectance measurements: tuneable monochrome lightsource)
- Additionally, provisions for repeatable and controlled specimen mounting and illumination have to be made

Silicone Photodiodes

- The light-sensitive elements in most radiometric and photometric instruments are silicon photo-diodes

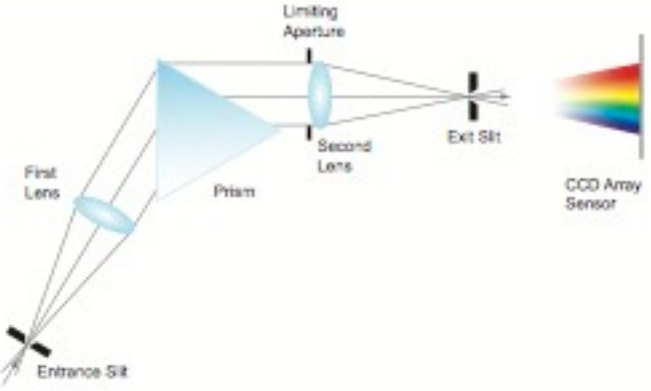


- These devices yield a wavelength-dependent current per radiant energy that hits them
- Used as absolute detector for monochrome l.

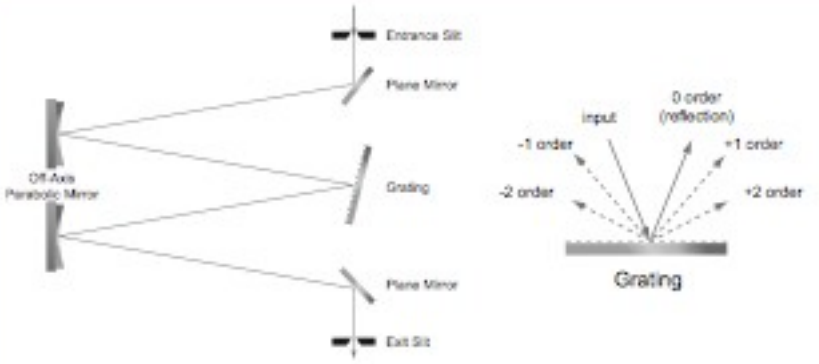
Monochromators

- Main sources of monochrome light and/or wavelength selection:
  - Filters – inflexible (one per sample, no intermediaries), plus usually too wide-band
    → Tuneable Lyot filters an exception!
  - Lasers – inflexible (same as above), expensive
  - Prisms – nonlinear wavelength dependency, but pure output
  - Gratings – linear dependency on wavelength, but higher-order diffraction effects affect purity of result

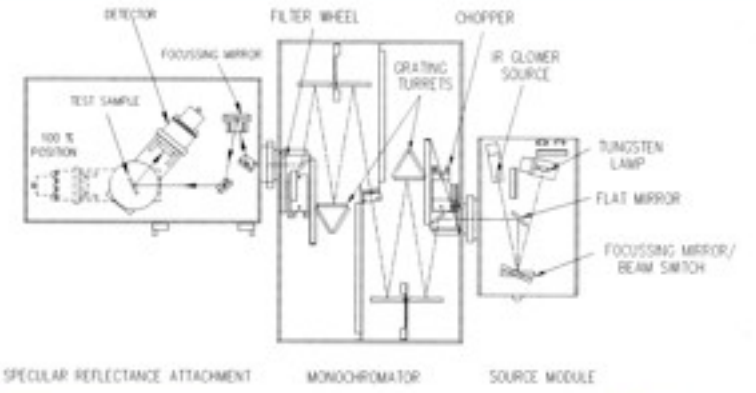
Prism Monochromator



Grating Type Monochromator



Spectro-Radiometer Components

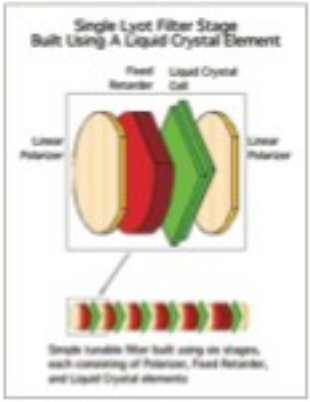


Bi-Spectral Capture

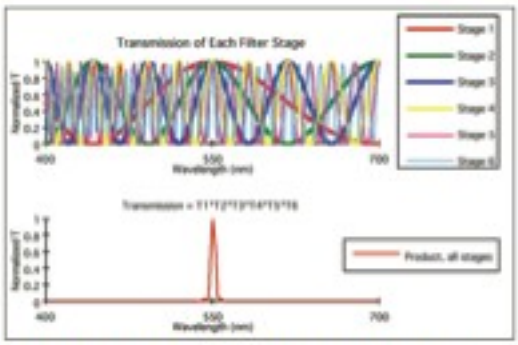
- Measurement of fluorescent samples requires a bi-spectral measurement setup
  - Tuneable, narrow-band monochrome lightsource that is used to illuminate the sample
  - Spectro-radiometer to measure each response
- Problem: high-power narrow-band monochrome light source
  - Narrow band implicitly means only a small percentage of lamp efficiency is used
  - Ideally, should include near UV range
  - Sensitive spectro-radiometer needed

(Tunable) Lyot Filters

- Basic idea: use a cascade of birefringent materials to achieve a narrow bandpass
  - Lyot filter
- Tunable version: insert materials of controllable birefringence into the optical path



Lyot Filter Cascade



- Each Lyot element is a bandpass of certain width
- A cascade results in a single transmission band

Lyot filter example: CRi VariSpec

- 55ms switching time
- No vibrations
- Can be used in front of digicams etc.
- Different models for various wavelength ranges
  - No one-size-fits-all filter possible

Professional Spectro-Radiometer 1

CM-3720d

- Internal arrangements basically as shown in previous sketch
- 10nm resolution from 380 to 780nm, 18 kg
- Two gratings, 15cm sphere
- Extremely high inter-device agreement
- Capable of eliminating fluorescence effects

Professional Spectro-Radiometer 2

- Internal arrangements differs from sketch - direct light path to grating
- Intended for light measurements
- 0.9 nm resolution from 380 to 780nm, 5 kg
- 1 degree measurement angle
- Various lenses (wide angle, tele etc.) available

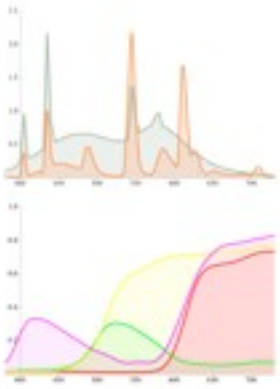
Gretag Macbeth Spectrolino

- Handheld device
- 10nm resolution, 380 to 780nm
- Has to be attached to a computer via serial port
- Not particularly useful without the accompanying software
- Serial interface fully documented
- Differences to previous device: less repeatability, less inter-device agreement
- Modern version: EyeOne (USB)

Options for Describing Spectra

Spectral Representations

- How do we store the data for rendering?
- Represent light as frequency distribution
  - Usually smooth
  - Sometimes sharp peaks (fluorescent lightsources, spectral colours)

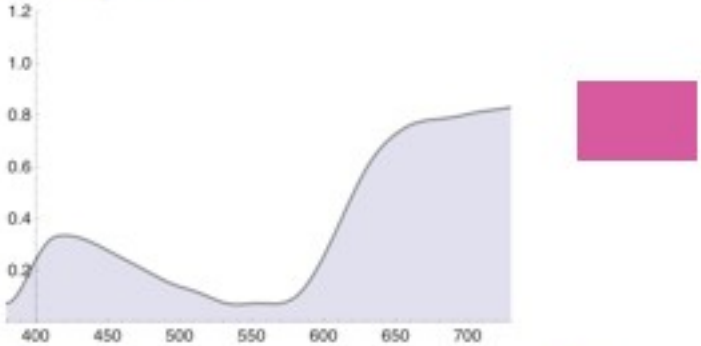


How to discritize these distributions?

- Regularly sampled spectra
  - Aliasing
  - Fast Convolution
- Linear or higher order representations
  - Efficient storage
  - Slow convolution
- Hybrids
  - Slow, but even more efficient w/r to storage

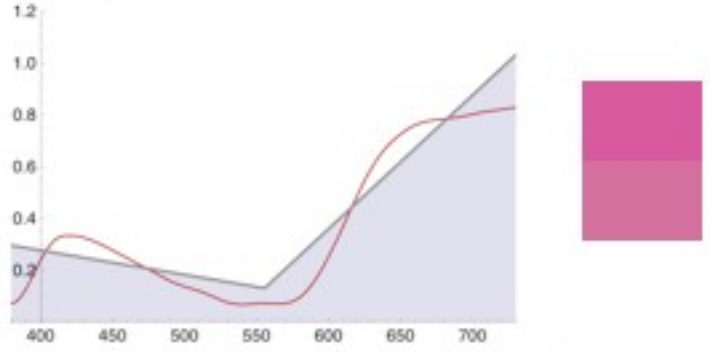
How Many Samples Do We Need?

- Depends on spectrum



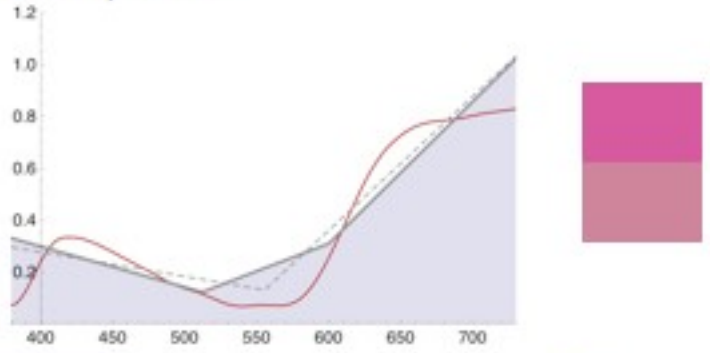
How Many Samples Do We Need?

- Depends on spectrum



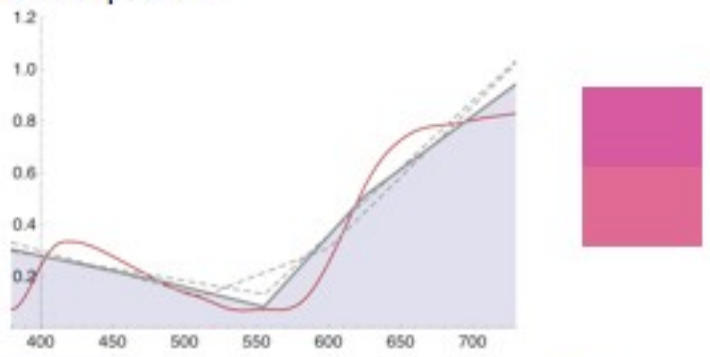
How Many Samples Do We Need?

- Depends on spectrum



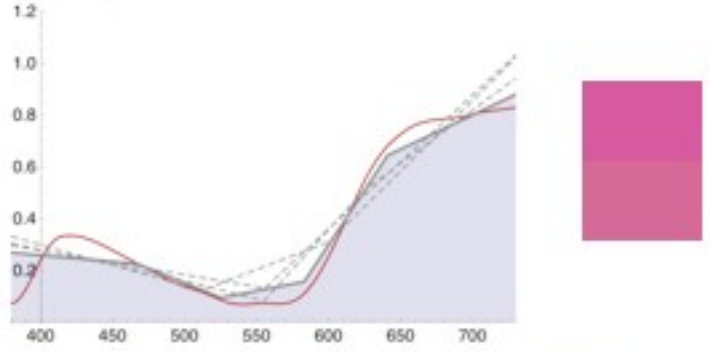
How Many Samples Do We Need?

- Depends on spectrum



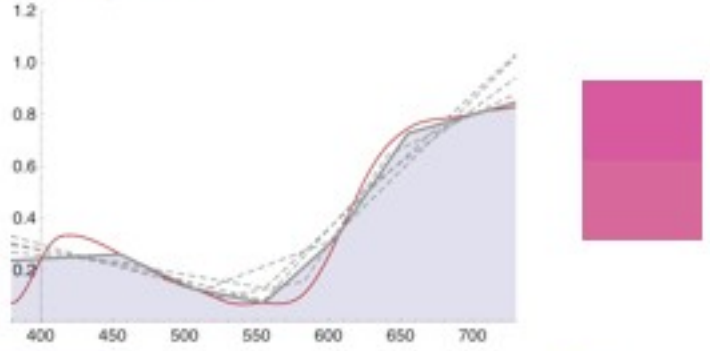
How Many Samples Do We Need?

- Depends on spectrum



How Many Samples Do We Need?

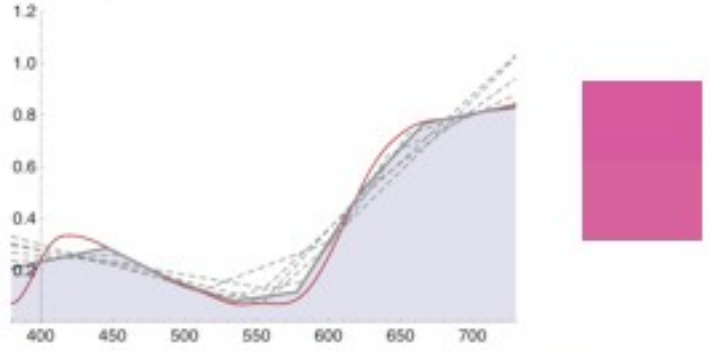
- Depends on spectrum



Eurographics
A. Weidlich and A. Wilkie

How Many Samples Do We Need?

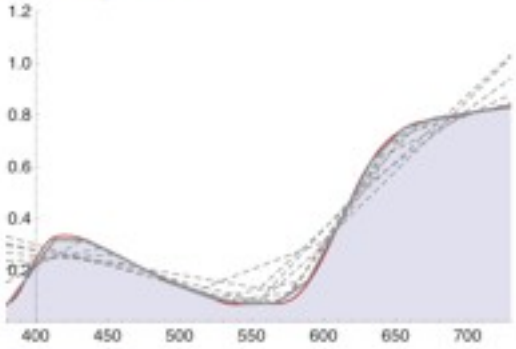
- Depends on spectrum



Eurographics
A. Weidlich and A. Wilkie

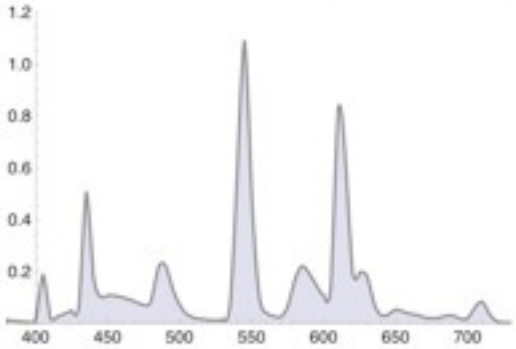
How Many Samples Do We Need?

- Depends on spectrum



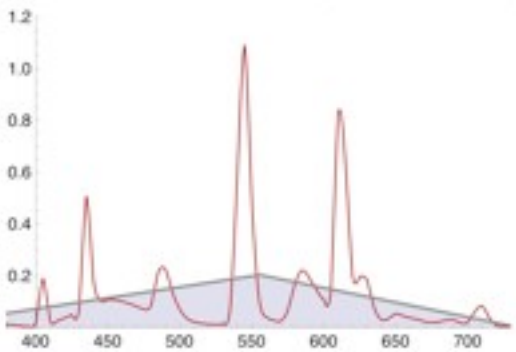
Fluorescent Spectra (CIE F11)

- More samples are needed for spectra with sharp peaks



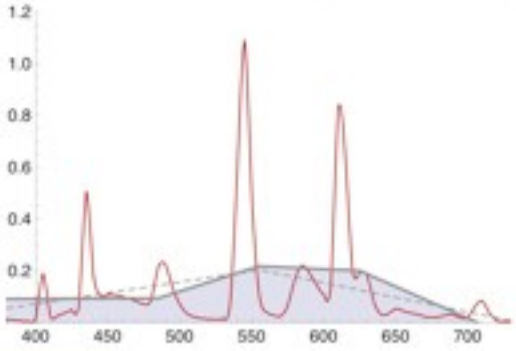
Fluorescent Spectra (CIE F11)

- More samples are needed for spectra with sharp peaks



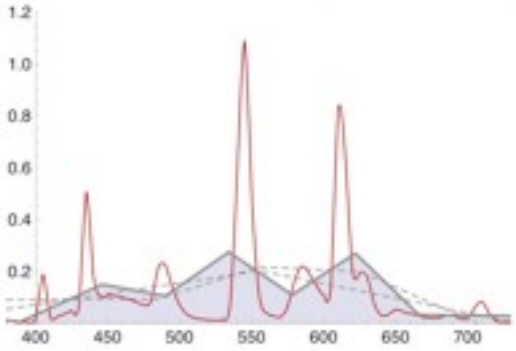
Fluorescent Spectra (CIE F11)

- More samples are needed for spectra with sharp peaks



Fluorescent Spectra (CIE F11)

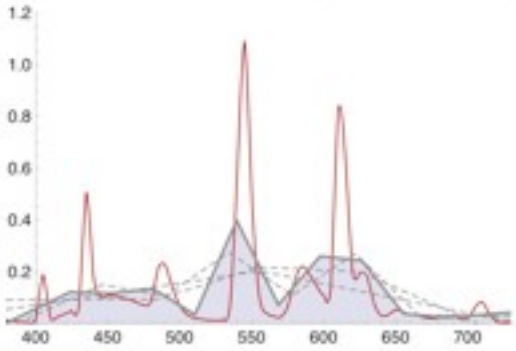
- More samples are needed for spectra with sharp peaks



Eurographics
A. Weidlich and A. Wilkie

Fluorescent Spectra (CIE F11)

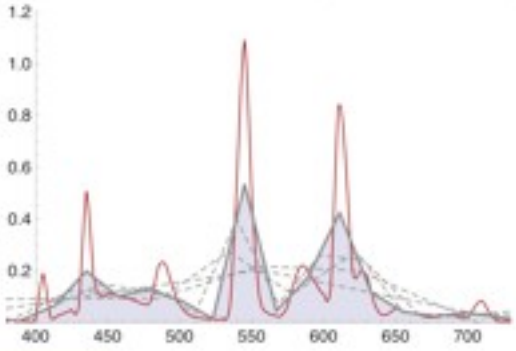
- More samples are needed for spectra with sharp peaks



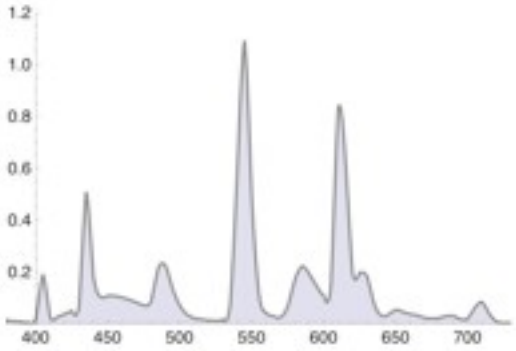
Eurographics
A. Weidlich and A. Wilkie

Fluorescent Spectra (CIE F11)

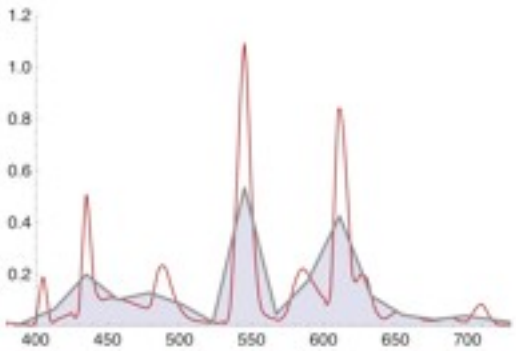
- More samples are needed for spectra with sharp peaks



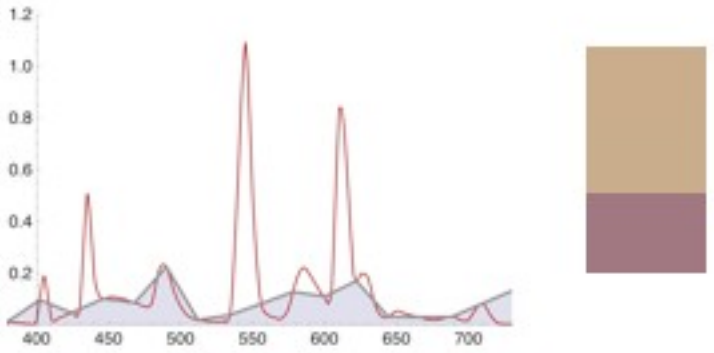
CIE F11 - Representations



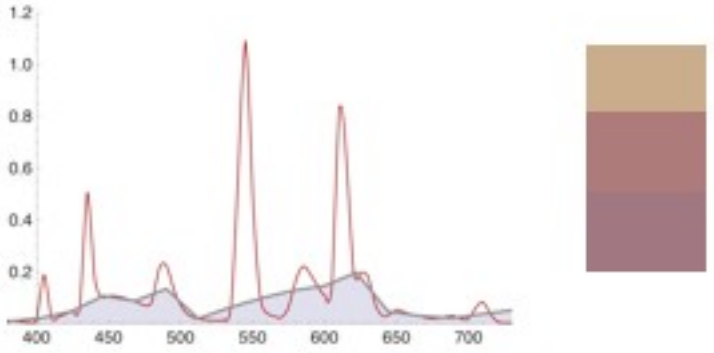
CIE F11 - Representations



CIE F11 - Representations

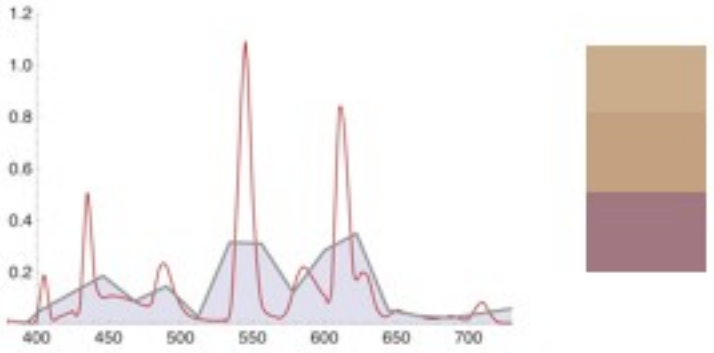


CIE F11 - Representations



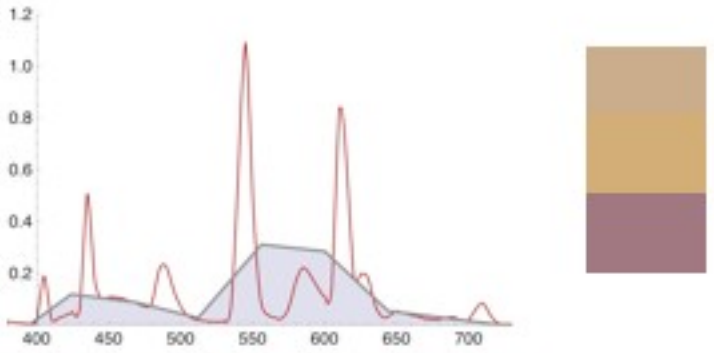
Eurographics
A. Weidlich and A. Wilkie

CIE F11 - Representations

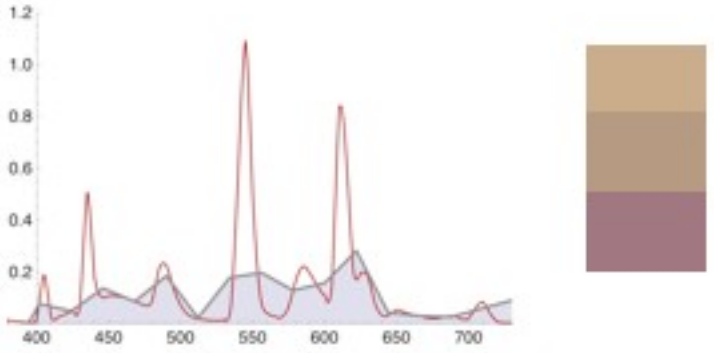


Eurographics
A. Weidlich and A. Wilkie

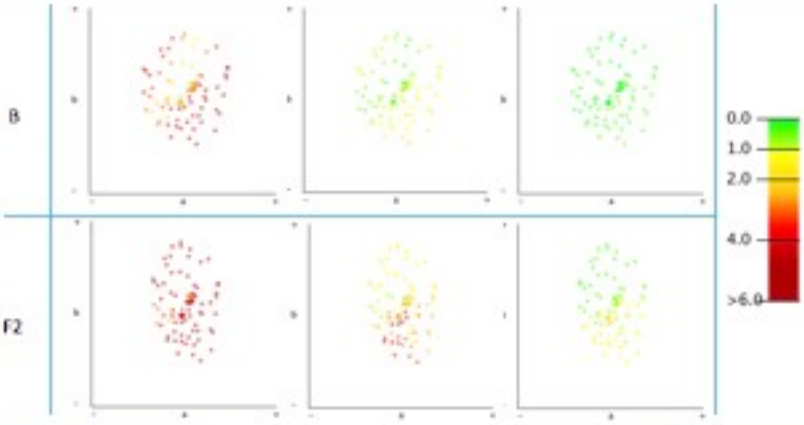
CIE F11 - Representations



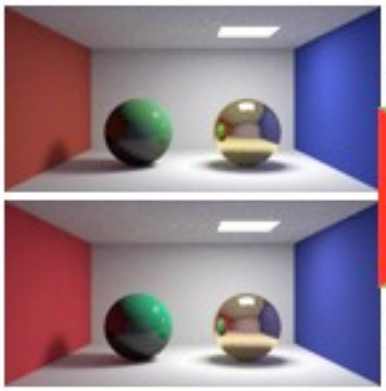
CIE F11 - Representations



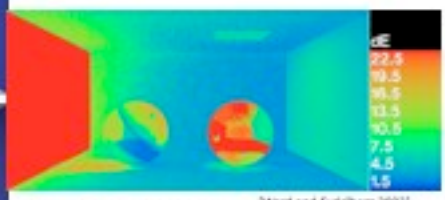
Error Map Sampling Points



Comparison Spectral/Tristimulus



Full spectral rendering (Fluorescent source)



[Ward and Eydelberg 2002]

Naïve tristimulus rendering (CIE *XYZ*)

Practical Aspects of Building and Debugging a Bi-spectral Hybrid Colourspace / Spectral Rendering System

Light vs. Attenuation

- Two types of energy-related entities exist in a rendering system:
  - **Radiant intensity**, e.g. emissions from lightsources („light")
  - **Attenuation**, i.e. the influence of surfaces and media („importance", or „filter")
- In most rendering systems they are not properly distinguished, and use the same data structures (i.e. RGB values)
- Both polarisation and fluorescence break this

Practical Example: ART

- System that is designed to avoid run-time polymorphism (performance!)
- System can be set to one particular colour computation type, and performs all calculations that involve light and reflectance using this type
  - All input data is converted to this representation!
  - Efficient multiplications!
- ART uses a single global state variable called `art_gv` to store, amongst other things, information on which CCT is being used

ART: Colour and Light Subsystem #1

- `art_gv` contains an array of function pointers - one for each function that manipulates light and attenuation
- Advantages of not using polymorphism, but a fixed set of pointers:
  - No lookups necessary - one uniform type used
  - No conversions of representations at runtime
- The contents of the array are switched to one particular set of functions once a representation is chosen

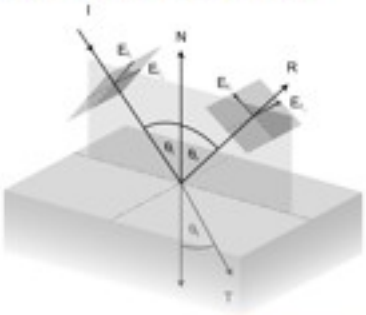
ART: Colour and Light Subsystem #2

- Scene graphs etc. contain "native" spectra
  - Each of those is converted to a CCT representation on loading
- Potential issue with such a switching system:
  - Later changes in CCT
  - Possible sequence: scene graph is loaded, contains info on which CCT is to be used (!)
    → scene graph was possibly built in memory based on some other CCT
    - Solution: scene graph functionality to refresh values that evaluate to a CCT

Side Note:
Describing Polarisation

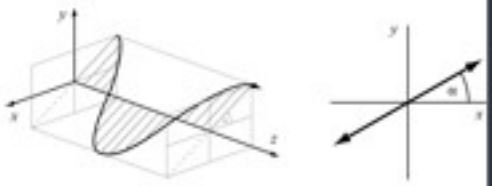
Polarisation: Causes

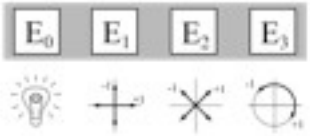
- Specular surfaces: degree of polarisation after reflection can be determined from Fresnel formulae (incl. retardance)
- Used in conjunction with Torrance-Sparrow model
- More complex models are uncommon [He ea.]



Describing Polarisation #2a

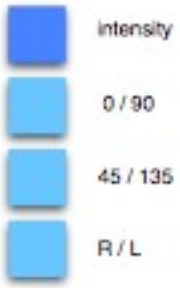
- Light:
  Stokes vectors
  - 4-vector with real elements
  - Range of first component is 0 to infinity, others -1 to 1
  - Blends in with non-polarising implementations





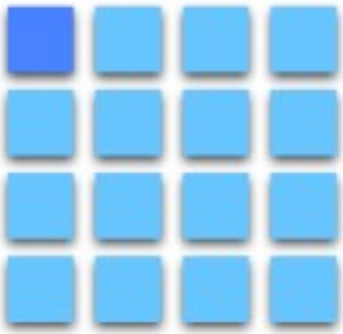
Describing Polarisation #2a

- Light:
  Stokes vectors
  - 4-vector with real elements
  - Range of first component is 0 to infinity, others -1 to 1
  - Blends in with non-polarising implementations



Describing Polarisation #2b

- Filter:
  Müller matrices
- 4 x 4 real matrices
- Element (0,0) is non-polarising filter
- Interaction of all components with each other is encoded



Combined Polarisation and Fluo Rendering

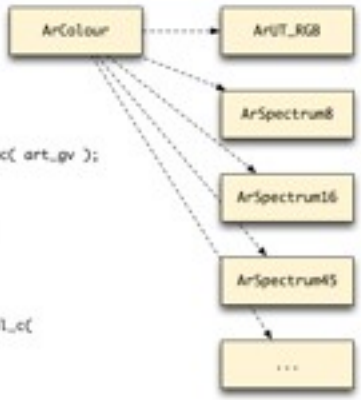
- Fluorescence is a diffuse phenomenon, hence cannot cause polarisation
  - Direct consequence: only the main diagonal of re-radiation matrix are Müller matrices
- All other entries below the main diagonal are just normal floats

Practical Example: ART

- Two levels of colour & spectrum-related data types co-exist
- `ArColour`: basic representation, either spectral or RGB colour values (*not* CIE XYZ!)
  - RGB space can be chosen!
- `ArLightIntensity`: synonymous with **`ArColour`** (via a typedef)
- `ArLight` / `ArAttenuation`: used by rendering algorithms, contain **`ArColour`**
  - Polarisation capabilities are switchable!

ART: `ArColour`

```
typedef struct ArColour
{
    void  * value;
}
ArColour;


ArColour  * colourValue = col_alloc( art_gv );


void col_d_mul_c(
        const ART_GV    * art_gv,
        const double      d0,
              ArColour  * cr
        )
{
    art_gv->arcolour_gv->_acf_d_mul_c(
        art_gv,
        d0,
        cr->value
        );
}
```



ART: `ArLight`

```
typedef struct ArLight
{
    void  * value;
}
ArLight;
```


```
typedef struct ArSVLight
{
    unsigned int        polarised;
    ArStokesVector    * stokesVector;
    ArReferenceFrame    referenceFrame;
}
ArSVLight;
```

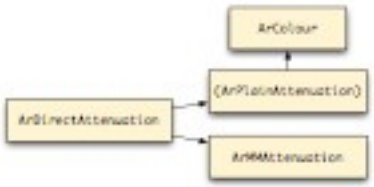
```
typedef struct ArStokesVector
{
    ArLightIntensity  * li[4];
}
ArStokesVector;


typedef struct ArReferenceFrame
{
    Vec3D  c[2];
}
ArReferenceFrame;
```

ART: `ArAttenuation`

```
typedef struct ArAttenuation
{
    ArDirectAttenuation  * attenuation;
    ArCrosstalk          * crosstalk;
    unsigned int           fluorescent;
    unsigned int           monochrome;
    double                 wavelength;
    unsigned int           monochromeChannelIndex[2];
    double                 monochromeChannelWeight[2];
}
ArFilter;
```

ArAttenuation – entire combined attenuation
ArDirectAttenuation – main diagonal
ArCrosstalk – fluorescence

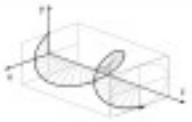
```
typedef struct ArCrosstalk
{
    double  * c;
}
ArCrosstalk;
```



$$n_X = \frac{n_C \cdot (n_C - 1)}{2}$$

Reference Frames for Polarised Light

```
typedef struct ArSVLight
{
    unsigned int        polarised;
    ArStokesVector    * stokesVector;
    ArReferenceFrame    referenceFrame;
}
ArSVLight;
```



```
typedef struct ArMMAttenuation
{
    ArMMAProperties     properties;
    ArMuellerMatrix   * muellerMatrix;
    ArReferenceFrame    referenceFrameEntry;
    ArReferenceFrame    referenceFrameExit;
}
ArMMAttenuation;
```

- Note that light values have one reference frame, but that attenuations have two!

ART: Colour and Light Subsystem #3

- If polarisation support is enabled, all operations with light and attenuation are subject to additional constraints, e.g.
  - Addition of light is only permissible if the light is in collinear reference systems
    (rotate along propagation axis if not aligned)
  - Concatenation (i.e. multiplication) of attenuation values is only permissible if the input and output coordinate systems of the two attenuations match
- These assertions are not checked in normal operation - only in debug mode!