SN Computer Science (2022) 3:12
https://doi.org/10.1007/s42979-021-00870-8

ORIGINAL RESEARCH

Web-Based Interaction and Visualization of Spectral Reflectance Images: Application to Vegetation Inspection

Philippe Colantoni[1] · Jean-Baptiste Thomas[2] · Mathieu Hébert[1] · Jean-Claude Caissard[3] · Alain Trémeau[1]

Received: 4 December 2020 / Accepted: 6 September 2021

Abstract
Visualization of spectral images and interaction with them is still a challenge. We demonstrate an edge-computing, web-technology based solution to handle spectral mage data and allow real-time interaction with it. The solution is flexible, efficient and applicable. It includes visualization strategies based on color science, image processing and statistics. An example of use is provided through a collaboration with a domain knowledge expert for an application related to vegetation inspection.

Keywords Spectral imaging · Data visualization · Real-time spectral image processing · Web technologies · Plant phenotyping

Introduction

Over the past 10 years, web browser has become a major visualization tool for a very wide variety of image contents. However, only Jpeg, PNG, GIF and WebP formats in RGB are supported. In the case of spectral images, there is not only a lack of standard image format, but also a challenging difficulty to cope with the high dimensionality of the information they contain.

In a previous conference communication [14], we proposed a new online visualization tool dedicated to multi- and hyper-spectral reflectance images. The proof of concept undertaken showed that it is possible to use the capabilities offered by modern web browsers to visualize and interact in real time with complex spectral images. This allowed us to demonstrate that these browsers, which normally allow to manipulate standard multimedia files (RGB images, videos, etc.), can also be adapted, with the help of standardized Web technologies (WebAssembly, WebGL, etc.), to other advanced image contents while having processing times close to those obtained by native applications.

In this article we present the methodology we have implemented to develop this tool. We worked on visualization techniques jointly developed with a domain knowledge expert of the scientific problem to be solved. To achieve this, we first detail how we built the foundations (data structures, algorithms and software techniques) needed to enable modern web browsers to manipulate and visualize multi- and hyper-spectral reflectance images using an edge computing approach. Then in a second step, we investigate the usability of this tool, by confronting it to specific purposes related to the spectral analysis of plants, in collaboration with a domain knowledge expert. And thus, we define new functionalities to upgrade our tool so that it fully meet the expectations of the expert. One of the major observation related to the evaluation of the techniques used is that the visualization of custom gradients computed on the spectra appears to be a very efficient approach.

This article is part of the topical collection "Advances on Signal Image Technology and Internet based Systems" guest edited by Albert Dipanda, Luigi Gallo and Kokou Yetongnon.

✉ Philippe Colantoni
philippe.colantoni@univ-st-etienne.fr

[1] Université de Lyon, Université Jean Monnet, CNRS, Institut d'Optique Graduate School, Laboratoire Hubert Curien UMR 5516, 42023, 18 Rue Professeur Benoît Lauras Bâtiment, 42000 Saint-Étienne, France

[2] Department of Computer Science, IDI, NTNU - Norwegian University of Science and Technology The Norwegian color and Visual Computing Laboratory, Teknologivegen, 22, 2815 Gjøvik, Norway

[3] Université de Lyon, Université Jean Monnet Laboratoire Biotechnologies Végétales appliquées aux plantes aromatiques et médicinales, FRE CNRS 3727, Faculté de Sciences et Techniques, 23 Rue Dr Paul Michelon, 42023 Saint-Etienne Cedex 2, France

Published online: 25 October 2021

The first section of this paper presents the types of spectral images that can be handled, as well as a state of the art on visualization methods. It is followed by a second section dedicated to a description of our tool and the demonstrator we have made available online. The third section presents the methodology we have applied to a practical use case, related to the spectral analysis of plants. This paper ends with a conclusion and perspectives that such a tool can offer to the scientific community.

Spectral Images

Spectral Reflectance Images

Reflectance is the fraction of light reflected from the surface of a material. It is defined as the ratio of reflected light flux to incident light flux. The reflectance of a surface generally varies according to the wavelength of the incident light. The curve representing the reflectance is a function of wavelength called a *spectral reflectance*.

It is possible, with specific sensors (filter wheels [8, 39], Spectral Filter Arrays (SFA) [28], diffraction gratings [6, 18], liquid crystal tunable filter [21], etc.) or by reconstruction from RGB images [3, 35], to produce reflectance images where each pixel contains a spectral reflectance. These spectral images are useful to analyze the nature of the materials that compose an object. However, one of the challenge for their use is the lack of standard image format, which should be solved soon attended the recent initiatives, e.g. the one reported in [1].

The very large variety of sensors and techniques to obtain spectral reflectance images generate a multitude of image types with various resolutions, numbers of spectral bands, and sensitivities. A Silios CMS-C multispectral camera based on SFA captures images of size 1280 × 1024 pixels with 9 bands, between 430 and 700 nm, and 10-bit coded information (this kind of image requires to use specific demosaicing algorithm [44]). A HySpex VNIR-1800 camera scans lines of 1800 pixels with 186 bands, between 400 and 1000 nm, and 16-bit coded information using diffraction gratings; A SpecimIQ hyperspectral camera, produces with an integrated scanner that uses diffraction grating, a 512 × 512 pixels image with 202 bands, between 400 and 1000 nm, with 16-bit encoded information. A HySpex VNIR-1800 camera placed on two translation axes X and Y, scanning 10 cm stripes, will generate for a painting of the size of Leonardo da Vinci's Mona Lisa (77 × 53 cm) an image of size 13860 × 9540 pixels with 182 double-byte coded strips which would represent 48.130 GB of data. For the same painting, the multispectral camera Silios CMS-C will produce an image of 1.64 MB of data, with a much lower spatio-spectral resolution.

We see that the amount of information that needs to be treated to process and/or display such images can be very large, making it difficult to handle them. It is therefore essential to have data structures adapted to their use. Due to the large number of possible radiance related data captured by any camera, our proposal assumes that the user has already transformed raw data from their specific devices to standardised reflectance data.

Spectral Image Visualization

In addition to the technical difficulties related to handling a huge amount of information, the visualization of spectral image data is also challenging. The standard representation for color image data is RGB values, i.e., a tri-dimensional information related to a visual and cognitive interpretation of color; while the number of bands of a spectral image is not directly related to any intuitive encoding process related to visualization.

Attempts to visualize spectral images are originally based on a gray-level representation of the scene, and related to a pixel manipulation, e.g., visualization of the spectrum for a specific pixel, band-based, information-based or target-based image processing, such as [25].

Another typical way to look at spectral images is to encode them in a standard colorspace. This can be done through several concepts:

- The selection of three of the spectral bands and their link to the three dimensions of the colorspace. The choice of the bands and their relation to a specific color dimension can be based on priors [5, 29, 32, 43]. This can be followed by an image enhancement process [36] to improve the visibility in the resulting image.
- In the visible range, we can use standard colorimetry rules to compute, or estimate, the colorimetric values in each pixel related to the acquired spectrum. This can be done in real time by the use of GPGPU [12]. Limitations related to this approach are due to: metamerism; settings of the display media that require calibration [13, 16, 46]; impact of illumination [23, 24]; etc. Alternative approaches using metameric blacks to emphasize metameric samples were recently proposed [45].
- Another approach is the dimension reduction based on information theory to three relevant dimensions, then related to a colorspace encoding with false colors. Those techniques include PCA [49] or wavelet [42] decomposition.
- It is possible to implement the fusion of several bands or information channels until convergence to a color or a panchromatic image [27]. Tentative techniques that

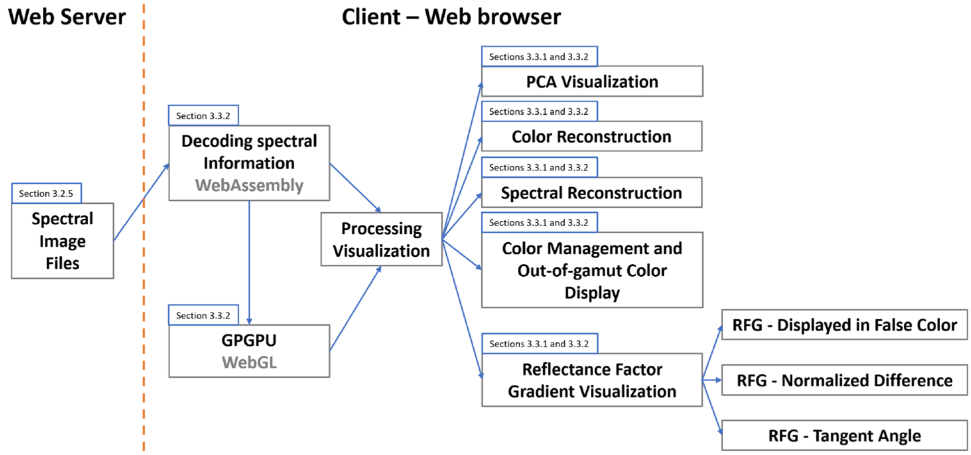


Fig. 1 Hierarchical description of the several modules composing the system. Those modules are described along Section "Online Visualization"

map the spectral information space to color space while keeping the content relevant for human observers were proposed: manifold alignment [31], moving least squares [30], etc.

The limitation of those approaches is the assumption of a working space of lower dimensionality than the spectral space used to visualize the data (i.e. 3 dimensions), and the combination of those spectral dimensions while encoded into a colorspace. To address this issue, other visualization strategies were proposed, e.g., to consider the spectral image as a volume [37] or to look at information in a color space [15]. Those proposals are alternatives to the image format, setting up the problem toward general information visualization, using graphs or other hierarchical structures.

The underlying statement for our research is that a certain level of dynamic interaction is required in some study cases to investigate the spectral image data visually. This interaction must be fast and fairly intuitive, even if the process is conducted by imaging experts. To satisfy this expectation, we propose to enhance our tool with a colorimetric visualization option allowing to play with the illumination properties, including a band selection visualization process and a gradient-based color visualization. These features seem to be new, as far as we could see in this technological domain.

Online Visualization

The choice of an online visualization system is interesting because it allows any user using a web browser (Mozilla Firefox, Apple Safari, Microsoft Edge, Google Chrome, ...) to interact with spectral data located on remote servers without having to install any specific software. In this section, we describe an innovative online visualization tool that provides solutions to various problems inherent to the manipulation of spectral images: reduced IT infrastructure required for its deployment; reduced bandwidth needed to transmit images; ability to modify multiple parameters with a real-time results visualization for an optimal user experience.

In the following sections, we describe various techniques and technologies we have implemented within our standards-only web tool based on HTML5, WebAssembly and WebGL. The Fig. 1 shows the different modules we had to create for our online visualization tool.

Edge Computing

Edge Computing is a form of IT architecture that provides an alternative to Cloud Computing [47]. Rather than transferring the calculated data from a server in a data center, the data to be processed are sent in an optimized way to clients which process them interactively using specialized software or a Web browser. Edge Computing allows us to overcome problems which may be encountered with Cloud Computing: server saturation; insufficient bandwidth; high latency.

It also provides a way to deploy powerful processing and visualization tools without having to invest in very expensive computing infrastructure. We thus selected this IT architecture for our viewer. Our spectral data are stored in an optimal way, using specific data structures, on our server as well as by our software, which is executed in the user's web browser. To speed up the calculations that are performed by the web browser, we have chosen to implement all the required algorithms with WebAssembly (for complex algorithms) and WebGL (for intensive real-time calculations) technologies.

Data Structure

Spectral Reflectance Images

The data is multi- or hyper-spectral images, depending on the dimensionality and the nature of the images. The reflectance estimation is based on a calibration process based on a standard reference. This calibration process allows to produce a reflectance factor image which can be considered as reflectance image only in the case of Lambertian surfaces (therefore perfectly diffuse, this hypothesis is never fully verified but is assumed in this paper).

Hierarchical Transformation of the Information

Spatial resolution We use a Gaussian pyramid that enables us to have a multi-resolution representation of our images. This representation lets us select the level of detail we want to manipulate or display [11]. Each level l of this pyramid has a resolution divided by 2 compared to level $l-1$, level $l=0$ containing the original image. Depending on the usage, it is sometimes interesting to decompose the image into tiles. This representation is particularly interesting when we want to visualize images in very high resolution (VHR) [33] from a server. This representation mode is particularly well suited for dynamic client/server visualization processes where the client only requests the subparts of the image that he/she needs for his/her process while keeping the still visible parts of the image.

Spectral dimension The spectral information that we manipulate with these images is highly correlated. This allows to use different techniques to simplify the interaction with the huge data. We can consider different information hierarchies based on dimensionality reduction techniques [41]. For this, we implemented a global principal component analysis (PCA) [2]. An image with reflectance dimension of m can thus be represented by an image of dimension k (with $k \leq m$), with minimum loss.

Metadata

To facilitate its manipulation, it is possible to add metadata to a complex and voluminous data structure. We integrate metadata that quantify the quality of the spectral and color reconstructions according to the number of dimensions used as a result of the PCA. For this, we used the eigenvalues (sorted in descending order) and 2 metrics described in [22]: the ΔE_{00} [34] on color reconstructions and the root mean square error (RMS error) on spectral reconstructions. This allows us to produce a set of 5 indicators:

1. The sum of the k first eigenvalues (k varying from 1 to m).
2. The average ΔE_{00} values between the color reconstruction, with a D65 illuminant, of the pixels in the original image (encoded on m dimensions) and those reconstructed with k dimensions of the PCA, k varying from 1 to m.
3. The 99 percentile of the ΔE_{00} values between the color reconstruction, with an illuminant D65, of all pixels in the original image (encoded in m dimensions) and the pixels reconstructed with k dimensions of the PCA, k varying from 1 to m.
4. The average value of the root mean square error for all pixels in the original image (encoded on m dimensions) with the corresponding reconstructed pixels with k dimensions of the PCA, k varying from 1 to m.
5. The 99 percentile of the root mean square error for all pixels in the original image (encoded on m dimensions) with the corresponding reconstructed pixels with k dimensions of the PCA, k varying from 1 to m.

Usage Scenarios

The benefits of the metadata are multiple because they allow to have a report on the analysis of the accuracy of the reconstruction according to the number of dimensions chosen. They also allow to define scenarios for the usage of the spectral images stored in this way. For example, we can define a colorimetric scenario based on the 99 percentile of the ΔE_{00} that will indicate the number of dimensions required to have a maximum value of 1.0 or a spectral and colorimetric scenario that defines the dimension necessary to have an average RMS error value below 0.003 and an average ΔE_{00} below 1.0.

For a given image, the chosen usage scenario permits to define the number of dimensions k' to be read on the available k. This allows limiting the network bandwidth required and the memory used while maintaining the digital quality of the reconstruction.

Data Structure

The basic element of our data structure is the *tile* (see Fig. 2). It has a fixed size $t \times t$ pixels. A tile contains m *PCA_Image* (see “Spectral Reflectance Images”) with values that are either floating numbers in single (4 bytes) or half (2 bytes) precision. The values of a tile are the ones of a spectral band at a position and level of our given pyramid.

Standard formats, such as JPEG2000 or TIFF, can be used to store this kind of data structure, but unfortunately it is currently impossible to decode these files in a web browser. We therefore had to choose a simple proprietary format that was easy to decode.

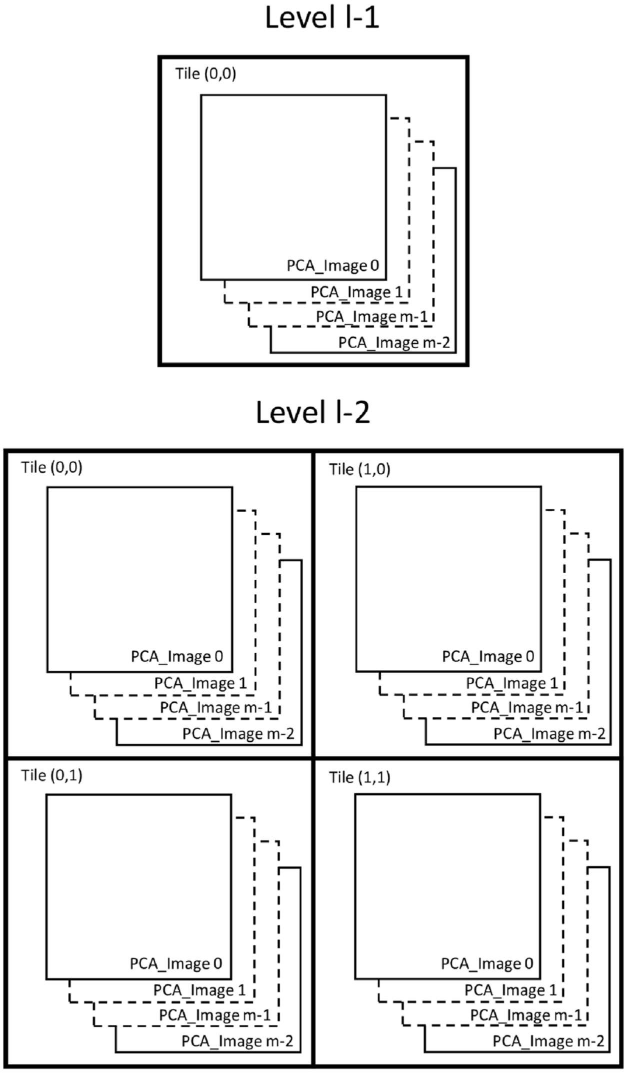


Fig. 2 The last two levels of the data structure (the top of the pyramid)

The tiles are stored successively (see Fig. 3), for a given image, in one or more files (several files can be used if the size of the structure is so large that it is not desired to handle it as a single large file). All metadata as well as data related to resolution, spectral dimension, PCA eigenvectors, pyramid size, coding of the information used (half or single precision) are stored in the file headers.

Calculations

The PCA and metadata values are calculated from the original image, with m spectral dimensions. For this, we use a double-precision algorithm which allows us to process very large hyper-spectral images. The processing time consumed by PCA computation depends on the image size and on the algorithm used. We used a separable algorithm to avoid limitation related to memory. To do this calculation on very large images we perform the computation of the covariance matrix by splitting the spectral image to be processed in tiles.

For each of the eigenvalues, sorted in descending order, we project each of the spectra contained in the pixels of the image using the corresponding eigenvector. Each component resulting from these projections generates an image that is used as basis for calculating a Gaussian pyramid that is decomposed into tiles of size $t \times t$ pixels, starting from the top left corner of the corresponding level in the pyramid, before being stored in the destination file(s).

The 5 indicators that constitute the metadata appear as 5 arrays of size m, the spectral dimension of the original image. The values of the indicators that form these arrays are, for a given index, calculated from the same spectral reconstruction. For the i-index of these tables, the reconstruction is calculated from the first i^{th} components of the PCA.

Test Images

We have chosen the 2 following hyperspectral images Fig. 4 to illustrate our metadata calculations.

Metadata calculation results

In Tables 1 and 2 , we can see the values of the calculated metadata (with $k = 1, 4, 8, 12, 32, 64$) for test images 1 and 2.

The values obtained indicate that it is not necessary to store the information with a single precision (float) encoding because the gain in terms of accuracy, compared to

Fig. 3 Data storage in the file(s)

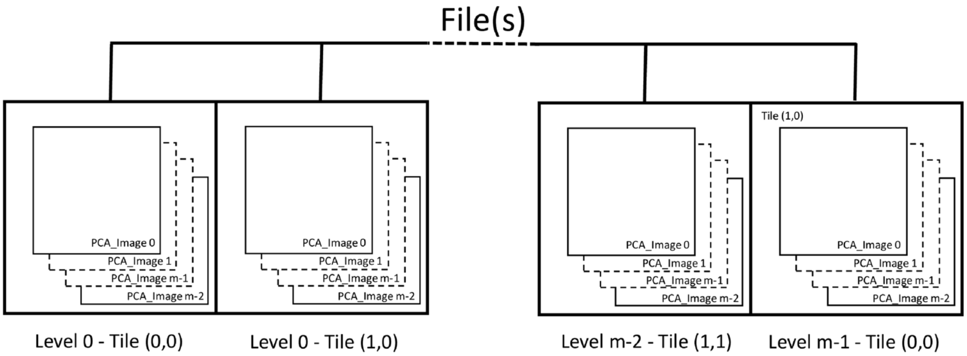
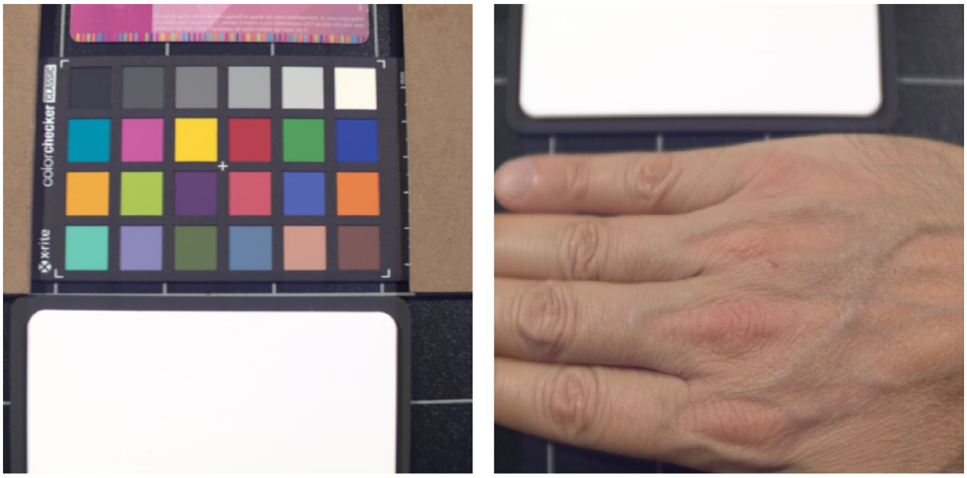


Fig. 4 Test images - Color reconstruction: with a D65 illuminant and a sRGB ICC profile



(a) Test image 1.

(b) Test image 2.

Table 1 Metadata results for test image 1. We can see the values of the calculated metadata (with $k = 1, 4, 8, 12, 32, 64$)

PCA (float)	Number of PCA dimensions, k					
	1	4	8	12	32	64
$sum(\lambda)$	0.8659	0.9964	0.9994	0.9998	0.9999	0.9999
$\overline{X}(\Delta E_{00})$	19.306	4.5692	0.2691	0.1061	0.01882	0.01359
$P_{99\%}\,\Delta E_{00}$	54.60	16.715	1.2437	0.4597	0.07398	0.05358
$\overline{X}(RMSE)$	0.1228	0.01808	0.00748	0.00472	0.00291	0.00186
$P_{99\%}$ RMSE	0.2518	0.06098	0.02416	0.00992	0.00440	0.00305
PCA (half)	Number of PCA dimensions, k					
	1	4	8	12	32	64
$sum(\lambda)$	0.8659	0.9964	0.9994	0.9998	0.9999	0.9999
$\overline{X}(\Delta E_{00})$	19.306	4.5692	0.2696	0.1072	0.02205	0.01757
$P_{99\%}\,\Delta E_{00}$	54.609	16.715	1.2437	0.4598	0.08469	0.07138
$\overline{X}(RMSE)$	0.1228	0.01808	0.00748	0.00472	0.00291	0.00186
$P_{99\%}$ RMSE	0.2518	0.06098	0.02416	0.00992	0.00440	0.00305

half-float, is negligible for a storage twice large. We also see that the gain is marginal for a k value higher than 32. For the two test images, 9 dimensions are enough to obtain a 99 percentile of the ΔE_{00} lower than 1.0.

Conversion Tool

We currently have a converter tool that can handle multi-spectral images from the CRISATEL project as well as hyper-spectral images from the SpecimIQ. This tool, implemented in C++, is cross-platform (Mac OS, Linux and Windows) and can be modified to support other multi- or hyper-spectral image formats. It is also in charge of the metadata computation and we can, if we want, add new statistics (metadata) in our files (for this it will be necessary to reconvert all the original spectral images). This tool is optimized to handle very high resolution multi- and hyper-spectral images.

Web-Based Viewer

In this section, we present the multiple functionalities and implementation methods of our Web visualization tool. This tool is dedicated to process and display the images described in the previous section It has been designed to integrate basic and evolutive functionalities.

Basic Functionalities

We have chosen to include the following basic functionalities in our visualization tool:

Table 2 Metadata results for test image 2. We can see the values of the calculated metadata (with $k = 1, 4, 8, 12, 32, 64$)

PCA (float)	Number of PCA dimensions, k					
	1	4	8	12	32	64
$sum(\lambda)$	0.9195	0.9995	0.9998	0.9999	0.9999	0.9999
$\overline{X}(\Delta E_{00})$	22.951	1.5975	0.34817	0.19544	0.08743	0.05218
$P_{99\%}\Delta E_{00}$	45.696	6.3196	1.7967	0.9407	0.37329	0.2129
$\overline{X}(RMSE)$	0.1006	0.00677	0.00373	0.00332	0.00229	0.00156
$P_{99\%}$ RMSE	0.2400	0.02650	0.00741	0.00553	0.00397	0.00289
PCA (half)	Number of PCA dimensions, k					
	1	4	8	12	32	64
$sum(\lambda)$	0.9195	0.9995	0.9998	0.9999	0.9999	0.9999
$\overline{X}(\Delta E_{00})$	22.951	1.5976	0.3485	0.1961	0.08908	0.05478
$P_{99\%}\Delta E_{00}$	45.701	6.3191	1.7970	0.9395	0.3744	0.2145
$\overline{X}(RMSE)$	0.1006	0.00677	0.00373	0.00332	0.00229	0.00157
$P_{99\%}$ RMSE	0.2400	0.02650	0.00741	0.00553	0.00397	0.00289

PCA visualization: We have raw data which are available as images composed of decimal numbers corresponding to the k first planes (sorted in decreasing order of eigenvalues) that result from the principal component analysis of our original hyper-spectral image. It seemed interesting in some study cases to visualize these images composed of positive or negative decimal values with variable dynamics.

Color reconstruction: The purpose of color reconstruction is to define for each pixel of the image the *CIEXYZ* tri-stimulus values generated when illuminated by a virtual light L. Once reconstructed, the colors are displayed on the screen using a color management process. The numerical values for the pixel below the mouse cursor are displayed in a dedicated information frame at the bottom left of the display window. This information box displays the calculated *CIEXYZ* values as well as the corresponding *CIELAB* values (under illuminant L) and the *RGB* values calculated during the color management.

Spectral reconstruction: The purpose of spectral reconstruction is to generate by interpolation, for a given wavelength, a reflectance image. We provide the possibility to simultaneously display 3 reflectance images (for 3 given wavelengths) assigned to the R, G and B channels of the displayed image which gives us a false color visualization. The wavelengths associated with the R, G and B channels of the pixel below the mouse cursor are displayed in the information frame (the spectrum will be available soon).

Color management and out-of-gamut color display: Color management consists in a controlled transformation between the color representations of different devices. In the context of a color reconstruction, the calculated *CIEXYZ* values are transformed into an *RGB* triplet with a complex transformation managed by the color management module (CMM). Since screens can only display a limited number of colors, we have also set up a specific display mechanism for the so-called "out of gamut" colors (not directly displayable on the screen) with a color code that assesses the distance between these colors and the gamut surface of the target screen. The gamut corresponds to the surface of the 3D shape containing all the colors displayable by a screen. When a color is out of the gamut, we bring it back to the gamut surface using a gamut clipping technique. The out-of-gamut distance is then defined by the Euclidean distance in the *CIELAB* space between the original color and the one on the surface of the gamut.

Reflectance factor gradient visualization (RFG): The gradient of a 1D-function at a given point C is a value that characterises the variability of this function in the neighbourhood of this point. It can be approximated by the difference between 2 values of this function which are respectively located to the right and to the left of this point. If we consider that the spectral reflectance in a pixel of our image can be modeled by a multi-dimensional function $R(\lambda)$ we can compute for each of its pixels the value of this gradient corresponding to the wavelength λ_C and thus obtain:

- An image of the gradients that can be directly displayed as false colors.
- An image of the normalized differences.
- An image containing the values of the angles of the tangent slopes corresponding to this wavelength.

Implementation

Our implementation is mainly based on 2 Web technologies. WebAssembly allowed us to transpose our C++ library which manages our spectral data files (and the associated data structures) as well as our color management functions as modules that can be accessed directly through JavaScript. WebAssembly allows us to have execution times that are close to those we can have with native applications. WebGL 1.0 allowed us to access to the computing power of the graphics card used to execute the web browser, via graphic shaders, and to transfer the image data as textures that are also stored in the graphics card's memory for maximum efficiency.

The heterogeneity in the sources of the reflectance spectra that we have to process (cf. subsection "Spectral Reflectance Images") requires to standardize the sampling of these spectra. Our color and spectral reconstructions are based on tabulated data available with a $1nm$ spectral sampling, so we have chosen to use this sampling. To do this, all the reflectance spectra we use for our pre-calculations are resampled using an Akima interpolation [4].

PCA visualization: Different strategies can be used for displaying the images corresponding to the k first planes from the PCA (images composed of positive or negative decimal values with variable dynamics):

- Scaling with grayscale visualization: the minimum value is set to 0 and the maximum to 1.
- Scaling of absolute values with visualization as grayscale: same as the previous method but with absolute value scaling.
- Use of pseudo-color shading with a maximum limit.

We have chosen to implement this third visualization method as it allows to have an accurate view of the values (positive and negative) contained locally and globally in the images. For this purpose we propose to use several parameters that can be adjusted as follow:

- A first color for the value 0.
- A second color for the maximum positive value and a shading for intermediate values.
- A third color for the minimum negative value and a gradient for the intermediate values.
- The ability to apply a gamma, a gain, and an offset correction.

Color reconstruction: The computation of the color reconstruction for a given spectral reflectance $R(\lambda)$ and a spectral radiance $L(\lambda)$ is performed in the color space *CIEXYZ* 1931 or 1964 with the following formula:

$$\begin{cases} X = \sum_{\lambda=400}^{\lambda=760} \bar{x}(\lambda)R(\lambda)L(\lambda) \\ Y = \sum_{\lambda=400}^{\lambda=760} \bar{y}(\lambda)R(\lambda)L(\lambda) \\ Z = \sum_{\lambda=400}^{\lambda=760} \bar{z}(\lambda)R(\lambda)L(\lambda) \end{cases}$$

where $\bar{x}$, $\bar{y}$ and $\bar{z}$ are the color matching functions of the CIE 1931 or 1964 standard observer, at a sampling of 1nm. In the case of a representation of spectral data after the PCA this color reconstruction is simplified because we only have to pre-calculate with the previous formula the *CIEXYZ* values for each of the eigenvectors used. The *CIEXYZ* values for a given pixel (i, j) is then, if we use k dimensions for reconstruction:

$$XYZ(i,j) = \sum_{d=0}^{d=k-1} PCA_d(i,j) * XYZ_{eigenvector_d}$$

The transformation of the tri-stimuli so obtained into *RGB* values dedicated for the screen is then done through a color management process that requires to transform these *CIEXYZ* values into *CIELAB* values that will be used as input for interpolation process made with the 3D Look Up Table (LUT) defined bellow. *Spectral reconstruction* To compute, at a given wavelength, a reflectance image, we use the tabulated values coming from the interpolated eigenvectors of the PCA. For this wavelength, the reflectance value W of a pixel (i, j) is computed with the following formula:

$$W(i,j) = \sum_{d=0}^{d=k-1} PCA_d(i,j) * eigenvector_d[\lambda]$$

Color management and out-of-gamut color display: Color management is performed using a pre-calculated 3D LUT [12] with a function from our WebCMM (our JavaScript color management module) based on a calibration technique described in [13]. The implementation of this technique was made possible by the creation of a color characterization tool dedicated to Web browsers that we have described in [17]. The computation of the out-of-gamut distance is also performed with this 3D LUT which contains *RGBA* color vectors where the component A is the out-of-gamut distance. The *CIELAB* values that has been calculated in the color reconstruction are used as input for the computation, by a tri-linear interpolation inside the 3D LUT, of the *RGB* color and the A value.

Reflectance Factor Gradient (RFG) computation : We propose a new, very general, definition to approximate a discrete gradient on spectra. The sensitivity and resolution of this gradient are flexible. This definition is a powerful tool to look for spectral pattern in images, when combined to a false colour visualization strategy. We define the gradient of the reflectance factor at a given wavelength λ_C as the difference of the

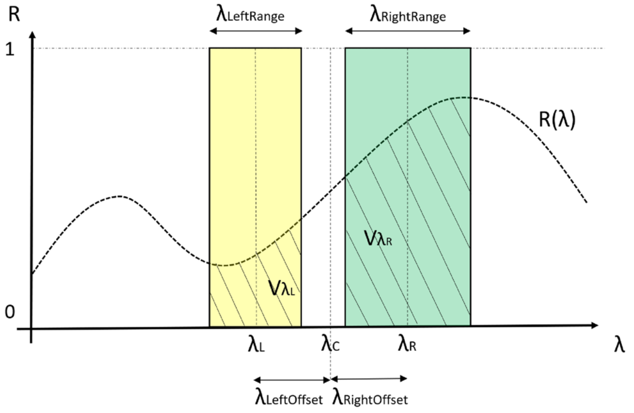


Fig. 5 Image to explain the computation of RFG_{λ_C} at λ_C, from V_{λ_L} (dashed yellow) and V_{λ_R} (dashed green). The characteristics of the filters used to compute V_{λ_L} and V_{λ_R} are based on two independent width and on two offsets on the left and right of λ_C

reflectance value V_{λ_R} at the right of λ_C with the value V_{λ_L} at the left of λ_C (see Fig. 5).

With :

$$\lambda_R = \lambda_C + \lambda_{RightOffset}$$

$$\lambda_L = \lambda_C - \lambda_{LeftOffet}$$

where $\lambda_{RightOffset}$ and $\lambda_{LeftOffset}$ correspond to the right and left offsets of λ_C. V_{λ_R} and V_{λ_L} are then given by:

$$V_{\lambda_R} = \frac{\sum_{\lambda=\lambda_R-\frac{\lambda_{RightRange}}{2}}^{\lambda=\lambda_R+\frac{\lambda_{RightRange}}{2}} R(\lambda)}{\lambda_{RightRange}}$$

$$V_{\lambda_L} = \frac{\sum_{\lambda=\lambda_L-\frac{\lambda_{LeftRange}}{2}}^{\lambda=\lambda_L+\frac{\lambda_{LeftRange}}{2}} R(\lambda)}{\lambda_{LeftRange}}$$

where $\lambda_{RightRange}$ and $\lambda_{LeftRange}$ correspond to the widths of the spectral bands over which the average reflectance value is calculated. Finally, the gradient value is given by:

$$RFG_{\lambda_C} = V_{\lambda_R} - V_{\lambda_L}$$

For reflectance values defined between 0 and 1, the value of an RFG_{λ_C} can vary between − 1 and 1.

In the case of a representation of spectral data after the *PCA* the RFG_{λ_C} computation is simplified because we only have to pre-calculate with the previous formula for each of the eigenvectors used. The RFG_{λ_C} values for a given pixel (i, j) is then, if we use k dimensions for reconstruction:

$$RFG_{\lambda_C}(i,j) = \sum_{d=0}^{d=k-1} PCA_d(i,j) * RFG_{\lambda_{C_{eigenvector_d}}}$$

RFG displayed in false color: The visualization technique we use to display RFG images is identical to the one we use to display PCA images without the ability to apply gamma or offset corrections.

Several parameters can be adjusted as follow:

- A gain that can be applied to RFG values.
- A first color C_0 for the value 0.
- A second color C_1 for values higher than or equal to 1, and a shading between C_0 and C_1 for values between 0 and 1.
- A third color C_2 for values lower than or equal to − 1, and a shading between C_2 and C_0 for values between − 1 and 0.

RFG- Normalized difference displayed in false color: The normalized difference for a gradient is defined as follows:

$$RFGN_{\lambda_C} = \frac{V_{\lambda_R} - V_{\lambda_L}}{V_{\lambda_R} + V_{\lambda_L}}$$

As for the *RFG* value, the *RFGN* value can vary between − 1 and 1. To be consistent with previous methods, we have decided to display the *RFGN* images using the same method with the same parameters as for the *RFG* images.

RFG -Tangent Angle (RFG-TA) displayed in false color: The gradient can be computed at any point in the reflectance spectrum. We can therefore use its value to approximate the slope of a pseudo-tangent at this point and thus obtain the angle corresponding to this slope. It is called a pseudo-tangent because the true tangent to the reflectance curve $R(\lambda)$ at a point A of abscissa λ_C is the limit position of the secant line (AB) when point B of the curve defined by $R(\lambda)$ tends to point A. It is possible to define $R(\lambda)$ as an interpolation function calculated from the N values of the spectral reflectance we have in each pixel and so have the exact value of the slope on the tangent by calculating the derivative $R'(\lambda)$. Instead, we chose to compute the slope of a pseudo-tangent as follow:

$$RFGS_{\lambda_C} = ScalingFactor \times \frac{RFG_{\lambda_C}}{\lambda_{RightOffset} + \lambda_{LeftOffset}}$$

We can then determine the value of the angle corresponding to the slope of this pseudo tangent by computing its arctangent. Instead, we have chosen to use the trigonometric function *atan*2 to calculate this angle with:

$$RFGTA_{\lambda_C} = atan2(ScalingFactor \times RFG_{\lambda_C}, \lambda_{RightOffset} + \lambda_{LeftOffset})$$

which yields angle values between $-\pi$ and $+\pi$. We can use the same visualization technique for $RFGTA_{\lambda_C}$ values as the one used for PCA visualization. To diversify our visualization modalities and better differentiate the computed angles, we have decided to use a completely different technique based on a rainbow false color LUT using the following algorithm:

- Step1 - Scaling the value of $RFGTA_{\lambda_C}$: $S(RFGTA_{\lambda_C}) = 6 \times \frac{RFGTA_{\lambda_C}+\pi}{2\times\pi}$. The values of S will thus vary between 0 and 6.
- Step 2 - Generating a HSV_S color from $S(RFGTA_{\lambda_C})$ with:
    - $Hue = S(RFGTA_{\lambda_C})$
    - $Saturation = 100\%$
    - $Value = 100\%$
- Step 3 - Color conversion from HSV_S to RGB_S.

Web integration: The spectral image is associated with a web page through a JavaScript initialization function with parameters that define the resolution level l used (the level inside the pyramid stored in the file) as well as the dimension k of the PCA used (on the m dimensions available). This dimension k can be defined as a constant or computed by a function that we provide. This function will determine the value of k according to the use scenario wanted by the user and the metadata embedded in the file. This function downloads all necessary data from a server into the browser's memory, which then transfers them inside 4-dimensional textures (*RGBA* encoded) either in single precision (float) or in half precision (half float) format (depending on the encoding of the information inside the file). The use of *RGBA* textures allows us to package our k dimensions inside $(k-1)/4+1$ textures. The limited number of texture units available in a graphical shader imposes this packing method (see 3.5).

All these calculations are integrated into a single graphical shader, a fragment shader, that is in charge of the source spectral image processing process. The source code of this shader is dynamically generated during the spectral image initialization in order to be adapted to the used k dimensions of the PCA (and to overcome some WebGL 1.0 limitations). The shader is then compiled by WebGL to be directly executable by the GPU of the graphics card. This shader, which is executed on all pixels of the source spectral image, generates an *RGB* image as a new WebGL texture (which have the same size as the source spectral image at the resolution level l), which is then displayed in our web page (with the possibility to zoom in) through the JavaScript Three.js framework.[1]

All computations related to implementation of these basic functionalities (see section "Hierarchical Transformation of the Information") are performed within this shader even if, in the end, the result of only one of these visualization modes is displayed on the screen.

Custom Shader

In the previous subsection, all of the processing/visualization is performed in a single graphical shader with a dynamically created source code. It is therefore possible to transform this shader's code to make it perform other processing/visualizations. It is the concept of custom shader [40] which we implement on the basis of a new function which will replace the one which implements the basic treatment/visualization functions of our tool. Custom shaders make it possible to implement treatment/visualization processes adapted to very specific uses of spectral images (medical, heritage culture, etc.)

In the previous section, we indicated that the main shader performs all the computations related to the basic functionalities we have chosen to implement, even if we do not display all of them. We made this choice to increase the possibilities offered by the custom shader that will be produced for our visualization tool.

Visualization Examples

Examples of visualization results obtained with the image test 1[2] for these basic functionalities are illustrated after: Fig. 6a for PCA visualization; Fig. 6b for color reconstruction; Fig. 6c for color management: images Fig. 6c for out-of-gamut color display; Fig. 7b for spectral reconstruction; Fig. 8a for reflectance factor gradient displayed in false colors; Fig. 8b for reflectance factor gradient - Normalized difference displayed in false colors; Fig. 8c for reflectance factor gradient - Tangent angle displayed in false colors.

Limitations

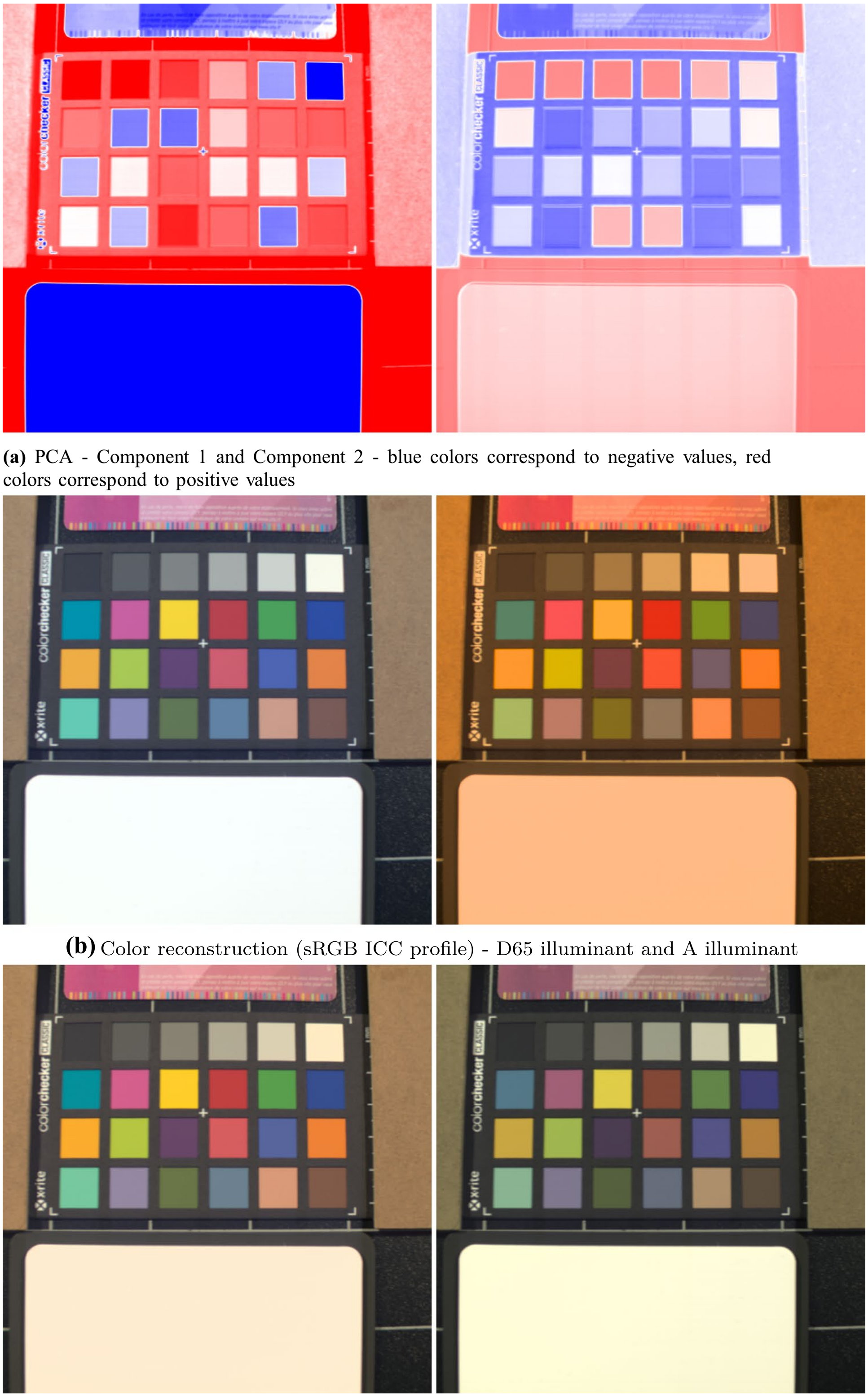
This tool depends on two web technologies which, unfortunately, suffer from several limitations.

- WebGL 1.0: The maximum texture size which is generally 16384 × 16384 and therefore, the images that we can process / view. The number of textures usable in a shader which is generally 16, this limits the dimension k that we can use up to 64 (16 × 4 with our packing method). The instructions number that a shader can execute which may limit the capabilities of the custom shaders that will be produced.

[1] https://threejs.org

[2] https://www.couleur.org/articles/SN-ComputerScience2021/pca-v2.html?image=data/specimiq-1.pca&dim=30

Fig. 6 Visualization results 1/3

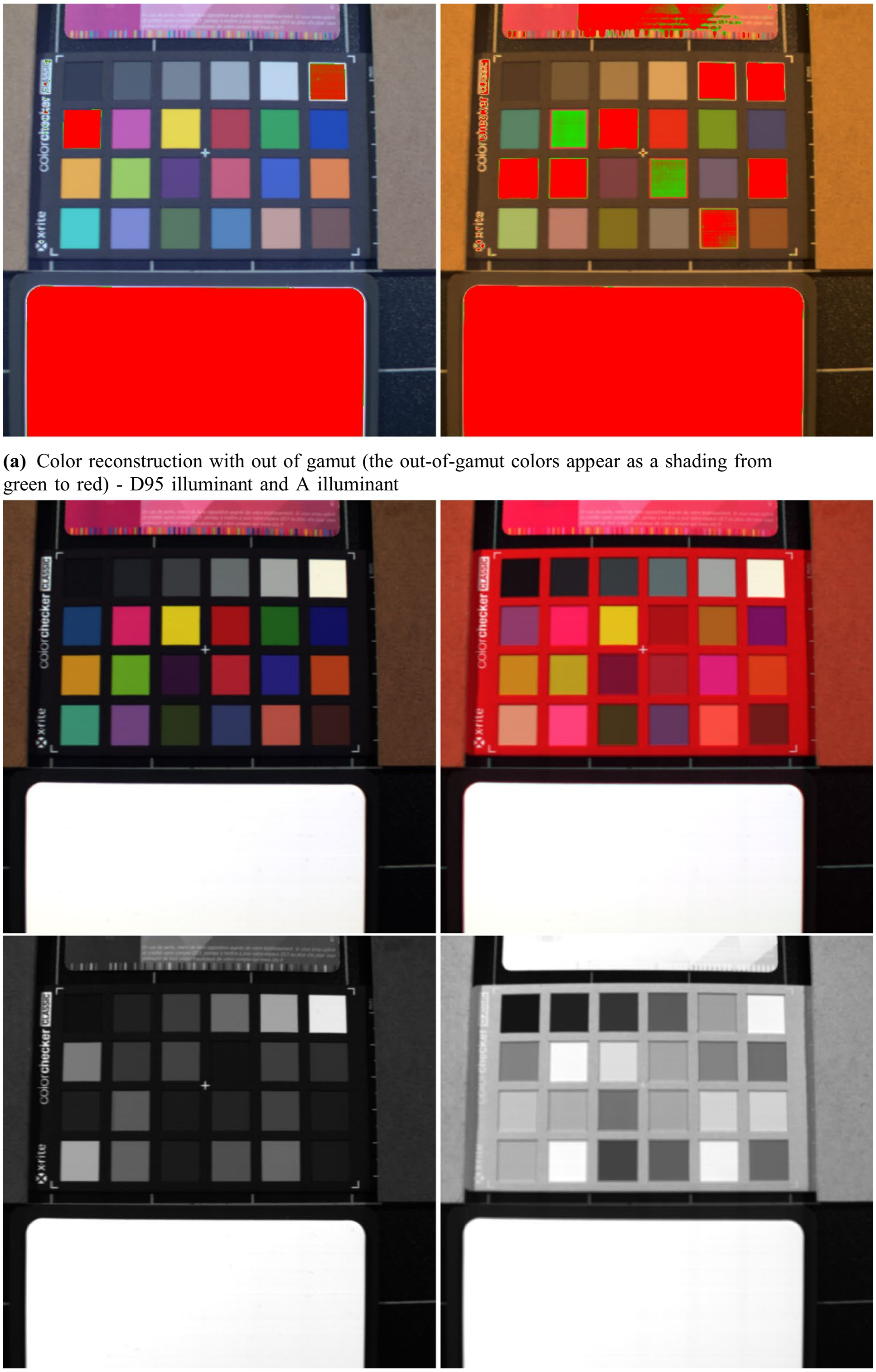


(a) PCA - Component 1 and Component 2 - blue colors correspond to negative values, red colors correspond to positive values

(b) Color reconstruction (sRGB ICC profile) - D65 illuminant and A illuminant

(c) Color management (D50 illuminant) - sRGB profile and ProPhoto profile

Fig. 7 Visualization results 2/3



(a) Color reconstruction with out of gamut (the out-of-gamut colors appear as a shading from green to red) - D95 illuminant and A illuminant

(b) Wavelength reconstruction - λ_R=650 nm, λ_G=550 nm, λ_B=450 nm (upper left) - λ_R=900 nm, λ_G=550 nm, λ_B=450 nm (upper right) - λ_R=500 nm, λ_G=500 nm, λ_B=500 nm (bottom left) - λ_R=850 nm, λ_G=850 nm, λ_B=850 nm (bottom right)

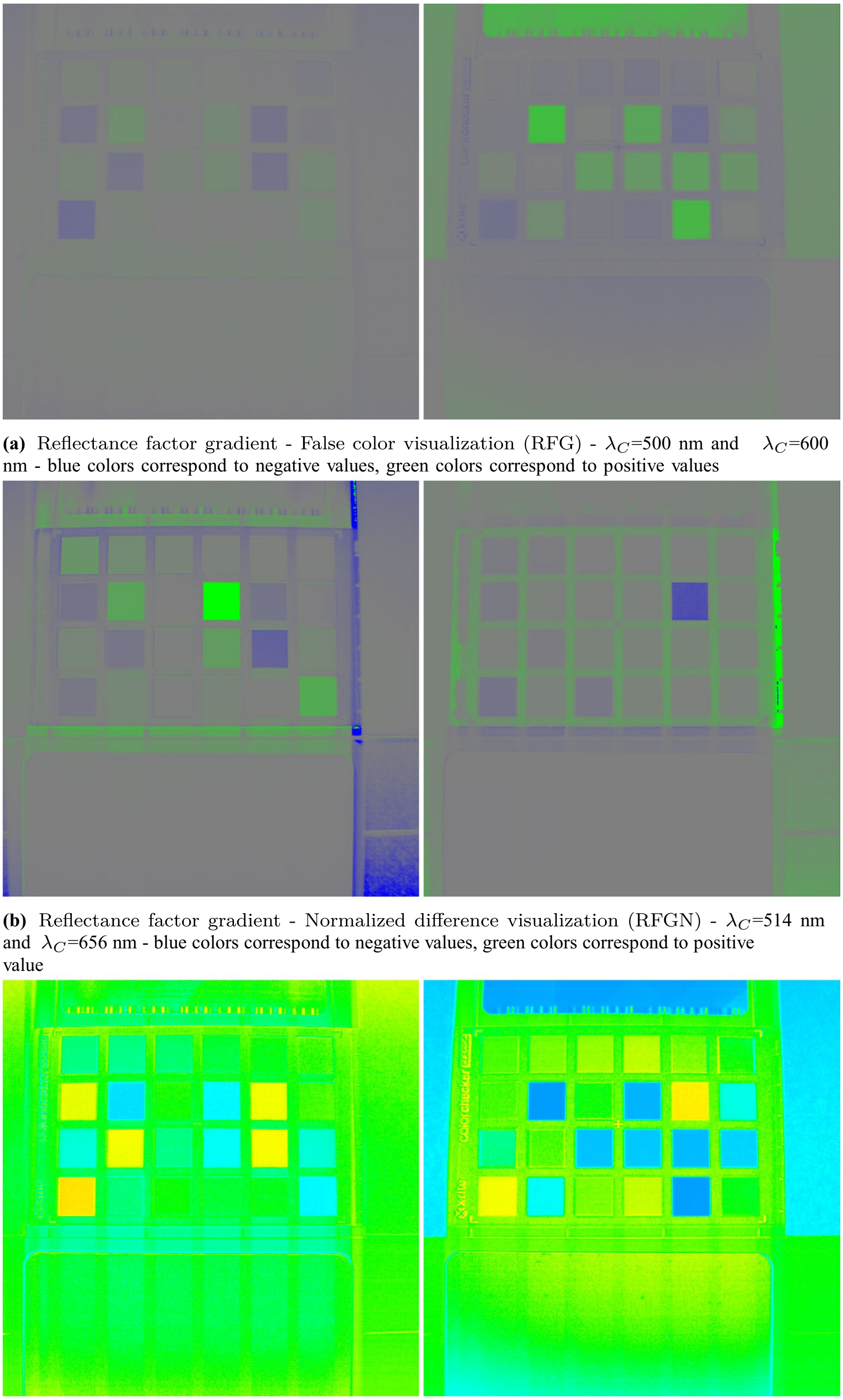
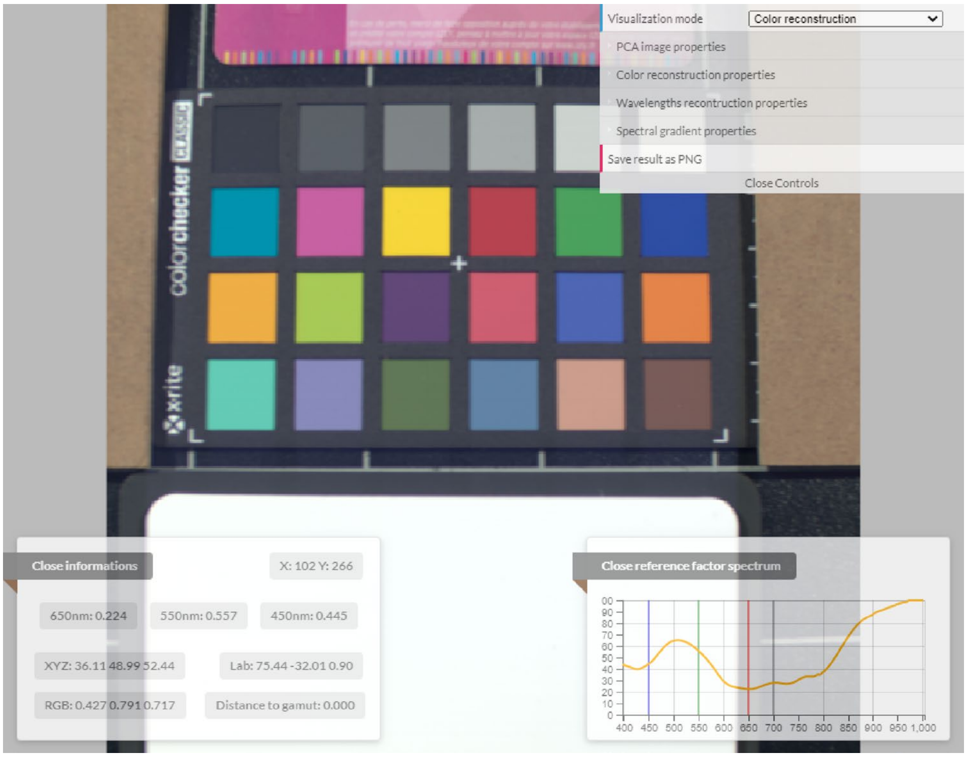


Fig. 8 Visualization results 3/3

(a) Reflectance factor gradient - False color visualization (RFG) - λ_C=500 nm and λ_C=600 nm - blue colors correspond to negative values, green colors correspond to positive values

(b) Reflectance factor gradient - Normalized difference visualization (RFGN) - λ_C=514 nm and λ_C=656 nm - blue colors correspond to negative values, green colors correspond to positive value

(c) Reflectance factor gradient - Tangent angle visualization (RFGTA) - λ_C=500 nm and λ_C=600 nm - the angle value is transferred into the hue component of a HSV color and then converted as a RGB color


Visualization mode: Color reconstruction

X: 102 Y: 266

650nm: 0.224 550nm: 0.557 450nm: 0.445

XYZ: 36.11 48.99 52.44 Lab: 75.44 -32.01 0.90

RGB: 0.427 0.791 0.717 Distance to gamut: 0.000

Fig. 9 Spectral image view window full interface

- WebAssembly: Compatibility issues with some Web browsers, mainly on smartphones and tablets.

For these reasons we recommend to use web browsers running on operating systems dedicated to laptops and PCs (Windows 10, Mac OS, Linux distributions like Ubuntu, etc.).

Performance

For a spectral image acquired with the SpecimIQ camera with $k = 30$, visualized under Windows 10 (PC with an AMD Ryzen 9 3900X processor and a GeForce RTX 2800 graphics card) and MacOS 10.15.7 (Intel Core i7-4980HQ processor with an integrated Intel Iris Pro graphics card), with different web browsers, we obtained the following execution times:

Windows 10 - 20H2	WebCMM	Loading image	Shader execution
Google Chrome 86	1567 ms	145ms	0.015 ms
Mozilla Firefox 82	3591 ms	338ms	0.024 ms
Microsoft Edge 86	1573 ms	145ms	0.015 ms

Mac OS	WebCMM	Loading image	Shader execution
Google Chrome 86	1001 ms	82 ms	0.41 ms
Apple Safari	1111 ms	142 ms	0.87 ms

Notes: The WebCMM initialization does not depend on the resolution of the source image and we use a local web server to evaluate the processing time to load the hyperspectral image. The image processing time correspond to the shader execution and our tool is not working with Firefox on MacOS because texture size limitations).

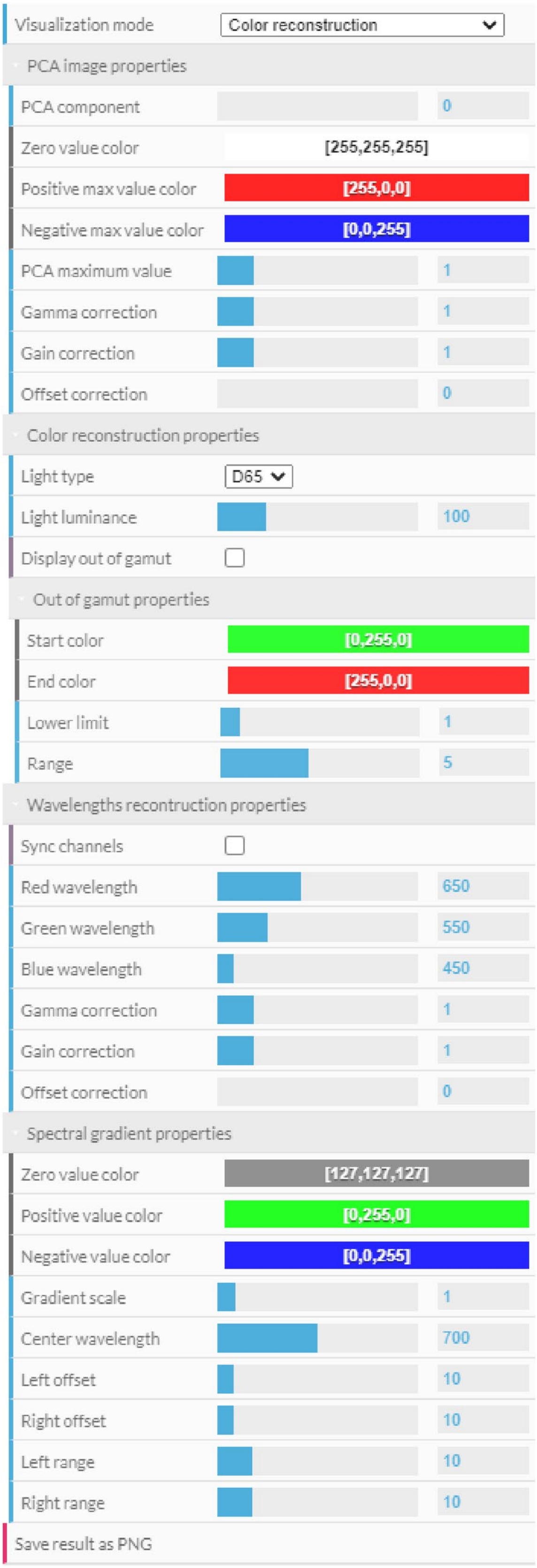
Demonstrator

A demonstrator is available at the following addresses:

- https://www.couleur.org/articles/SN-ComputerScience2021/.
- https://ipem.univ-st-etienne.fr/articles/SN-ComputerScience2021/.

It allows:

- The visualization of the two test images presented in the section 3.2.7 with several initial configurations where we can change:
  - The number of dimensions used.
  - The testing of color management settings.
- To test 2 custom shaders dedicated:



Visualization mode	Color reconstruction
PCA image properties	
PCA component	0
Zero value color	[255,255,255]
Positive max value color	[255,0,0]
Negative max value color	[0,0,255]
PCA maximum value	1
Gamma correction	1
Gain correction	1
Offset correction	0
Color reconstruction properties	
Light type	D65
Light luminance	100
Display out of gamut	☐
Out of gamut properties	
Start color	[0,255,0]
End color	[255,0,0]
Lower limit	1
Range	5
Wavelengths recontruction properties	
Sync channels	☐
Red wavelength	650
Green wavelength	550
Blue wavelength	450
Gamma correction	1
Gain correction	1
Offset correction	0
Spectral gradient properties	
Zero value color	[127,127,127]
Positive value color	[0,255,0]
Negative value color	[0,0,255]
Gradient scale	1
Center wavelength	700
Left offset	10
Right offset	10
Left range	10
Right range	10
Save result as PNG	

Fig. 10 Control panel

- For color segmentation.
- For highlighting spectral properties, such as the infrared reflectance of an image, during its color reconstruction.

Each of the links presented on the home page of our demonstrator opens a new web page that displays the corresponding image with a classic or custom shader-based view. All spectral image view windows (see Fig. 9) have a control panel (see Fig. 10) that allows changing many display settings in real-time presented in section "Hierarchical Transformation of the Information".

These spectral image view windows contain, in addition to the computed image and the control panel, two subwindows which give access to various information related to the current pixel (the pixel under the mouse cursor):

- the xy position of this pixel;
- the values of the reconstructed color in the RGB, CIELAB and CIEXYZ spaces;
- the distance to the gamut (to quantify the out-of-gamut error);
- the values of the 3 reconstructed wavelengths;
- the full spectrum reconstructed corresponding to this pixel.

Only this demonstrator is currently available. The C++ code of our conversion tool as well as the one used for our Web tool is not available, for the moment.

Application to Vegetation Inspection

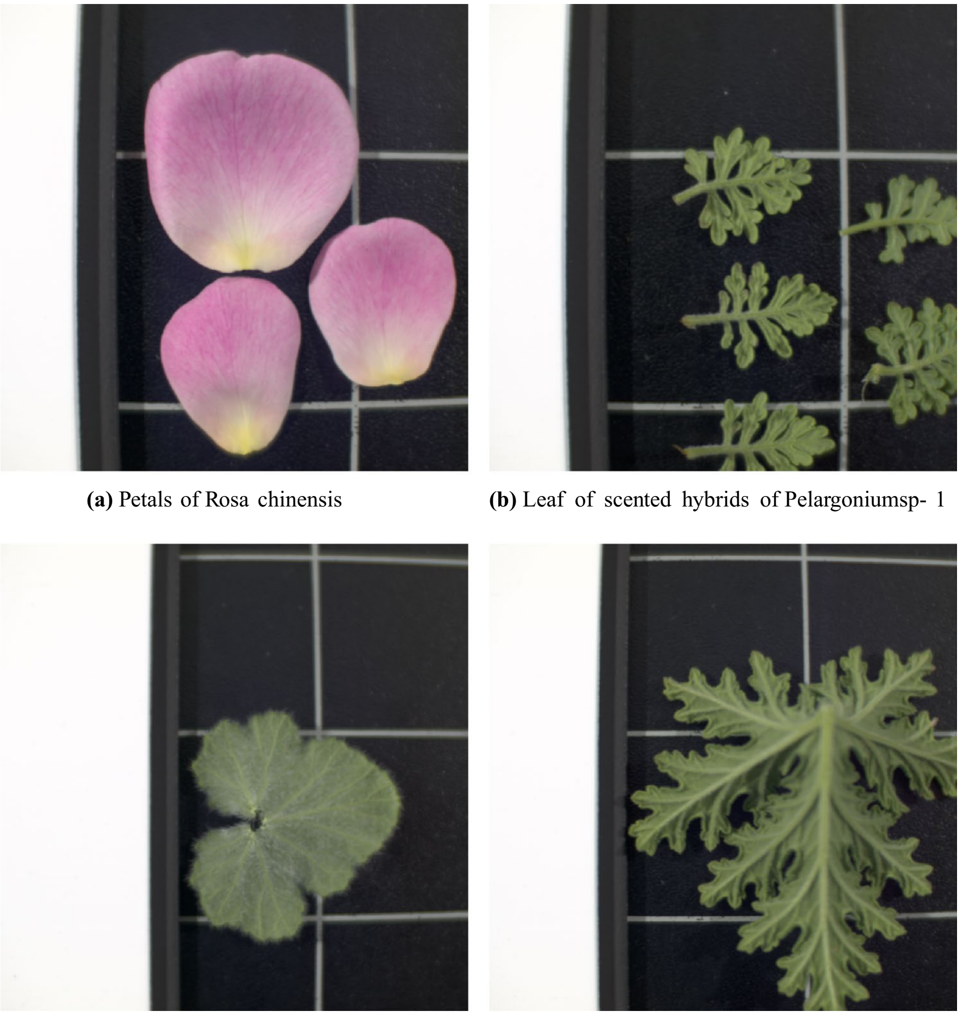
This tool has been used, in collaboration with the *Laboratoire de biotechnologies végétales appliquées aux plantes aromatiques et médicinales (LBVpam)* of University Jean Monnet at Saint-Etienne, for scientific purposes. As first application we used this tool for plant inspection.

The LBVpam is specialized in the study of the diversity and biosynthesis of volatile plant compounds, particularly in perfume, aromatic and medicinal plants. The research conducted by this laboratory aims to better understand the mechanisms of production, secretion and roles of volatile compounds in these types of plants. It is important to note that our partners at the LBVpam are not imaging specialists.

Methodology

To make the best use of our tool that contains several visualization methods and a large number of adjustable parameters, we propose to use it in presence of two experts: one

Fig. 11 Selected images



(a) Petals of Rosa chinensis

(b) Leaf of scented hybrids of Pelargoniumsp- 1

(c) Leaf of scented hybrids of Pelargonium sp- 2

(d) Leaf of scented hybrids of Pelargonium sp- 3

in spectral imaging and another one in the field of application studied. The expertise in the application domain is necessary to help identify, characterize, categorize relevant information from our visualization tool. Following a first exchange meeting with LBVpam expert, we agreed on a 4-step strategy:

1. Captures of hyper-spectral images with the SpecimIQ camera from living leaves and flower petals selected for analysis by the LBVpam.
2. Pre-analysis of 4 images selected by the LBVpam with the previous version of our visualization tool and recommendation for new functionalities.
3. Implementation of the new functionalities.
4. Analysis of the images with the new functionalities available.

The wish-list fixed by the LBVpam were:

1. To detect the presence of aromatic glands on leaves of scented Pelargonium species [7];
2. To reconstruct the structures of leaf veins;
3. To analyze the growth processes on rose petals.

Capturing images

The acquisition of the hyper-spectral images was carried out in a dedicated room with a lighting system composed of two 750 W halogen sources placed on each side of the camera and positioned at 2 m and 45° from the samples. With such a lighting arrangement we were able to produce a good homogeneous illumination on the acquisition surface (surfaces varying from $100mm \times 100mm$ to $140mm \times 140mm$. We have chosen to keep a white reference sample in all our captures to achieve an optimal calibration for acquired hyper-spectral images. Four plants selected by the LBVPam have been used for these captures and allowed us to produce several spectral images.

Pre-analysis

From these captures we have selected four images that were considered interesting for this study (see Fig. 11). These four images, which were used to perform a pre-analysis on leaf and petal samples, enabled us to establish some initial observations.

Detection of leaf aromatic glands: It does not seem possible to visually highlight the aromatic glands present in the leaves, even in the infrared range (wavelengths above 780 nm). The area covered by one pixel of our sensor (which has a resolution of 512 × 512 pixels and can only be placed at a minimum distance of 150mm from the samples) is too large compared to the size of these glands.

Reconstruction of leaf vein structures: Methods already exist to produce this type of images [9] and our real-time analysis and visualization tool is not particularly well adapted to reveal this kind of information, even if it is partially present in some PCA images, due to the spatial resolution of the camera. Computations related to the vein structure image can nevertheless be performed simultaneously to our PCA calculation. The resulting image may then be added into the files that store the data structures of our hyper-spectral images.

Analyzing growth of petals: The image corresponding to Fig. 11a shows three petals of the same rose (*Rosa chinensis*) with different states of growth, from the inside to the outside of the flower. This image was selected in order to study if the growth processes were colorimetrically and/or spectrally uniform. Possible discontinuities could be interpreted as chemical changes within the petals. Knowing that the color appearance of flower petals is based on optical models that can be approximated by stacking micro-structured layers and chemical pigments [19, 20, 26, 48] we decided to perform first a colorimetric study and then a spectral study.

Colorimetric analysis: The first analysis was qualitative, it was based on color reconstructions visualized on a calibrated screen (with no "out of gamut" colors) observed by our two experts. Despite the use of several virtual illuminants we did not perceive anything particular. The growth appears as a color gradient without any perceptible discontinuities (see left image in Fig. 12a). The second analysis was quantitative with color segmentations made with our custom shader from reference colors taken in the gradients from the base of the petal to its edge (see Fig. 12b). Same observation, the gradients seem continuous and do not differ in the three petals present in the image.

Spectral analysis: We chose to perform a qualitative spectral analysis by displaying spectral information as grayscale images for a given wavelength with wavelengths varying between 400 nm and 1000 nm. By dynamically varying the wavelength we were able to quickly highlight a tenuous phenomenon at the base of one of the petals having reached full maturity (red square in the left image of Fig. 12c corresponding to 675 nm), phenomenon that was not visible in our color analysis. By analyzing the reflectance spectra in this petal area we were able to see a valley around 675 nm in the spectrum (red square in the right image of Fig. 12c), which looked like the distortion of photosynthetic pigments reflectance spectra [10]. Indeed, it is known that some petals have some chloroplasts that developed at their basis [38]. This observation means that the image analysis tool makes it possible to study the development of chloroplasts during petal growth with very good precision.

Implementing the New Functionalities

Following these first promising results, we wondered about the implementation of a visualization method that would allow us to highlight this type of phenomena induced by changes in local curvatures in the reflectance spectrum. The visualization of gradients in the spectra (see gradient section) seemed to be an appropriate answer.

Analysis of the Images According to the New Functionalities Available

In the Fig. 13a and 13b we can see the results of the gradient visualization, centered at 675 nm, for the "Reflectance Factor Gradient in false color" and "Reflectance Factor Gradient - Tangent Angle" modes.

The information that we were able to perceive during our qualitative spectral analysis is now clearly apparent. Indeed, the orange zone that appears in the Fig. 13b corresponds exactly to the zone we detected. This new visualization method is therefore well adapted to this problem (at least for this type of flower).

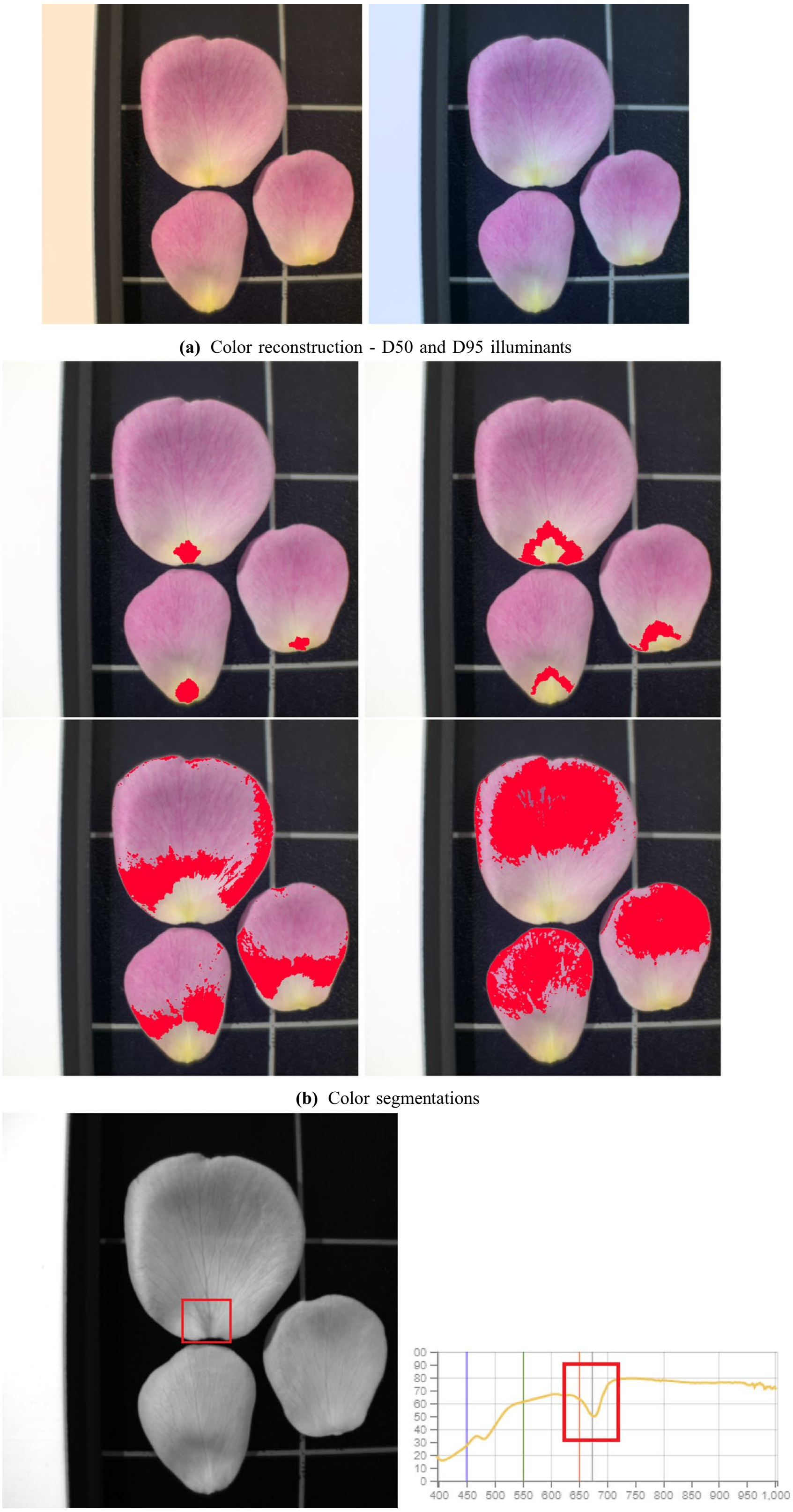
Demonstrator

The four hyper-spectral images that have been selected by the LBVPam are available here https://www.couleur.org/articles/SN-ComputerScience2021/ on the same web page dedicated to our demonstrator with a mirror page at the following addresses: https://ipem.univ-st-etienne.fr/articles/SN-ComputerScience2021/. For these four images we have chosen to use 30 dimensions of the PCA in order to get high quality spectral reconstructions.

Conclusion and Perspectives

In this paper, we presented a new online visualization tool dedicated to multi- and hyper-spectral reflectance images based on edge computing concepts. This tool, presented in

Fig. 12 Pre-analysis



(a) Color reconstruction - D50 and D95 illuminants

(b) Color segmentations

(c) Left: Wavelength reconstruction 675 nm - Right: Reflectance spectrum in the center of the red square (with a distortion of photosynthetic pigments reflectance spectra)

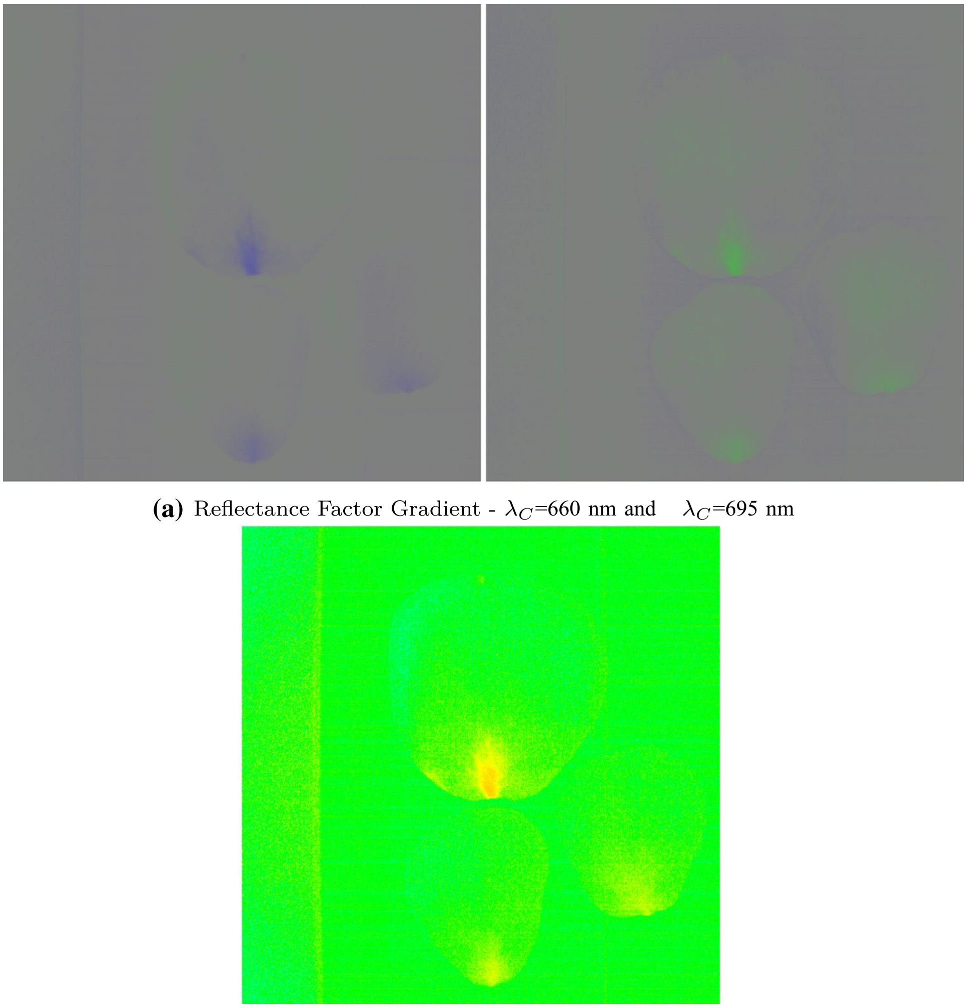


Fig. 13 Analysis

(a) Reflectance Factor Gradient - λ_C=660 nm and λ_C=695 nm

(b) Reflectance Factor Gradient - Tangent Angle - λ_C=675 nm

2019 as a proof of concept, has evolved through a new scientific application that we have developed with a domain expert. This collaboration allowed us to set up a methodology requiring the presence of 2 experts to guide us in producing new functionalities relevant for this new scientific application. It will allow us to produce a specific tool with custom shaders and predefined parameters based on a simplified, customized and intuitive interface dedicated for some scientific communities. We are now ready to reiterate this process for other scientific applications than the one studied in this paper where spectral imaging is interesting (dermatology, culture heritage, etc.) and thus to continue enriching our visualization tool.

Currently built to allow interaction with only one image per web page we also want it to be able to use several images simultaneously within the same web page. This last point is essential for its transformation into a more complete tool for visualizing these images in a 3D environment accessible via augmented reality devices (smartphones and tablets), mixed reality devices (Magic Leap One, Hololens 2) and virtual reality headsets thought this will call to multi-angle measurements and relighting.

Funding Funded by the *Fondation de l'Université Jean Monnet de Saint-Etienne*.

Declarations

Conflicts of interest The authors declare that they have no conflict of interest.

References

1. Multispectral image formats, cie 223:2017, isbn: 978-3-902842-10-7, cie division 8. http://www.cie.co.at/publications/multispectral-image-formats
2. Agarwal A, El-Ghazawi T, El-Askary H, Le-Moigne J. Efficient hierarchical-pca dimension reduction for hyperspectral imagery. 2007; p. 353–6.

3. Akhtar N, Ajmal M. Hyperspectral recovery from rgb images using gaussian processes. IEEE Trans Pattern Anal Mach Intell. 2018.
4. Akima H. A new method of interpolation and smooth curve fitting based on local procedures. J ACM. 1970;17(4):589–602.
5. Amankwah A, Aldrich C. A spatial information measure method for hyperspectral image visualization. In: 2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), 2015; p. 4542–5.
6. Behmann J, Acebron K, Emin D, Bennertz S, Matsubara S, Thomas S, Bohnenkamp D, Kuska MT, Jussila J, Salo HAMA, Rascher U. Specim iq: Evaluation of a new, miniaturized handheld hyperspectral camera and its application for plant phenotyping and disease detection. Sensors 2018;18(2):441.
7. Blerot B, Baudino S, Prunier C, Demarne F, Toulemonde B, Caissard J. Botany, agronomy and biotechnology of pelargonium used for essential oil production. Phytochem Rev. 2015;15:935–60.
8. Brauers J, Schulte N, Aach T. Multispectral filter-wheel cameras: Geometric distortion model and compensation algorithms. IEEE Trans Image Process. 2008;17(12):2368–80.
9. Bühler J, Rishmawi L, Pflugfelder D, Huber G, Scharr H, Hülskamp M, Koornneef M, Schurr U, Jahnke S. phenovein-a tool for leaf vein segmentation and analysis. Plant Physiol. 2015;169(4):2359–70. https://doi.org/10.1104/pp.15.00974.
10. Buschmann C, Lenk S, Lichtenthaler HK. Reflectance spectra and images of green leaves with different tissue structure and chlorophyll content. Israel J Plant Sci. 2012;60(1–2):49–64. https://doi.org/10.1560/IJPS.60.1-2.49.
11. Choudhary B, KumarSinha N, Shanker P. Pyramid method in image processing. J Inf Syst Commun. 2012;3(1):269–73.
12. Colantoni P, Thomas JB. A color management process for real time color reconstruction of multispectral images. In: Salberg AB, Hardeberg JY, Jenssen R, editors. Image Analysis. Berlin: Springer; 2009. p. 128–37.
13. Colantoni P, Thomas JB, Hardeberg JY. High-end colorimetric display characterization using an adaptive training set. J Soc Inf Disp. 2011;19(8):520–30. https://doi.org/10.1889/JSID19.8.520.
14. Colantoni P, Thomas JB, Hebert M, Trémeau A. An online tool for displaying and processing spectral reflectance images. In: 2019 15th International Conference on Signal-Image Technology Internet-Based Systems (SITIS), 2019. p. 725–31.
15. Colantoni P, Thomas JB, Pillay R. Graph-based 3d visualization of color content in paintings. In: A. Artusi, M. Joly, G. Lucet, D. Pitzalis, A. Ribes (eds.) VAST: International Symposium on Virtual Reality, Archaeology and Intelligent Cultural Heritage - Short and Project Papers. The Eurographics Association. 2010. https://doi.org/10.2312/PE/VAST/VAST10S/025-030.
16. Colantoni P, Thomas JB, Trémeau A, Hardeberg JY. Web technologies enable agile color management. In: 2019 15th International Conference on Signal-Image Technology Internet-Based Systems (SITIS), 2019. p. 303–10.
17. Colantoni P, Trèmeau A. Web browsers colorimetric characterization. 2019. p. 145–61.
18. Cucci C, Delaney JK, Picollo M. Reflectance hyperspectral imaging for investigation of works of art: Old master paintings and illuminated manuscripts. Acc Chem Res. 2016;49:2070–9.
19. Feng L, Zhang Y, Li M, Zheng Y, Shen W, Jiang L. The structural color of red rose petals and their duplicates. Langmuir. 2010;26(18):14885–8. https://doi.org/10.1021/la102406u.
20. Gkikas D, Argiropoulos A, Rhizopoulou S. Epidermal focusing of light and modelling of reflectance in floral-petals with conically shaped epidermal cells. Flora - Morphol Distrib Funct Ecol Plants. 2015;212:38–45. https://doi.org/10.1016/j.flora.2015.02.005http://www.sciencedirect.com/science/article/pii/S0367253015000110.
21. Hardeberg J, Schmitt F, Brettel H. Multispectral color image capture using a liquid crystal tunable filter. Opt Eng. 2002;41(10):2532–48.
22. Imai FH, Rosen MR, Berns RS. Comparative study of metrics for spectral match quality. 2002. p. 492–6.
23. Khan HA, Thomas JB, Hardeberg JY, Laligant O. Spectral adaptation transform for multispectral constancy. J Imaging Sci Technol 2018;**62**(2), 20504–1–12.
24. Khan HA, Thomas JB, Hardeberg JY, Laligant O. Multispectral camera as spatio-spectrophotometer under uncontrolled illumination. Opt Express. 2019;27(2):1051–70. https://doi.org/10.1364/OE.27.001051.
25. Kim SJ, Zhuo S, Deng F, Fu C, Brown M. Interactive visualization of hyperspectral images of historical documents. IEEE Trans Visual Comput Graph. 2010;16(6):1441–8.
26. van der Kooi CJ, Elzenga JTM, Staal M, Stavenga DG. How to colour a flower: on the optical principles of flower coloration. Proc R Soc B: Biol Sci. 2016;283(1830):20160429. https://doi.org/10.1098/rspb.2016.0429.
27. Kotwal K, Chaudhuri S. Visualization of hyperspectral images using bilateral filtering. IEEE Trans Geosci Remote Sens. 2010;48(5):2308–16.
28. Lapray PJ, Wang X, Thomas JB, Gouton P. Multispectral filter arrays: recent advances and practical implementation. Sensors. 2014;14(11):21626–59.
29. Le Moan SAM, Voisin Y, Hardeberg JY. A constrained band selection method based on information measures for spectral image color visualization. IEEE Trans Geosci Remote Sens. 2011;49(12):5104–15.
30. Liao D, Chen S, Qian Y. Visualization of hyperspectral images using moving least squares. In: 2018 24th International Conference on Pattern Recognition (ICPR), p. 2851–6.
31. Liao, D., Qian, Y., Zhou, J.: Visualization of hyperspectral imaging data based on manifold alignment. In: 22nd International Conference on Pattern Recognition, 2014. p. 70–75.
32. Liu D, Wang L, Atli Benediktsson J. Interactive multi-image colour visualization for hyperspectral imagery. Int J Remote Sens. 2017;38(4):1062–82. https://doi.org/10.1080/01431161.2016.1277041.
33. Martinez K, Cupitt J, Perrya S. High resolution colorimetric image browsing on the web. Comput Netw ISDN Syst. 1998;30:399–405.
34. Mokrzycki W, Tatol M. Color difference delta e-a survey. Mach Graph Vis. 2011;20:383–411.
35. Nguyen RMH, Prasad DK, Brown MS. Training-based spectral reconstruction from a single rgb image, lecture notes in computer science. 2014. p. 186–201.
36. Pardo A, Gutiérrez-Gutiérrez JAAHJM, Conde OM. Context-free hyperspectral image enhancement for wide-field optical biomarker visualization. Biomed Opt Express. 2020;11(1):133–48.
37. Polder G, van der Heijden GW. Visualization of spectral image. In: Censor Y, Ding M, editors. Visualization and Optimization Techniques, vol. 4553. SPIE: International Society for Optics and Photonics; 2001. p. 132–7.
38. Pyke KA, Page AM. Plastid ontogeny during petal development in arabidopsis. Plant Physiology. 1998;116(2):797–803. https://doi.org/10.1104/pp.116.2.797.
39. Ribés A, Brettel H, Schmitt FJM, Liang H, Cupitt J, Saunders D. Color and multispectral imaging with the crisatel multispectral system. 2003. p. 215–219.
40. RoBler F, Botchen RP, Ertl T. Dynamic shader generation for flexible multi-volume visualization. 2008. p. 17–24.
41. Sánchez Sorzano C, Vargas J, Pascual-Montano A. A survey of dimensionality reduction techniques. 2014.
42. Schockling M, Bonce R, Gutierrez, A, Robila SA. Visualization of hyperspectral images. In: S.S. Shen, P.E. Lewis (eds.) Algorithms

and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XV, vol. 7334, p. 715–26. SPIE. 2009.

43. Su H, Du Q, Du P. Hyperspectral image visualization using band selection. IEEE J Select Top Appl Earth Obser Remote Sens. 2014;7(6):2647–58.
44. Thomas JB. Multispectral imaging for computer vision. In: Habilitation à diriger des recherches. Université de Bourgogne, Franche-Comté. 2018. http://jbthomas.org/Thesis/2018HDRThesisCompactVersion.pdf
45. Thomas JB, Hardeberg JY. How to look at spectral images? a tentative use of metameric black for spectral image visualisation. In: J.B. Thomas, G.C. Guarnera, S. George, P. Nussbaum, S.A. Amirshahi, V. Kitanovski (eds.) Proceedings of the 10th Colour and Visual Computing Symposium 2020 (CVCS 2020), 2020;2688, p. 1–11.
46. Thomas JB, Hardeberg JY, Foucherot I, Gouton P. The PLVC display color characterization model revisited. Color Res Appl. 2008;33(6):449–60. https://doi.org/10.1002/col.20447.
47. Wazir ZK, Ejaz A, Saqib H, Ibrar Y, Arif A. Edge computing: A survey. Fut Gen Comput Syst. 2019;97:219–35. https://doi.org/10.1016/j.future.2019.02.050. http://www.sciencedirect.com/science/article/pii/S0167739X18319903.
48. Wilts B, Rudall P, Moyroud E, Gregory T, Ogawa Y, Vignolini S, Steiner U, Glover B. Ultrastructure and optics of the prism-like petal epidermal cells of eschscholzia californica (california poppy). New Phytol. 2018;219. https://doi.org/10.1111/nph.15229.
49. Xudong K, Puhong D, Shutao L. Hyperspectral image visualization with edge-preserving filtering and principal component analysis. Inf Fusion. 2020;57:130–43.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.