

Color data visualization for color imaging

Alain TRÉMEAU, Philippe COLANTONI
Laboratoire LIGIV EA 3070 – Université Jean Monnet
Bat. E – 10 rue Barrouin – 42000 Saint-Étienne – FRANCE
E-Mail: {tremeau|colantoni}@ligiv.org

ABSTRACT

There are several ways to display color data of a color image. In this paper we present different methods that we have developed in order to understand and to analyze color image information. These methods use traditional 2D and 3D visualization model associated with specific color transformation. We also introduce a new multidimensional visualization model usefull to analyse spatiocolorimetric data.

1. INTRODUCTION

The aim of this study is to propose a set of tools based on 2D and 3D visualization modes to display color information in different color spaces and different color representations, first in order to analyze which color space is the most relevant in regards to a computer vision process and a given image, and next to test in real-time different algorithms to select the most relevant algorithms, or to understand why an algorithm or a color space gives better results than another, in regards to a given process (segmentation, quantization, indexing,...) and a given image.

The methods used in our color image visualization tool have been developed to: (1) 2D image visualization; color images are then considered as 2D matrix, (2) 3D visualization; color images are then considered as 3D vectors, (3) 5D visualization, color images are then considered as 5D vectors, (4) chromatic motion analysis; colorimetric evolution between color images is then studied.

All these visualization methods can be used with one of the 19 color spaces available in our software. In order to visualize color data of a color image, we have developed different tools*, using :

- a viewer. The main window is an image reader/viewer. It can be used to read an image file using: a command line, a “File/Open” menu or a drag and drop process from the file explorer of Windows. Dealing with color accuracy, this viewer has been designed to load and display color images using 16 bits per channel coding.

A selection mechanism using color transparency has been included in the image viewer. This allows the user to select different regions of interest (RoI) in the image. At each of these RoIs corresponds one transparency color. This selection process has been chosen in order to build non-connected pixels set.

- a window to display colors in different spaces of representation. The user can select:
 - either a color space (3D) representation. The main aim of this toolbox is to propose an interactive method to visualize the 3D dimensional aspect of the color information. For this purpose we have built a specific interface adapted for color space visualization.
 - either a color image (2D) representation. This 2D color representation enables the user to display the image conversion content. Each channel is mapped respectively to the red, green and blue channels of the displayed image. These channels can be displayed separately using different look up tables (LUT).

*Most of these tools can be accessed at address <http://www.couleur.org>

- a window to select a color space. The color space converter window is a selector which includes three menus associated to different methods to display color data images. The main component of the window is dedicated to the color space selection and configuration. All color transform computations done in this software use a floating point color data internal representation.

This window is the control center of this software; it has been designed to launch the different windows used to display the color information: (a) either within an image (2D) where each color channels of the original color space are associated with the red, green and blue channels of the displayed image, (b) either within a 3D viewer where colors are represented by their 3D coordinates.

For color spaces using XYZ as intermediate color space, the primaries, the XYZ offset and the reference white/illuminant can be changed directly or by using some predefined values.

- a 3D navigation interface. The 3D navigation interface uses a virtual camera which allows the user a 3D environment interaction using the mouse or a 3D camera window. The 3D camera window can be used to define precisely the 3D point of view (position of the camera and its target in cartesian coordinates or in spherical coordinates, field of view of the virtual camera, visualization mode: perspective or orthographic projection).

Several visualization options have been included in order to customize the display: 3D primitive selection (sphere, cube, tetrahedron...), primitive display mode: point, wireframe, flat, smooth, CIE standard observer visualization: CIE 31 and/or CIE 64 (for color spaces that use XYZ as intermediate color space).

According the options used in the color space selection window the user will be able to display: (a) either a direct 3D representation of all selected colors; (b) either the 3D histogram of colors; 3D primitive of different sizes are then used to represent the color frequency.

This software uses: (a) the Gtk+ library as graphic user interface (GUI), this library originally developed for X-Windows graphic interface is available on Windows systems. By using Gtk+ (<http://www.gtk.org>), we can build, without code modification, UNIX and Windows versions of this software. (b) the ImageMagick library, this library has been used to read image files (image with 16 bits by channel are accepted). This library can read a tremendous variety of file formats (<http://www.imagemagick.org>). (c) the OpenGL toolbox, this toolbox has been used to display the 3D color information (<http://www.opengl.org>)

The main characteristic of this tool is its interactivity, since the user can select in real-time multiple windows to display different data, or multiple windows for same data (different representations). An important feature of "ColorSpace" is its ability to save its state, allowing a user to save at each moment his current session (windows contents, position, 3D orientation and position, visualization mode...). This feature is very useful for tutorial and demonstration purpose (offline and online). In order to show the online ability of this software we have integrated it to the ICobra system[†].

2. COMPUTATION OF COLOR DATA

2.1. Usual color spaces

In a general way, we consider that there is five families of color spaces: (a) Primaries color systems RGB , CMY , XYZ (xyY), LMS ; (b) Luminance-chrominance color systems YUV , YIQ , YC_bC_r , HSV , HSI , LHS ; (c) Perceptual color systems UVW , $L^*a^*b^*$, $L^*u^*v^*$, LHC , $L\alpha\beta$; (d) Usual non-correlated color systems $X_1X_2X_3$, $I_1I_2I_3$; and (e) Hybrid color spaces.¹

The reasons for applying color spaces transformations are very varied. The choice of an appropriate color space can be an important factor determining the results of an algorithm on a color image (e.g. the quality of image segmentation, compression ratio, etc.). In practice there is no ideal color space for all image processing applications. The decision on which color space to use depends on the processing task. An optimal decision can be very hard to find. However, knowledge of the properties of the various color space makes the choice easier.

[†]ICobra (Content-based image retrieval system), developed in our laboratory, can be accessed at address <http://www.ligiv.org/icobra/visualexec.php>

Some of the properties of the various transformations between *RGB* and other color spaces are summarized in table 1: ‡

Color space	Linearity of transformation	Stability of calculations	Perceptual uniformity
<i>rgb</i>	No	No	No
<i>XYZ</i>	Yes	Yes	No
<i>xyz</i>	No	No	No
$L^*a^*b^*$, $L^*u^*v^*$	No	Yes	Yes
YUV , YIQ , YC_bC_r	Yes	Yes	No
AC_1C_2 , Opponent	Yes	Yes	No
$X_1X_2X_3$, $I_1I_2I_3$	Yes	Yes	No
HSI , Perceptual	No	No	No
Munsell	No	Yes	Yes

Table 1: Properties of color transformations between *RGB* and other color spaces.

2.2. Usual Non-correlated color systems

In color imaging, the basic problem of usual color spaces is that: (a) color components are highly correlated the ones to the others and mainly to the luminance component; (b) most of color components, except luminance component, have small discriminant power (i.e. a small variance). Consequently, they are not adapted to separate (to discriminate) color clusters in color space. For these reasons the $X_1X_2X_3$ color space has been developed to derive color features with large discriminant power from the Karhunen-Loeve transformation (see Figure 1).



Figure 1. (a) Image *Lenna*. (b) *RGB* color distribution of *Lenna*. (c) *Lenna* projected on the X_1 first principal component. (d) *Lenna* projected on the X_2 component. (e) *Lenna* projected on the X_3 component. (Nb. Even if *Lenna* is a color image, it can be seen as a “black and white image” because most of its colors can be represented by the first axis.)

Let be \mathbf{P} the covariance matrix of color components P_1 , P_2 , and P_3 taken into account. The covariance matrix of $P_1P_2P_3$ color distribution can be defined by :

$$\mathbf{P} = \begin{bmatrix} \text{var}(P_1) & \text{cov}(P_1, P_2) & \text{cov}(P_1, P_3) \\ \text{cov}(P_1, P_2) & \text{var}(P_2) & \text{cov}(P_2, P_3) \\ \text{cov}(P_1, P_3) & \text{cov}(P_2, P_3) & \text{var}(P_3) \end{bmatrix}$$

where

$$\mathbf{P}(i, j) = 1/N \sum_{k=1}^N (c_k^i - \bar{c}_i)(c_k^j - \bar{c}_j) \quad \text{if } i \neq j \quad \text{else } \mathbf{P}(i, i) = 1/N \sum_{k=1}^N (c_k^i - \bar{c}_i)^2$$

‡N.b. Non-linear transformations like hue, saturation, and normalized color have non-removable singularities, near which a small perturbation of input R , G , and B can cause a large jump in the transformed values. Moreover, the distribution of the non-linearly transformed values can show spurious modes and gaps. By these reasons and from the computational point of view, linear transformation such as Y , I , and Q would be more preferable than nonlinear ones.

represent, respectively, the covariance of P_i component with P_j component, and the variance of P_i component, and

$$\bar{c}_i = 1/N \sum_{k=1}^N c_k^i$$

represents the mean value of P_i component.

The Karhunen-Loeve transformation used is defined by the following steps: (1) a calculation of eigenvalues $\lambda_1, \lambda_2, \lambda_3$ and eigenvectors of the covariance matrix \mathbf{P} , such as $\lambda_1 > \lambda_2 > \lambda_3$, (2) a calculation of the first principal axis from the eigenvector $(\omega_1^1, \omega_1^2, \omega_1^3)$ corresponding to the eigenvalue λ_1 ,[§] (3) a calculation of the second principal axis from the eigenvector $(\omega_2^1, \omega_2^2, \omega_2^3)$ corresponding to the eigenvalue λ_2 . This second axis must be orthogonal to the first one,[¶] (4) a calculation of the third principal axis from the eigenvector $(\omega_3^1, \omega_3^2, \omega_3^3)$ corresponding to the eigenvalue λ_3 . This latter axis must be orthogonal to the two first ones.

Then, we have :

$$\begin{aligned} X_1 &= \omega_1^1 P_1 + \omega_1^2 P_2 + \omega_1^3 P_3 \\ X_2 &= \omega_2^1 P_1 + \omega_2^2 P_2 + \omega_2^3 P_3 \\ X_3 &= \omega_3^1 P_1 + \omega_3^2 P_2 + \omega_3^3 P_3 \end{aligned}$$

2.3. Hybrid color systems

The basic idea of hybrid color spaces is to combine either adaptively, either interactively, different color components from different color spaces to: (a) increase the effectiveness of color components to discriminate color data, and (b) reduce rate of correlation between color components.¹

It is established that we can all the more reduce, from K to 3, the number of color dimensions that: (a) most of color spaces are linked the ones to the others, either by linear transformations or by non-linear transformations, and (b) all color spaces are defined by a 3 dimensional system.

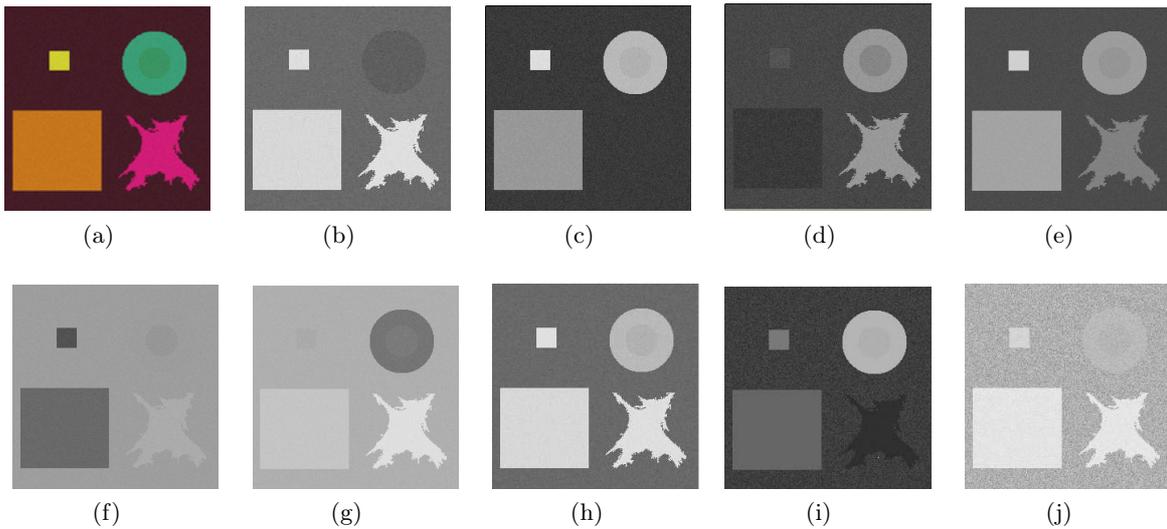


Figure 2. (a) *RGB* Color image, made of 6 regions (Brown, Orange, Yellow, Pink, Green and Dark Green), projected on different color components. (b), (c), (d) *R, G, B* projections. Among the three *R, G, B* color components, at most 3 regions can be identified with the component *G*. (e), (f), (g) *Y, C_b, C_r* projections. Among the three *Y, C_b, C_r* color components, at most 3 regions can be identified with the component *Y*. (h), (i), (j) *H, S, V* projections. Among the three *H, S, V* color components, at most 3 regions can be identified with the component *H*. In combining *G, Y, H* color components, all regions can be identified.

[§]For most of natural images, this first axis corresponds to luminance component.

[¶]For most of images, color distribution can be represented by only the two first features.

Considering that there is a high redundancy between colors components it is, in a generay way, quite difficult to define criteria of analysis to compute automatically the most relevant color components corresponding to a selected set of color components. That is the reason why, in order to build a hybrid color space, based on K' color components, from K selected color components, such as $K' \ll K$ (see Figure 2), we propose the following method: (1) select K color components, by using a specific interface which enables the user to weight each selected color components, and build the corresponding image of dimension K , (2) compute the covariance matrix (of size $K \times K$) of K color components selected, (3) compute the eigenvectors and the eigenvalues of this matrix, (4) reduce to K' the number of color components in computing the K' most significant eigenvalues of the covariance matrix from a principal component analysis (PCA).

Next, the three first principal components computed (i.e. the hybrid colors components) are used to compute the 3D representation which best characterizes the image studied (see Figure 3).

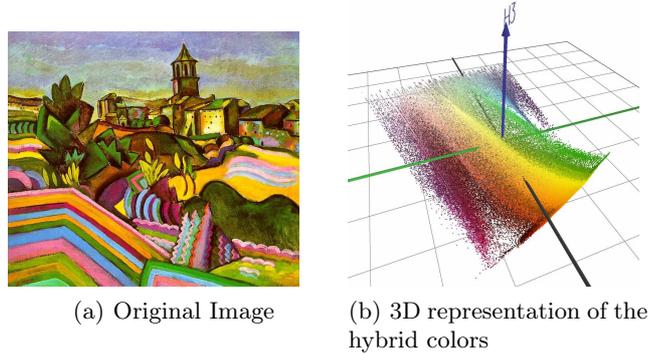


Figure 3: Hybrid color space visualization

3. 2D VISUALIZATION

Rather than presenting the principles of 2D visualization methods, which are quite common, it seems to us more interesting to insist on the fact that, in our case of study, data to visualize are coded with floating numbers in order to ensure both stability and precision of calculations.

Color data conversion process is done during the display process; it uses 2 buffers, one for floating number representation and another for the display system.

When we transform a image from one color space to an another one, we are faced to the problem of how visualizing it in pseudo-colors on a RGB color display. Knowing the limits and the characteristics of the color space used, we propose a visualizing process which uses these limits and adjusts, without equalization of dynamics, the dynamic of each color component to the dynamic of R , G and B channels of the display system (see Fig. 4).

Rather than displaying a 3D color image in falses colors with a RGB Look Up Table (LUT), the user can also choose to display separately each 1D image of each color component with a pre-selected LUT such as spectral, binary, red, green, blue, or black and white LUT.

4. 3D VISUALIZATION

Color spaces are commonly defined by 3 dimensional systems, so they are naturally well adapted to 3D visualization. Thanks to the latest generation of graphic cards currently available, the number of data to process when we work with images of size lesser than 512×512 enable us to visualize color data in 3D spaces in real time. For images with greater size, to reach real time it is necessary to use more powerfull graphic cards.



Figure 4. (a) Image *Lenna* represented in *RGB* coordinates (i.e. original image). (b) *Lenna*, represented in $L^*a^*b^*$ coordinates, digitized with 8 bits by component and equalization of dynamics, visualized with a “false colors” look up table. (c), (d), (e) L^* , a^* , b^* color components of image (b), respectively. (f) *Lenna*, represented in $L^*a^*b^*$ coordinates, digitized with 8 bits by components without equalization of dynamics, visualized with a “false colors” look up table. (g), (h), (i) L^* , a^* , b^* color components of image (f), respectively. (N.b. $L^*a^*b^*$ images can not be directly visualized on a 8 bit display. The interest of using such a representation is for example to help the observer to analyse if colors (regions) are globally or separately better segmented in this color space than in other color space such as *RGB*.)

4.1. Color space representation

One aim of this work was to propose a new visualization software as flexible and interactive as possible for users. So, with this software, the user has the possibility to visualize either all colors of an image, or only colors of Regions of Interest (RoI), either with a point primitive or with a 3D geometric primitives (sphere, cube, tetrahedron, ...), with either or not virtual lighting. This latter characteristic is very useful for users using visualization system based on stereovision.

In order to reduce the number of colors to visualize, we use a hash table. This technic is well adapted to reduce significantly the number of colors to visualize due to the fact that for natural images the number of colors is quite limited.

4.2. Color histogram representation

Color histogram representation is all the more useful in image processing that histogram enables to reduce (i.e. to quantify) the numbers of color representatives of an image meanwhile preserving the frequency of occurrence of each color. It is quite difficult for a user to distinguish, only with a 3D color representation, color clusters of high density of color clusters of low density, especially when clusters are compound of numerous points, i.e. when clusters have a high density.

This notion of density is absolutely necessary to compute color histogram from color representation. In order to compute and to visualize color histogram we propose to use the following method: (1) split the color space in $I \times J \times K$ cells, next put to zero the value of each cell. (2) for each pixel (x, y) of the image studied, or of each Region of Interest selected, find the cell corresponding to the color of this pixel, next increment the value of this cell. (3) compute the list of cells non empty, next to each cell of this list associate to it a value proportionnal to the frequency of occurrence of colors of this cell. (4) associate at each cell a geometric primitive with a size proportionnal to the value of the cell.

The main problem we have when we compute a color histogram is due to the fact that most of histogram cells have a frequency of occurrence very small. So, for a *RGB* color image of resolution $n \times m$ the smallest probability to have a cell with no occurrence is $1 - \frac{n \times m}{256^3}$. The main difficulty we have to face, when we want

to pre-quantify a color histogram from X bits (i.e. 24 bits for a RGB histogram) to Y bits (i.e. 15 bits for a RGB histogram in removing the 3 least significant bits of each color component), consists to find a compromise which minimizes both the number of empty cells and the number of merging cells between adjacent cells of high density. In order to solve this problem, we propose that the user adjusts himself interactively the number of cells necessary to visualize a color histogram.

4.3. Projection

The user has two reasons to display color projections: (1) in order to increase the relief effect. The human visual system will consider the projection as a shadow. That shadow will be used by the brain to build the 3D representation of the 2D image. (2) in order to visualize the colors density. For this purpose, we use transparency effect which allows us to modulate the projected colors when multiple points are projected at the same place.

User can modify projection properties such as: projection planes (multiple projections planes are allowed), projection color and projection transparency.

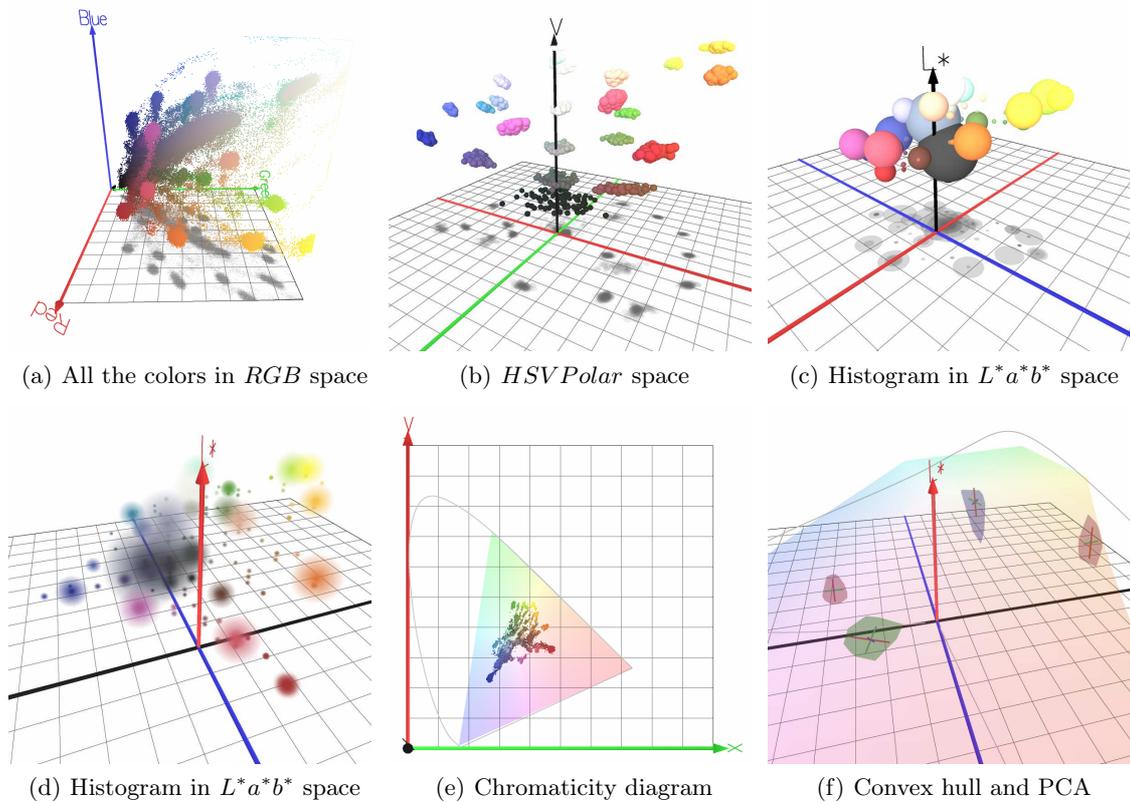


Figure 5: 3D color visualization

4.4. Color gammut, convex hull, PCA and CIE standard observers

In some cases of study it is very usefull to have some visualization tools which enable to better understand or better analyse how color clusters or color data are distributed in one color space or in another one color space. That is the reason why the 3D visualization system that we have developed enables to display:

- the color gammut corresponding to the limits of each color space. These limits are derived from the limits of the RGB color cube. This tool enables the user to see how the color gammut of a device (i.e. a camera or a display) is distributed relatively to the color gammut of the color space selected (see Figures 5(e), 5(f)).

- the principal component axes (PCA) of each color cluster selected and their first statistical moments. This tool enables the user to find the most relevant color components to discriminate the color clusters selected (see Figure 5(f)).
- display color data, as the Human Visual System (HSV) perceive colors, relatively to CIE XYZ 1931 and CIE XYZ 1964 standard observers. These standards are usefull when we use color spaces based on XYZ as intermediate color space.

5. CHROMATIC MOTION VISUALIZATION

In order to see the chromatic evolution of an image versus different lighting conditions or versus different grabbing conditions (i.e. from on sensor to an another one), we have added a specific visualization toolbox. This toolbox allows the user to follow color change in a stack of images from three methods:

- per color motion. This method allows to display the chromatic change of a single pixel or a group of pixels. We get one segment between each image; the final result is for each pixel a 3D chromatic path.
- per region motion. This method allows to display the chromatic motion of images regions. Rather than using the previous method which may display too much information, and in order to get a synthetic view of the chromatic motion, we propose to use the chromatic mean of each region (see Figure 6(b)).
- histogram motion. This method can be also used to have a synthesis of the chromatic motions from different versions of an image. In order to display the chromatic paths corresponding to this method we compute in a first time the image histogram then we determine the chromatic behavior of the corresponding regions in the images stack (see Figure 6(c)).

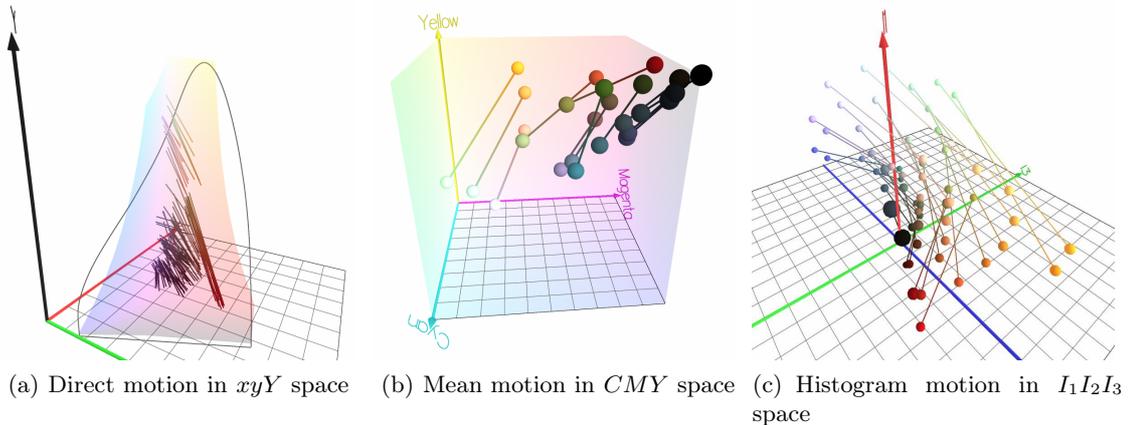


Figure 6: Chromatic motion visualization

6. SPATIOCOLORIMETRIC DATA VISUALIZATION

As we saw before, different methods can be used to visualize color information (2D and 3D representation). We propose to introduce in this section a new visualization method based on multi-dimensional (i.e. spatio-color) aspect of color.

Color image information can be represented by a 5D vector including color and spatial information; 5D vectors are required to display this color image information.

6.1. Transformation pipeline

This method is an extension of a previous work^{2,3} developed for 4 dimensional objects. The main principle of this technique is the following: if the observer of a high-dimensional environment does not know how to apprehend high-dimensional objects, then the user will experience the geometry of a normal 3D environment.

This principle is implemented by using observer position and orientation to determine a 4D cross-section or projection of the high-dimensional environment, then we use this new 4D dimensional environment in order to determine a 3D cross-section or projection giving us a 3D representation of our 5D spatiochromimetric information.

In fact the technic that we have developed enables to use 4 different display methods: (1) 5D \rightarrow cross-section \rightarrow 4D \rightarrow cross-section \rightarrow 3D, (2) 5D \rightarrow cross-section \rightarrow 4D \rightarrow projection \rightarrow 3D, (3) 5D \rightarrow projection \rightarrow 4D \rightarrow cross-section \rightarrow 3D, (4) 5D \rightarrow projection \rightarrow 4D \rightarrow projection \rightarrow 3D The cross-section process requires that point have thickness, thus we will consider 5D points as 5D spheres with a non null radius.

6.2. 5D to 4D transformations

The first step produces a set of 4D spheres. Depending the method used we get as much 4D spheres as 5D initial points (projection) or less (cross-section).

Observer position is represented by a 5D vector $p = [p_x, p_y, p_z, p_w, p_v]^T$. Observer orientation is represented by a vector q of unit length or, equivalently, by a 5×5 rotation matrix Q that describes the rotation of some canonical orientation vector q' into the current orientation q . The canonical orientation vector is simply a reference vector which may, for instance, correspond to an initial orientation for the observer such as $q' = [0, 0, 0, -1, 0]^T$.

6.2.1. Observer-dependent 5D to 4D cross section

The computation of the 4D cross-section determined by the observer's position and orientation is best conceived in terms of the axis which is perpendicular to the 4D cross-section. This "hidden axis" changes relatively to observer motion. It is readily computed using the matrix Q that describes the observer's current orientation. In particular, if the coordinates of the unit vector h' along the hidden axis are known when the observer is in the canonical orientation q' , then the matrix Q applied to h' produces the current hidden axis h : $h = Q.h'$.

The current value of the observer along the hidden axis h determines which cross-section is taken perpendicular to h . The observer's position g along the hidden axis h is given by: $g = p.h$.

The idea, then, is to consider only the 4D subspace which lies perpendicular to the current hidden axis and that has the same position along the hidden axis as the observer.

Our method for determining 4D cross-section of 5D spheres involves checking its intersection with the 4D sub-space. In order to compute this cross-section we determine the radius of the corresponding 4D sphere given by: $t = v - p$ and $d = -h * t$, if $|d| < r$ then the point is visible in the 4D cross-section.

6.2.2. 5D to 4D projection

To project a 5D point v in a 4D sub-space, we use the following method: (1) $t = v - p$, (2) $d = -h.t + v$, (3) $t = d * t$, (4) $result = Q * (t - p)$, where t is a 5D vector; d a scalar value; $result$ the 4D result point.

6.3. 4D to 3D transformations

We use the same methods (cross-section or projection) to transform the 4D spheres resulting from the previous step to a set of 3D spheres.

6.4. Results

The figure 7(b) shows a multiple projection process of the 7(a) image. The figure 7(c) shows a more complex result: a cross-section process follows by a projection. In that case only a sub-set on the original image is visualized.

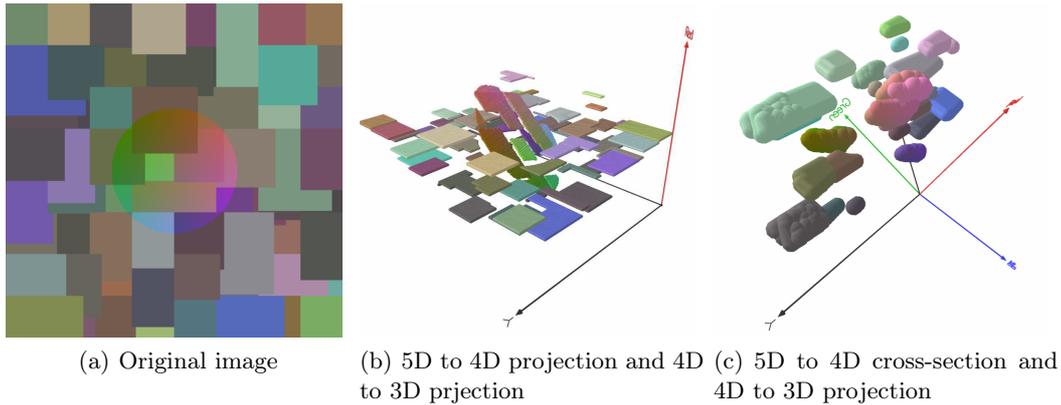


Figure 7: Chromatic motion visualization

7. CONCLUSION AND PERSPECTIVE

The main characteristic of this work has been to gather in one toolbox several color image analysis tools based on 3D visualization of color data. Most of these tools are devoted to 3D visualization of color data. The next step of this work will be to propose other tools more devoted to color imaging.

One aim, at most long term, will be to develop another toolbox devoted to color space quality assessment in order to assess the quality of image processing algorithms. This in order, for example, to optimize the adjustment of threshold values of algorithms in regards to the color space selected. Another aim will be to add another toolbox devoted to color constancy and chromatic change assessment. This in order for example to simulate changes of lighting conditions, i.e. to predict the color rendering of an image viewed under other spectral lighting condition than those used to grab it which, most of the time, is unknown.

We are waiting for OpenGL 2.0 version in order to introduce in our tool vertex and fragment programs. This will allow us to increase the rendering speed and the visual quality of our visualization tool.

REFERENCES

1. N. Vandenbrouke, L. Macaire, and J. G. Postaire, "Color pixels classification in a hybrid color space," in *Proceedings of IEEE*, pp. 176–180, 1998.
2. G. D. Seyranian, P. Colantoni, and M. D’Zmura, "Navigation in environments with four spatial dimensions," in *European Conference on Visual Perception*, (Trieste), 1999.
3. M. D’Zmura, P. Colantoni, and G. Seyranian, "Virtual environments with four or more spatial dimensions," *Presence* **9**, pp. 616–631, 2001.