14th European Signal Processing Conference (EUSIPCO 2006), Florence, Italy, September 4-8, 2006, copyright by EURASIP

ANALYSIS OF MULTISPECTRAL IMAGES OF PAINTINGS

Philippe Colantoni, Ruven Pillay, Christian Lahanier and Denis Pitzalis

Centre de Recherche et de Restauration des Musées de France
Palais du Louvre, Porte des Lions
14, quai Franois Mitterrand
75001 Paris - France
{ruven,christian.lahanier,denis.pitzalis}@c2rmf.cnrs.fr, colantoni@couleur.org

ABSTRACT

One hundred paintings conserved in several museums have been scanned by the C2RMF using the multi-spectral CRISATEL camera [3]. These high resolution images allow us to not only generate an accurate colour image under any chosen illuminant, but also allow us to reconstruct the reflectance spectra at each pixel. Such images can be used for a visual qualitative as well as measurement-based quantitative scientific analysis of the work of art.

Several image processing tools have been developped to allow us to perform these analyses. The IIPImage system [2] enables us to visualize high resolution multi-spectral 16 bit images, view image details in colour or for each spectral channel and to super-impose and compare different wavelengths.

A complementary viewing system uses an innovative 3D graphics hardware-accelerated viewer to allow us to reconstruct the resulting colour dynamically while interactively changing the light spectrum. The system also allows us to perform segmentation, view the colour distribution for a particular colour-space and perform dynamic spectral reconstruction.

1. INTRODUCTION

Since 1931, the C2RMF (Laboratoire de Recherche des Muses de France) has produced scientific photography using a variety of methods. For example, visible reflectance, raking light, ultra-violet fluorescence, infra-red reflectography, X ray radiography and emissiography as well as with new advanced techniques such as auto-radiography.

Over the last 15 years, attempts at multi-spectral imaging has been carried out. These started in 1990 with the VASARI camera [1]. This produced 7 channels of data, with the final image stitched together from a mosaic of low-resolution shots. The CRISATEL project [3] developed an new camera able to scan in one shot much larger paintings than previously possible. It generates 13 very high definition images of up to 12,000 x 20,000 pixels in size. Each image is made sequentially with a narrow 40 nm bandwidth in order to cover the whole of the visible spectrum and the near infra-red (from 380 to 1050 nm).

Multi-spectral image processing allows us to investigate a painting in ways that were impossible previously. The visualization of high resolution images made different wavelengths, for example, allows us to locate information at different levels of depth in the paint layer. Near the ultra-violet and blue, the image corresponds to the reflectance of the upper layers of the painting (the varnish first). Close to the red and into the infra-red, the paint layer is partly transparent and we are able to see the under-painting such as under-drawings. In fact, very high definition digital images contain more details that the artist would have been able to see. In addition, with a dynamic range of 16 bits dynamic range, the colour accuracy contains more colour levels than the human eye is able to identify. The digital content is more accurate than what the artist would have been able to perceive themselves and can gives subtle information on the painter's technique.

Applied to paintings, the visual examination through the paint layer, the colour analysis by image processing, and the pigment characterisation by means of their spectral reflectance curves provides new quantitive information for a complete study of a painting.

2. COLORIMETRIC MEASUREMENT

Our aim is to provide a customized image processing tool for art preservation and analysis using reflectance images (see figure 1). In this section we will describe all the colorimetric measurements and visualizations that will be used in our application.

Figure 1: *Reflectance visualization mode*

2.1 Image colour cloud and gamut

The colour cloud of an image in a specific colour space is a 3D representation where pixel values are used to define the 3D position for each colour. The colour gamut of an image

is defined by the convex hull of the colour cloud (see figure 2).

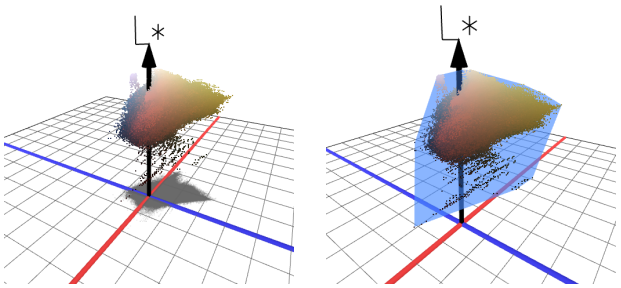

Figure 2: *Image colour cloud and gamut*

2.2 Colour management

In order to provide calibrated colour (colours which are similar to the real colour stimuli) on a specific display device we have to characterize this device.

Device characterization is based on measurements of input values (e.g. RGB input values of the display device) and output values (e.g. XYZ values measured on the display device by a spectrometer).

This characterization process is used to calibrate the colour previously generated using a virtual illuminant (see figure 3). We use a custom calibration process for this application which generates a 3D Look Up Table (LUT). This 3D LUT will be used to compute the *RGB* values needed to generated the closest colour stimuli corresponding to a $L^*a^*b^*$ value on a display device.

2.3 Colour segmentation

The segmentation process needed for this application is based on a colour clustering and thresholding approach. It can be time consuming, especially when we are using complex colour distance measures such as ΔE_{2000} and a large set of reference colours.

1. The first method associated a ΔE threshold to a reference colour. This generates a sphere in the $L^*a^*b^*$ colour space. All the colours in this sphere are associated with the same label (see figure 5).
2. The second method uses an algorithm which isolates and labels individual colours by finding in an set of reference colours the closest colour. This method provides a full segmentation of the image.

3. DESCRIPTION OF METHODS OF IMAGE PROCESSING USING GPU

3.1 Introduction

Image processing and analysis are fields which naturally require high performance and precision. This is particularly true in colour and multi-spectral image processing. Real time and high frame-rate digital image processing can only be obtained by using dedicated and expensive hardware. The latest 3D graphics cards, however, are not only more affordable, but include new features useful for image processing.

3.2 *XYZ* colour space

Colour is the perceptual result of light in the visible region of the spectrum, having wavelengths in the region of 380 nm to 780 nm. The human retina has three types of colour photoreceptor cells cone, which respond to incident radiation with somewhat different spectral response curves. Because there are exactly three types of colour photoreceptor, three numerical components are necessary and theoretically sufficient to describe a colour.

The *CIE* colour standard is based on imaginary primary colours *XYZ* i.e. which don't exist physically. These virtual primary colours have, however, been selected so that all colours which can be perceived by the human eye lie within this colour space.

The *XYZ* system is based on the response curves of the three colour receptors of the eye's. Since these differ slightly from one person to another person, *CIE* has defined a "standard observer" whose spectral response corresponds more or less to the average response of the population. This objectifies the colorimetric determination of colours.

In our case we have to determine for each pixel of the multispectral image the resulting *XYZ* value generated by the virtual light (see figure 3) interaction with the reflectance image using the following formulae:

$$\begin{cases} X = \int_\lambda x(\lambda).R(\lambda).L(\lambda).d\lambda \\ Y = \int_\lambda y(\lambda).R(\lambda).L(\lambda).d\lambda \\ Z = \int_\lambda z(\lambda).R(\lambda).L(\lambda).d\lambda \end{cases} \quad (1)$$

Where $R(\lambda)$ is the reflectance spectrum and $L(\lambda)$ is the light spectrum (used as illuminant).

In this application we introduced another feature allowing varnish removal using the transmittance properties of the varnish (see figure 3). The CRISATEL project produces 13 monochrome images which correspond to the following wavelengths: 400, 440, 480, 520, 560, 600, 640, 680, 720, 760, 800, 900 and 1000nm. Only the 10 first wavelengths interact with the visible part of the spectrum. Taking this into account, the previous formulae (equ. 1) become:

$$\begin{cases} X = \sum_{\lambda=400}^{\lambda=760} x(\lambda).R(\lambda).V(\lambda).L(\lambda) \\ Y = \sum_{\lambda=400}^{\lambda=760} y(\lambda).R(\lambda).V(\lambda).L(\lambda) \\ Z = \sum_{\lambda=400}^{\lambda=760} z(\lambda).R(\lambda).V(\lambda).L(\lambda) \end{cases} \quad (2)$$

Using these formulae we can compute the resulting *XYZ* values for each pixel of the source image. Of course, the obtained values are only an approximation of the values that will be measured in front of the original painting.

3.3 Programmability

Thus far, 3D graphics cards have been designed to offer high performances in two main areas:

- arithmetic calculations on vectors and matrices, which are essential for efficient computations of geometric transformations and colour manipulations. In our case, colours are represented by 3D or 4D vectors.
- access to different memory locations, such as the texture memory, the Z-buffer or the frame buffer, requiring large memory bandwidth.

Due to the rigidity of the graphics pipeline, it used to be the case that these capabilities could not be exploited for any other purpose than 3D rendering.

Today, the programmability provided by the new generation of GPU's allows us to alter the two main engines of the pipeline: the transforming and lighting engine (T&L) and the multi-texturing engine. Basically, the vertex processor allows us to replace the traditional transformation and lighting process by a custom geometrical manipulation program, called a "vertex program". A vertex program handles the 3D geometrical data (vertices) entering the graphic pipeline and outputs, for each vertex, the vertex itself and its parameters (colour, texture coordinates...). Further in the pipeline, the fragment processor, usually in charge of the multi-texturing task, can instead run any complex colour manipulation program, and thus processes each polygon per pixel.

This flexibility, especially at the multi-texturing stage, is essential since the goal is to use the fragment processor for image processing purposes.

This approach has already been partially explored by [4] within the scope of colour image segmentation based on basic programming methods. More recently, GPU computational capabilities have been used for other applications such as numerical computations [5] and simulations. We demonstrate in [6] that a graphics card can be suitable for colour image processing.

3.4 Image processing using the fragment processor

In addition to this programmability feature, these recent 3D graphics cards are particularly well suited to performing floating point computations, giving us high precision calculations. Moreover, off-screen rendering, can be done by the use of frame buffer objects, and mathematical functions implemented in hardware make the computation of complex expressions more efficient. The combination of these features allows us to think that this type of hardware is well adapted to run image processing algorithms.

However, some types of image processing algorithms cannot be implemented due to the fragment processor limitations. The first limitation is due to the fact that a fragment program can only give, for a given pixel, its colour and depth values. Secondly, vertex and fragment programs are restricted to a maximum number of instructions[1]: 65536 for vertex programs and 1024 for fragment programs. Moreover, they cannot contain more than 256 loops.

Finally, the parallel "architecture" of the fragment processor enables several pixels to be processed at the same time. Consequently, this optimizes the performances of the algorithms implemented but also prevents the processor to know or to monitor in which order pixels need to be processed. Therefore, the fragment processor is limited to per pixel processing. On the other hand, algorithms based on sequential scans of the image, where the processing of the current pixel needs the result obtained by the processing of a previous pixel (e.g. algorithms such as labelling or edge following segmentation) are totally unsuitable for the GPU and cannot (or with difficulty) be implemented with such a parallel architecture.

This severe limitation can be, to some extent, overcome. Indeed, as for 3D image rendering, a multi-pass approach can be used to process an image: 2 textures are needed, while the first contains the data to be processed, the other one receives the processed data. Once finished, a new process can be performed from the resulting image, then, the second texture receives in its turn the recently processed data, and so forth.

Consequently, all the processing algorithms needed in our application can be implemented using 3D graphics cards.

3.5 Implementation

As we explained before this application provides a full interface allowing us to interact in real time with an reflectance image (see figure 3). We will describe here only the processing part of this application. The framework, the graphic interface and the colour management process will not be detailed in this publication.

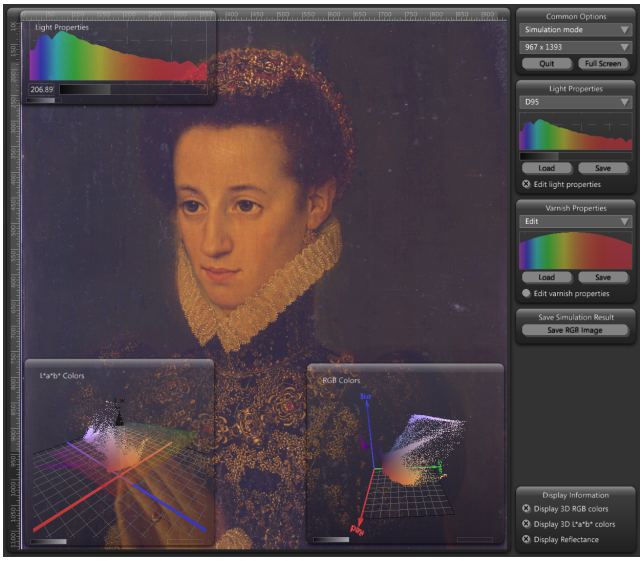

Figure 3: *Illuminant simulation mode*

Our fragment programs have been implemented and compiled using version 1.4 of CG [7] and have been executed with OpenGL 2.0 (graphics cards supporting shader model 3.0 must be used). This software program has been developed for multiple operating systems and has been tested under Windows XP, 64 bit Windows XP and Linux.

3.5.1 Reflectance images

In order to optimize the process we transformed the reflectance information into packed RGBA images (4 RGBA images which will contain each 4 planes). As mentioned above, the GPU is designed for vectorial computations, using these packed versions of the reflectances images allow us to optimize the computation of the XYZ values. These RGBA images are the input of all the process. They are uploaded once in the video memory of the graphics card.

3.5.2 Processing steps

The figure 4 shows all the processing steps that will be done in order to produce a segmented image as result. Each kernels (basic process) produce a 2D rectangular texture which will contain the result of the corresponding processing (reflectance to XYZ transform , XYZ to $L^*a^*b^*$ transform ...).

There are two stages in this global process:

[1] In our study we have used the *n*VIDIA^{TM} G70 processor

- The first stage produces a calibrated *RGB* image and a $L^*a^*b^*$ measured image.
- The second stage uses this $L^*a^*b^*$ image in order to generate a segmented image.

These two stages, which can be activated independently, are very fast. This allows us to interactively change parameters such as: light spectrum, varnish transmittance, reference colours number and references colours position...

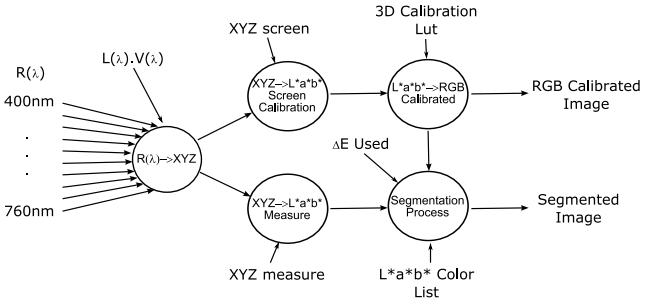

Figure 4: *Stream processing graph*

3.5.3 *Kernels*

In this section we will describe briefly all the kernels used in this application.

1. Reflectance images to *XYZ* image: The reflectance information (packed in RGBA 2D rectangular textures) and a 10 dimensions vector corresponding to the $L(\lambda).V(\lambda)$ members of the *XYZ* equation are used in order to compute for each fragment (pixel) the corresponding *XYZ* value. This kernel needs a lot memory bandwidth.
2. *XYZ* to $L^*a^*b^*$ kernel: Conversion between the *XYZ* and $L^*a^*b^*$ colour spaces necessitates the computation of non-linear mathematical expressions; that requires to use intensely hardware mathematical functions and involves on the other hand negligible memory transfers. In this process we use this kernel twice:
   - for computing the $L^*a^*b^*$ values used during the calibration process (the reference *XYZ* is the white value of the screen);
   - for computing the $L^*a^*b^*$ values used during the measure process (the reference *XYZ* is the white value of the selected illuminant).
3. $L^*a^*b^*$ to *RGB* kernel: This kernel provide a calibration process based on a 3D LUT previously generated. It allows to compute using a trilinear interpolation in this 3D LUT (a 3D texture in the memory of the graphics card) the corresponding *RGB* values.

The two last kernels are used in the segmentation process, both of them use as input the $L^*a^*b^*$ measured texture previously computed.

1. one reference colour segmentation: We compute for each fragment the distance to a reference colour. If this distance is smaller than a ΔE threshold then we label this fragment.
2. labels each fragment by finding in a set of reference colours the closest colours.

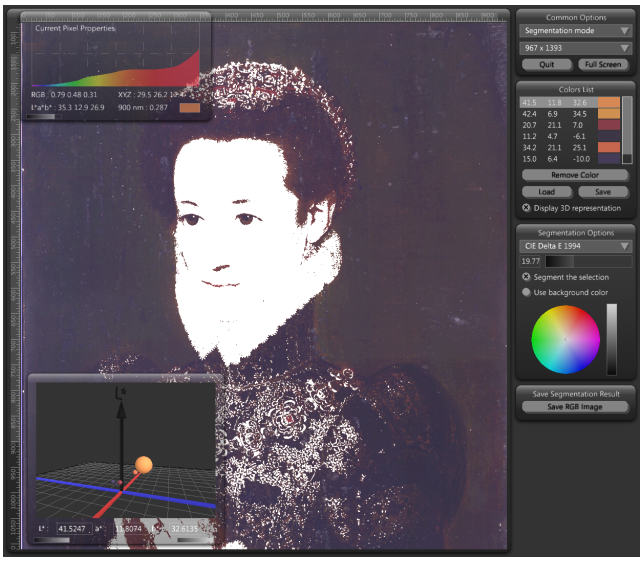

Figure 5: *Segmentation process using 1 colour*

3.6 Performance

The performance depends on the formats of the memory buffers used for textures and the number of tempory textures needed to complete the entire process pipeline (5 RGB textures in half float format, 16 bits per channel). All these basic processes correspond to per-pixel processing which requires the execution of several mathematical complex functions and vector calculations. Table 1 indicates that the GPU performs very well in this case study.

Colours	ΔE	Time	Frames/s	Pixels/s
1	1976	0.00189	526.9509	$899*10^6$
1	1994	0.00302	330.8694	$564*10^6$
1	2000	0.01083	92.27731	$157*10^6$
5	1976	0.0168	59.2515	$101.1*10^6$
5	1994	0.0274	36.3934	$62.1*10^6$
5	2000	0.0922	10.8448	$18.5*10^6$
10	1976	0.0326	30.6803	$52.3*10^6$
10	1994	0.0530	18.8393	$32.1*10^6$
10	2000	0.1827	5.4720	$9.3*10^6$

Table 1: Segmentation process

4. ANALYSIS OF THE "LADY PRAYING"

The multi-spectral images examination prove that this painting is, in fact, a part of a larger painting (Fig. 6). Two figures appear on its left side and on its upper part (Fig. 6e, 6f). The poplar panel support was cut on its left and upper sides for an unknown reason and the paint layer was modified to mask the figures belonging to the original composition. The right side of the panel, without preparation is less absorbing to X Rays (Fig. 6a, 6d). The dark paint layer applied on the ground and on the left arm of the lady are different in the IR image (Fig. 6d). Curiously the painting was enlarged on its right side. A varnish was applied on the paint layer after these modifications and a few restorations are visible on top of this varnish in the UV fluorescence image (Fig. 6c). Pentimento on the right hand of the lady are visible on the X ray image (Fig. 6g).

Figure 6: *oil painting on wood*, 1575 – 1599, *Louvre museum (RF 2090). From left to right: X Ray overview, raking light, ultraviolet fluorescence, infra-red, details of X Ray*

The colours of the painting (Fig. 7a) are identified with the GPU colour spectral viewer. By choosing a set of pixels per type of colour in the L*a*b* D65 image, we can calculate clusters based around the mean ΔE value of 4. The location of the clusters in the painting composition validates their pertinence. For example, the black green of the background is made of chromium green (a nineteenth century pigment) to cover the masked figures (Fig. 7e), the restored parts of the background (Fig. 7f), the dress and the hairs (Fig. 7g) and the left arm (Fig. 7h), the lips, the gemstones, the dress and the hairs, the gold, and finally the flesh-tones and the ruff (Fig. 7l). The segmentation reveals also the location of the highlights and of the shadows in the flesh tones (Fig. 7i, j, k). The 10 clusters are visible in the views of the L*a*b* graph (Fig. 7c). 70% of the segmented pixels are black. The highlights of the face are made with a different pigment from that of the rest of the fleshtones within the face. Furthermore, the red of the lips and that of the gemstones are again different pigments from that used in the highlights. In other words, the different red tones are not single pigments mixed with white.

For pigment characterisation, we need to visualize the individual pixels of colour in the L*a*b* graph for each cluster determined (Fig. 7d) to identify saturated pure pigments from mixtures. Then, the spectral reflectance curves of these pigments are compared to references of pure pigments and mixtures in L*a*b* space. Finally, the varnish removal simulation can be calculated taking the highlight of the pearls in the ruff as a white reference (figs 7b).

cluster	%	L*	a*	b*	dΔE
1 (lips)	0.07	37.1693	19.913	21.8421	4
2 (face highlights)	2.4	43.6511	10.6978	28.8944	4
3 (face)	1.8	29.3757	9.18943	11.8552	4
4 (face shadows)	1.3	28.6658	12.4819	7.94663	4
5 (dress and hair)	2	19.9879	8.90325	-1.50598	4
6 (gemstones)	0.08	23.7208	20.5283	0.9951	4
7 (left dark background)	20	17.8084	0.829414	-12.0315	3
8 (general background)	38	16.6693	1.8774	-6.44509	4
9 (sleeve)	10	13.3013	2.29524	-10.52213	3
10 (gold embroidery)	1.4	41.514	3.97144	23.0814	4
sum	77.05				

Table 2: *Segmentation localisation and quantitatve results in L*a*b* colour space*

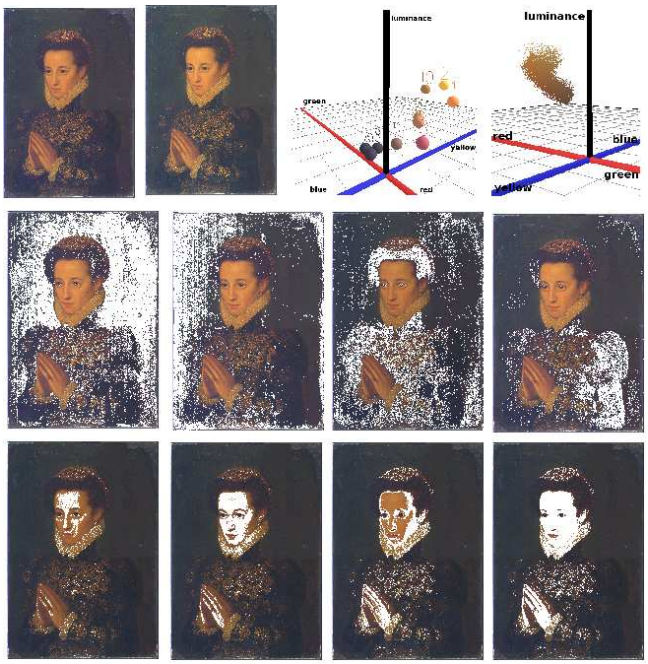

Figure 7: *Segmentation of the "Lady Praying"*

5. CONCLUSION

The main aim of this work has been to propose new tools devoted to the analysis of multispectral images of paintings. Two approaches were used. The first visualized in depth the paint layer with the IIPimage viewer. The second was a colorimetric approach using the GPU spectral viewer for colour clustering and pigment identification by means of the spectral reflectance curves with reference to a bank of pure and pigment mixtures spectral reflectance curves.

Such tools are able a valuable aide in the field of art conservation and restoration and we intend to make use these methods of study more generally with other works of art.

REFERENCES

[1] Martinez, K. (1991) High Resolution Digital Imaging of Paintings: The Vasari Project. Microcomputers for Information Management 8(4) pp. 277-283.

[2] IIPImage, High Resolution Remote Image Viewing, http://iipimage.sf.net

[3] CRISATEL, (programme IST2-1999 n20163), Septembre 2001 - February 2005

[4] R. Yang and G. Welch, "Fast Image Segmentation and Smoothing Using Commodity Graphics Hardware", *Journal of graphics tools*, to appear, special issue on "Hardware-Accelerated Rendering Techniques", 2003.

[5] J. Krüger and R. Westermann, "Linear Algebra Operators for GPU Implementation of Numerical Algorithms", *SIGGRAPH*, 2003.

[6] P. Colantoni, N. Boukala and J. Da Rugna, Fast and Accurate Color Image Processing Using 3D Graphics Cards, 2003, pages 383-390, Vision Modeling anv Visualization, VMV 2003.

[7] *n*Vidia, "C for Graphic", *www.nvidia.com*, 2002.