1992-January 3

# Optimal Spectral Sampling for Color Imaging

Borges, Carlos F.

C.F. Borges, Optimal Spectral Sampling for Color Imaging, Color Hard Copy and Graphic Arts, SPIE Vol. 1670, February 1992, pp. 353-358.
https://hdl.handle.net/10945/38065

# Optimal Spectral Sampling for Color Imaging

Carlos F. Borges

Code Ma/Bc
Naval Postgraduate School
Monterey, CA 93943

January 3, 1992

ABSTRACT

I consider the problem of numerically computing tristimulus values for a given spectral power density. In particular, I examine the use of interpolatory quadrature rules for the solution of this problem. A good deal of effort has gone into creating tables of weights and abcissas for solving this problem [3]. Wallis [2] has proposed a more sophisticated approach using Gauss quadrature rules. I show that the performance of these techniques can be improved in a well-defined sense, and derive a method based on a new class of quadrature rules. These rules give optimal performance in the sense that they maximize the overall degree of precision while simultaneously minimizing the number of function evaluations.

## 1 Introduction

A fundamental problem in color science is the determination of the color of a light with a given spectral power density (SPD). Physically, an SPD is assumed to be a continuous functions over the interval of visible wavelengths (roughly 380 to 780nm). Following the Young-Helmholtz theory of trichromatic vision, the tristimulus coordinates can be found by evaluating the following definite integrals

$$\begin{aligned} X &= \int_{\mathcal{V}} P_\lambda \bar{\mathbf{x}}(\lambda) d\lambda \\ Y &= \int_{\mathcal{V}} P_\lambda \bar{\mathbf{y}}(\lambda) d\lambda \\ Z &= \int_{\mathcal{V}} P_\lambda \bar{\mathbf{z}}(\lambda) d\lambda \end{aligned} \tag{1}$$

where $\mathcal{V}$ is the visible interval and $\bar{\mathbf{x}}(\lambda), \bar{\mathbf{y}}(\lambda)$, and $\bar{\mathbf{z}}(\lambda)$ are the *color matching functions*, a set of empirically determined weights whose existence follows from the linearity of color vision and the invariant color appearance of monochromatic light, see [3].

Historically, these definite integrals have been approximately evaluated using interpolatory quadrature rules. The most common of these methods are called the *weighted ordinate method* and the *selected ordinate method.* Wallis [2] proposed an improved approach based on Gauss quadrature rules and gives weights and abcissas in his paper.

Generally, an approximate quadrature rule is a linear functional $\tilde{\mathbf{Q}} : \mathcal{F} \rightarrow \Re$ such that

$$\tilde{\mathbf{Q}} f \approx \int_a^b f(x)\omega(x)dx.$$

The set of definite integrals in 1 is a collection of simple quadrature problems; there are many numerical methods for evaluating them, and we can construct a set of approximate quadrature rules, $\tilde{\mathbf{Q}}_k$, such that

$$\tilde{\mathbf{Q}}_k f \approx \int_{\mathcal{V}} f(\lambda)\omega_k(\lambda)d\lambda$$

0-8194-0824-7/92/$4.00

for $k = 1, 2, 3$ where $\omega_1(\lambda) = \bar{\mathbf{x}}(\lambda)$, $\omega_2(\lambda) = \bar{\mathbf{y}}(\lambda)$, and $\omega_3(\lambda) = \bar{\mathbf{z}}(\lambda)$.

We shall restrict ourselves to the case where each of the $\tilde{\mathbf{Q}}_k$ is an $n$-point primitive rule, that is, the individual rules have the form

$$\tilde{\mathbf{Q}}_k f = \sum_{i=1}^{n} f(\lambda_{k,i}) w_{k,i}.$$

where the $\{x_{k,i}\}_{i=1}^{n}$ are a fixed set of *abcissas* and the $w_{k,i}$ are fixed weights. The primitive rules are categorized by different methods of choosing the weights and abcissas.

Note that the problem in 1 does not in any way demand uniformity in the rules. If the problem is viewed as a collection of independent, unrelated definite integrals then any haphazard collection of rules is a solution. However, there are obvious advantages to uniformity – it will reduce book-keeping, it will enhance performance, and it will simplify error analysis.

We need only decide on the method of choosing the weights and abcissas to complete the specification of a set of quadrature rules. It seems reasonable to derive each of the $\tilde{\mathbf{Q}}_k$ in the same way. We consider

Method I. *Let $\tilde{\mathbf{Q}}_k$ be the $n$-point Gauss rule*[1] *for the admissible*[2] *weight function $\omega_k(\lambda)$.*

This approach is straightforward. Each Gauss rule has degree of precision $2n - 1$. For our particular problem, such a set of rules will be said to have an *overall* degree of precision $2n - 1$. Since Gauss rules give the maximum achievable degree of precision for $n$-point primitive rules, it follows that no other collection of $n$-point primitive rules has a higher overall degree of precision (i.e. $2n - 1$ is maximal).

Unfortunately, this method uses data poorly. In particular, it requires that $f(\lambda)$ be evaluated more often than necessary to achieve the observed degree of precision. Consider our situation. Each of the three rules has a set of $n$ associated abcissas, $\{\lambda_{k,i}\}_{i=1}^{n}$. There is no reason to suspect redundancies, so we probably need to evaluate the integrand at $3n$ distinct points. Were we to use all of this data in each rule we could easily construct three rules with an overall degree of precision $3n - 1$. Important information is being wasted; this can be seen if we introduce the notion of a performance ratio, defined as

$$R = \frac{\text{Overall degree of precision} + 1}{\text{Number of integrand evaluations}}$$

It is easy to see that for this approach $R = 2/3$. This indicates rather poor performance. Another method for choosing the weights and abcissas, one that accepts a lower degree of precision in return for a more efficient use of information, is

Method II. *Select a set of $n$ distinct abcissas, $\{\lambda_1, \lambda_2, ..., \lambda_n\}$, in the interval $[a, b]$ and let each $\tilde{\mathbf{Q}}_k$ be of the form*

$$\tilde{\mathbf{Q}}_k f = \sum_{i=1}^{n} f(\lambda_i) w_{k,i}.$$

*Choose the weights $w_{k,i}$ so that they satisfy the following Vandermonde system*

$$\begin{bmatrix} 1 & 1 & \cdots & 1 \\ \lambda_1 & \lambda_2 & \cdots & \lambda_n \\ \lambda_1^2 & \lambda_2^2 & \cdots & \lambda_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_1^{n-1} & \lambda_2^{n-1} & \cdots & \lambda_n^{n-1} \end{bmatrix} \begin{bmatrix} w_{k,1} \\ w_{k,2} \\ w_{k,3} \\ \vdots \\ w_{k,n} \end{bmatrix} = \begin{bmatrix} \mu_{k,0} \\ \mu_{k,1} \\ \mu_{k,2} \\ \vdots \\ \mu_{k,n-1} \end{bmatrix}$$

*for $k = 1, 2, 3$. Where*

$$\mu_{k,i} = \int_a^b \lambda^i \omega_k(\lambda) d\lambda$$

This is the *weighted ordinate* method and it binds the rules together with a single set of abcissas (we will call these *shared-abcissa rules*). The integrand must be evaluated $n$ times, and the overall degree of precision is $n - 1$. This method uses information efficiently; each rule uses all of the available data about the integrand.

[1] This is the approach considered by Wallis [2]

[2] For the Gauss rules a weight function is admissible if it is bounded, continuous, and non-negative over the interval of integration.

This method has a performance ratio of $R = 1$. This is better but there is still room for improvement. Because the selection of abcissas is arbitrary, the rule may not be the best possible. A clever method for choosing the shared abcissas could increase the precision of this approach.

Our goal is to find an *optimal* set of shared abcissas by mimicking the development of the Gauss rules. Recall that the derivation of a Gauss rule involves simultaneously finding a set of abcissas and weights that satisfy the following equations

$$\begin{aligned} \sum_{i=1}^{n} w_i &= \int_a^b \omega(\lambda)d\lambda \\ \sum_{i=1}^{n} \lambda_i w_i &= \int_a^b \lambda\omega(\lambda)d\lambda \\ \sum_{i=1}^{n} \lambda_i^2 w_i &= \int_a^b \lambda^2\omega(\lambda)d\lambda \\ &\vdots \\ \sum_{i=1}^{n} \lambda_i^j w_i &= \int_a^b \lambda^j\omega(\lambda)d\lambda \end{aligned}$$

for the largest possible value of $j$.

It is well known that if the weight function satisfies the admissibility conditions then this can always be done for $j = 2n-1$ since there are a total of $2n$ unknowns ($n$ weights and $n$ abcissas). Moreover, it is not generally possible to do it for a larger value of $j$. We propose an analogous approach. Note that for a set of three shared abcissa rules we have a total of $4n$ unknowns ($n$ weights for each of the three weight functions, plus $n$ abcissas) so it seems reasonable to try and satisfy a total of $4n$ equations. To proceed with the derivation let $S_n = \{\nu_1, \nu_2, \nu_3\}$ be an indexed triplet of non-negative integers such that $\nu_1 + \nu_2 + \nu_3 = n$. We shall call $S_n$ a degree sequence. Consider the following method

Method III. *Let each* $\tilde{\mathbf{Q}}_k$ *be of the form*

$$\tilde{\mathbf{Q}}_k f = \sum_{i=1}^{n} f(\lambda_i) w_{k,i}$$

*where the weights,* $w_{k,i}$*, and the abcissas,* $\{\lambda_1, \lambda_2, ..., \lambda_n\}$*, satisfy the following* $4n$ *equations*

$$\begin{aligned} \sum_{i=1}^{n} w_{k,i} &= \int_a^b \omega_k(\lambda)d\lambda \\ \sum_{i=1}^{n} \lambda_i w_{k,i} &= \int_a^b \lambda\omega_k(\lambda)d\lambda \\ \sum_{i=1}^{n} \lambda_i^2 w_{k,i} &= \int_a^b \lambda^2\omega_k(\lambda)d\lambda \\ &\vdots \\ \sum_{i=1}^{n} \lambda_i^{n+\nu_k-1} w_{k,i} &= \int_a^b \lambda^{n+\nu_k-1}\omega_k(\lambda)d\lambda \end{aligned} \tag{2}$$

*for* $k = 1, 2, 3$.

This method yields a total of $4n$ equations in the unknown weights and abcissas. These equations have been carefully divided among the three weight functions. Note that, whenever the system has a solution, the quadrature rule $\tilde{\mathbf{Q}}_k$ will have a degree of precision of $n + \nu_k - 1$. Moreover, if $n = 3l$ and we choose $S_n = \{l, l, l\}$ then the *overall* degree of precision will be $n + l - 1$ which is, in general, the maximum achievable. Such a set of rules will be called *uniform* and the performance ratio is $R = 4/3$.

## 2 Finding the Weights and Abcissas

If the abcissas, $\lambda_i$, are known then solving for the weights is straightforward and can be accomplished by solving a Vandermonde system. In particular, the weights for the $k$'th weight function are given by

$$\begin{bmatrix} 1 & 1 & \cdots & 1 \\ \lambda_1 & \lambda_2 & \cdots & \lambda_n \\ \lambda_1^2 & \lambda_2^2 & \cdots & \lambda_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_1^{n-1} & \lambda_2^{n-1} & \cdots & \lambda_n^{n-1} \end{bmatrix} \begin{bmatrix} w_{k,1} \\ w_{k,2} \\ w_{k,3} \\ \vdots \\ w_{k,n} \end{bmatrix} = \begin{bmatrix} \mu_{k,0} \\ \mu_{k,1} \\ \mu_{k,2} \\ \vdots \\ \mu_{k,n-1} \end{bmatrix}$$

There are well-known methods for solving Vandermonde systems, and we will not discuss them here.

However, solving the equations in 2 is more difficult since they are not linear in the unknowns. However, they can be transformed into a form which can be solved more easily by applying Prony's algorithm.

Consider the monic polynomial $q(\lambda) = (\lambda - \lambda_1)(\lambda - \lambda_2)\cdots(\lambda - \lambda_n)$ whose roots are the abcissas $\lambda_i$. There certainly exist coefficients $c_i$ for $i = 0, 1, ..., n$ such that $q(\lambda) = c_0 + c_1\lambda + \cdots + c_n\lambda^n$. Fixing $k$ and using these coefficients to weight the first $n+1$ equations from 2 yields

$$\begin{aligned} w_{k,1}c_0 + w_{k,2}c_0 + \cdots + w_{k,n}c_0 &= \mu_{k,0}c_0 \\ w_{k,1}c_1\lambda_1 + w_{k,2}c_1\lambda_2 + \cdots + w_{k,n}c_1\lambda_n &= \mu_{k,1}c_1 \\ &\vdots \\ w_{k,1}c_n\lambda_1^n + w_{k,2}c_n\lambda_2^n + \cdots + w_{k,n}c_n\lambda_n^n &= \mu_{k,n}c_n \end{aligned} \tag{3}$$

summing these equations and rearranging terms yields

$$\begin{aligned} \mu_{k,0}c_0 + \mu_{k,1}c_1 + \cdots + \mu_{k,n}c_n = \quad & w_{k,1}(c_0 + c_1\lambda_1 + \cdots + c_n\lambda_1^n) \\ + \quad & w_{k,2}(c_0 + c_1\lambda_2 + \cdots + c_n\lambda_2^n) \\ & \vdots \\ + \quad & w_{k,n}(c_0 + c_1\lambda_n + \cdots + c_n\lambda_n^n) \end{aligned} \tag{4}$$

Notice that the typical term in parentheses is $\sum_{i=0}^{n} c_i\lambda_k^i = q(\lambda_k) = 0$. Hence, all of the terms on the right vanish leaving

$$\mu_{k,0}c_0 + \mu_{k,1}c_1 + \cdots + \mu_{k,n}c_n = 0 \tag{5}$$

and since $q(\lambda)$ is monic this becomes

$$\mu_{k,0}c_0 + \mu_{k,1}c_1 + \cdots + \mu_{k,n-1}c_{n-1} = -\mu_{k,n} \tag{6}$$

This procedure can be repeated by keeping $k$ fixed and picking out the next block of $n+1$ equations from 2. This yields

$$\begin{aligned} w_{k,1}c_0\lambda_1 + w_{k,2}c_0\lambda_2 + \cdots + w_{k,n}c_0\lambda_n &= \mu_{k,1}c_0 \\ w_{k,1}c_1\lambda_1^2 + w_{k,2}c_1\lambda_2^2 + \cdots + w_{k,n}c_1\lambda_n^2 &= \mu_{k,2}c_1 \\ &\vdots \\ w_{k,1}c_n\lambda_1^{n+1} + w_{k,2}c_n\lambda_2^{n+1} + \cdots + w_{k,n}c_n\lambda_n^{n+1} &= \mu_{k,n+1}c_n \end{aligned} \tag{7}$$

summing and rearranging yields

$$\begin{aligned} \mu_{k,1}c_0 + \mu_{k,2}c_1 + \cdots + \mu_{k,n+1}c_n = \quad & w_{k,1}\lambda_1(c_0 + c_1\lambda_1 + \cdots + c_n\lambda_1^n) \\ + \quad & w_{k,2}\lambda_2(c_0 + c_1\lambda_2 + \cdots + c_n\lambda_2^n) \\ & \vdots \\ + \quad & w_{k,n}\lambda_n(c_0 + c_1\lambda_n + \cdots + c_n\lambda_n^n) \end{aligned} \tag{8}$$

and the terms on the right vanish as before leaving

$$\mu_{k,1}c_0 + \mu_{k,2}c_1 + \cdots + \mu_{k,n}c_{n-1} = -\mu_{k,n+1} \tag{9}$$

This procedure is repeated until one obtains the following set of equations

$$\begin{aligned} \mu_{k,0}c_0 + \mu_{k,1}c_1 + \cdots + \mu_{k,n-1}c_{n-1} &= -\mu_{k,n} \\ \mu_{k,1}c_0 + \mu_{k,2}c_1 + \cdots + \mu_{k,n}c_{n-1} &= -\mu_{k,n+1} \\ &\vdots \\ \mu_{k,\nu_k-1}c_0 + \mu_{k,\nu_k}c_1 + \cdots + \mu_{k,n+\nu_k-2}c_{n-1} &= -\mu_{k,n+\nu_k-1} \end{aligned} \tag{10}$$

Similar sets of equations can be obtained for each value of $k = 1, 2, 3$, producing a set of $n$ linear equations in the unknowns $c_i$. Gathering these together and putting them in matrix form yields

$$\begin{bmatrix} \mu_0^{(1)} & \mu_1^{(1)} & \cdots & \mu_{n-1}^{(1)} \\ \mu_1^{(1)} & \mu_2^{(1)} & \cdots & \mu_n^{(1)} \\ \vdots & \vdots & \ddots & \vdots \\ \mu_{\nu_1-1}^{(1)} & \mu_{\nu_1}^{(1)} & \cdots & \mu_{n+\nu_1-2}^{(1)} \\ \mu_0^{(2)} & \mu_1^{(2)} & \cdots & \mu_{n-1}^{(2)} \\ \mu_1^{(2)} & \mu_2^{(2)} & \cdots & \mu_n^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ \mu_{\nu_2-1}^{(2)} & \mu_{\nu_2}^{(2)} & \cdots & \mu_{n+\nu_2-2}^{(2)} \\ \mu_0^{(3)} & \mu_1^{(3)} & \cdots & \mu_{n-1}^{(3)} \\ \mu_1^{(3)} & \mu_2^{(3)} & \cdots & \mu_n^{(3)} \\ \vdots & \vdots & \ddots & \vdots \\ \mu_{\nu_3-1}^{(3)} & \mu_{\nu_3}^{(3)} & \cdots & \mu_{n+\nu_3-2}^{(3)} \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_{n-1} \end{bmatrix} = - \begin{bmatrix} \mu_n^{(1)} \\ \mu_{n+1}^{(1)} \\ \vdots \\ \mu_{n+\nu_1-1}^{(1)} \\ \mu_n^{(2)} \\ \mu_{n+1}^{(2)} \\ \vdots \\ \mu_{n+\nu_2-1}^{(2)} \\ \mu_n^{(3)} \\ \mu_{n+1}^{(3)} \\ \vdots \\ \mu_{n+\nu_3-1}^{(3)} \end{bmatrix} \tag{11}$$

This system has a unique solution whenever the matrix is non-singular (note that the matrix is square, despite its appearance) and this solution can be used to obtain the coefficients of the polynomial $q(\lambda)$ from which the abcissas can be computed.

In the classical one-weight case the matrix of interest is an $n \times n$ Hankel matrix constructed from the moment sequence associated with the weight function. When applied to the problem now under consideration the structure of the matrix is somewhat different. In particular, we get a matrix where the first $\nu_1 \times n$ block is Hankel in the moments of the first weight function, the next $\nu_2 \times n$ block is Hankel in the moments of the second weight function, and the last $\nu_3 \times n$ block is Hankel in the moments of the third weight function.

In order to construct these rules for the 1931 CIEXYZ matching functions we need to first generate the moments. This was accomplished by taking analytic approximations to the matching functions based on those proposed by Moon and Spencer [1] but with some slight modifications. The particular approximations used were

$$\begin{aligned} \bar{x}(\lambda) &= \frac{10^{32.0927}}{\lambda^{365.3388}} e^{\frac{-164.9506}{\lambda}} - \frac{10^{54.5115}}{\lambda^{500.0}} e^{\frac{-237.5}{\lambda}} + \frac{10^{69.5948}}{\lambda^{336.0385}} e^{\frac{-199.1054}{\lambda}} \\ \bar{y}(\lambda) &= \frac{10^{32.4138}}{\lambda^{182.1905}} e^{\frac{-100.937}{\lambda}} \\ \bar{z}(\lambda) &= \frac{10^{32.7526}}{\lambda^{365.3388}} e^{\frac{-164.9506}{\lambda}} \end{aligned}$$

Where the wavelength is assumed to be measured in microns, in effect, the visible spectrum is $0.360 \leq \lambda \leq 0.830$.

Evaluating the moments of these *analytic matching functions* is straightforward. Consider the $m$'th moment of any function of the form

$$\int_0^\infty \frac{A}{\lambda^p} e^{\frac{-q}{\lambda}} d\lambda \tag{12}$$

By definition of the Gamma function, the $m$'th moment is given by

$$\int_0^\infty \lambda^m \frac{A}{\lambda^p} e^{\frac{-q}{\lambda}} d\lambda = A \frac{\Gamma(p-m-1)}{q^{p-m-1}} \tag{13}$$

In the case under consideration $A, \Gamma(p-m-1)$, and $q^{p-m-1}$ are all quite large, even though the moments are small. This calls for the numerically unstable task of evaluating a very small number with very large ones, which can lead to catastrophic cancellations. It is better to use the natural logarithm and consider

$$e^{\ln A + \ln \Gamma(p-m-1) - (p-m-1)\ln q} \tag{14}$$

As there are stable numerical methods for evaluating the natural logarithm of the Gamma function, the moments can be evaluated far more precisely in this way.

Software was developed that uses these moments and equation 11 to find a polynomial, $q(\lambda)$, that is quasi-orthogonal with respect to the set of weights $\mathcal{W} = \{\bar{x}(\lambda), \bar{y}(\lambda), \bar{z}(\lambda)\}$ and a given degree sequence. This software then finds and reports the roots of $q(\lambda)$ that lie in the visible interval. The abcissas for a number of quasi-Gaussian rules were determined using this software and are listed in table 1. Note that the degree sequences in this table are ordered for $\bar{x}(\lambda), \bar{y}(\lambda)$, and $\bar{z}(\lambda)$ respectively.

| Degree | Abcissas (by wavelength in nm) |
|---|---|
| 1,1,1 | 445.3, 535.8, 619.4 |
| 2,1,1 | 441.0, 523.4, 574.3, 643.1 |
| 1,2,1 | 443.1, 531.8, 607.6, 720.0 |
| 1,1,2 | 425.3, 469.3, 543.0, 627.6 |
| 2,2,2 | 419.8, 461.7, 526.6, 582.7, 638.6, 707.7 |
| 3,3,3 | 391.4, 422.6, 453.8, 488.8, 539.1, 587.7, 635.2, 689.4, 767.2 |

Table 1: Some Quasi-Gaussian Abcissas for the CIEXYZ matching functions.

## 3 Acknowledgements

I would like to acknowledge Ken Joy and the other members of the Computer Graphics Research Laboratory at UC Davis for their help in conducting this research.

## References

[1] P. Moon and D. Spencer, *Analytical representation of trichromatic data*, J. Optical Soc. Am., (1945), pp. 399–427.

[2] R. Wallis, *Fast computation of tristimulus values by use of Gaussian quadrature*, J. Optical Soc. Am., (1975), pp. 91–94.

[3] G. Wyszecki and W. Stiles, *Color Science: Concepts and Methods, Quantitative Data and Formulae*, John Wiley and Sons, 1982.