Making the calculation of Logvinenko's coordinates easy.

Graham D. Finlayson*[1]*, Michal Mackiewicz*[1]*, Anya Hurlbert*[2]*;* [1] *School of Computing Sciences, University of East Anglia, Norwich, UK;* [2] *Newcastle University, Newcastle, UK

Abstract

In 2009 Logvinenko introduced a new object colour space that utilises representation of non-luminous surfaces as reflectance spectra of a rectangular shape i.e. a reflectance spectrum takes two values $\frac{1+\alpha}{2}$ *or* $\frac{1-\alpha}{2}$ *for* $0 \leq \alpha \leq 1$ *with two transitions at* λ_1 *and* λ_2*. The calculation of Logvinenko colour coordinates for a large set of reflectance spectra (or tristimulous values) is time consuming, even despite a more recent more efficient algorithm of Godau and Funt. In this paper, we propose two approximate, but fast solutions to finding the Logvinenko coordinates (ADL) that exploit the combinatorial properties of rectangular spectra. The proposed algorithm takes around 0.02s to calculate the ADL coordinates for 1600 surface database as opposed to earlier implementation that reported 90s. Both algorithms are approximate, but the precision should be acceptable for most of the applications as the calculated mean, median, 95 percentile and max* ΔE *Lab errors were respectively 0.3, 0.2, 0.7 and 1.6.*

Introduction

Colours of all reflecting objects under certain fixed illumination are contained within a closed convex volume called *object-colour solid* [11]. In Figure 1, we plot the volume of all reflecting objects under illuminant D65 in the CIE 1931 colour space (XYZ) [11]. Unlike the points in the interior of the object-colour solid that represent different *metameric classes* where each class corresponds to infinitely many reflectance spectra, each point on the object-colour solid surface has only one corresponding reflectance spectrum. These 'surface' spectra are called *optimal* and have the property of being functions of zeros and ones with two transitions only [10, 11]. They are also the spectra that for a given chromaticity have the greatest reflectivity.

The uniqueness property of the object colour surface (sensor responses mapping to unique) spectra is useful as it teaches that a trichromatic response can be mapped uniquely to a surface reflectance. Moreover, this reflectance has the simple 'two transition' form. This uniqueness property allows one to map an XYZ measured under one illuminant exactly to the corresponding XYZ under a second light.

For interior points of the object colour solid there is a set (a metamer set [4, 11]) of different reflectances that integrate to form a single tristimuli. Yet we could still map responses across lights if we could systematically recover plausible reflectance spectra from tristimulus measurements. We would simply compute the desired reflectance (for a tristimulus under the first light) and then integrate this reflectance with the second target light. Finlayson and Morovic followed this approach in what they called "Metamer Constrained Colour Correction" [3]. Of course, the interesting question that arises is "which reflectance to choose from the metamer set?".

Logvinenko defines a *colour atlas* to be a set of reflectance spectra which, for any illuminant (with power through the visible spectrum), map bijectively to the corresponding tristmulus values [6]. Each tristimulus in the object colour solid corresponds to a single atlas reflectance and vice versa.

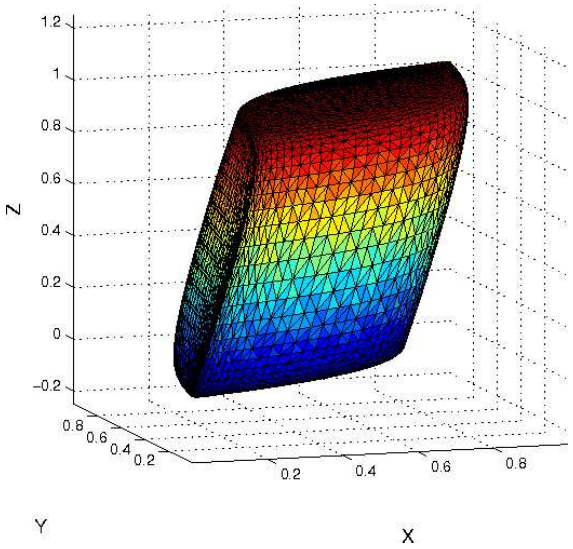

Figure 1. *Object colour solid as evaluated for the CIE 1931 CMFs and the illuminant D65. Every 4nm sampling.*

Logvinenko colour atlas

The Logvinenko colour atlas comprises spectra fulfilling the following equation:

$$x(\lambda) = (1-\alpha)x_{0.5}(\lambda) + \alpha x_{opt}(\lambda) \tag{1}$$

where $x_{opt}(\lambda)$ is the optimal spectrum and $x_{0.5}(\lambda) = 0.5$. The variable α denotes the *chromatic amplitude* of the optimal metamer and is beween 0 and 1. As can be seen in Figure 2, α signifies the degree to which the optimal spectrum x_{opt} is mixed with the grey spectrum $x_{0.5}$.

The Logvinenko colour coordinates (ADL) are three parameters, one having been defined as α and the other two describing the optimal spectrum x_{opt}. These two are *spectral width* and *central wavelength* denoted as:

$$\delta = (\lambda_{max} - \lambda_{min}) - |\lambda_1 - \lambda_2| \tag{2}$$

and

$$\overline{\lambda} = \begin{cases} \lambda_1 + \frac{\delta}{2} & \text{if } \lambda_1 + \frac{\delta}{2} < \lambda_{max} \\ \lambda_2 - \frac{\delta}{2} & \text{else} \end{cases} \tag{3}$$

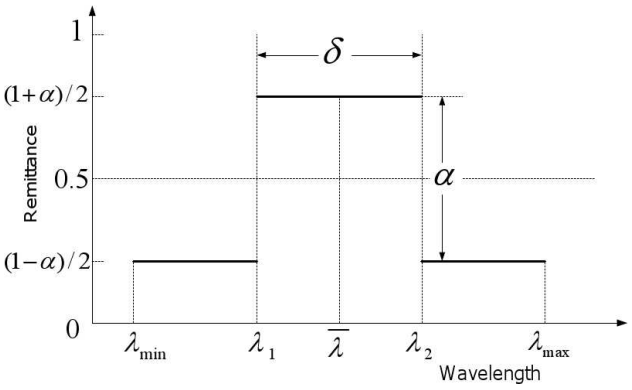

Figure 2. *Logvinenko Type I rectangular spectrum (non-optimal as $\alpha \neq 1$).*

where λ_{min} and λ_{max} denote the ends of the visible spectrum. Thus, there are two alternative parametrisation of ADL coordinates: the first comprises spectral width and central wavelength; and the second transition wavelengths λ_1 and λ_2.

There are two types of optimal spectra $x_{opt}(\lambda;\lambda_1,\lambda_2)$: Type I are those that are 1 for $\lambda_1 < \lambda < \lambda_2$ and zero everywhere else (see Figure 2) and Type II are those that are 0 for $\lambda_2 < \lambda < \lambda_1$ and one everywhere else. This notation follows Logvinenko's where $\lambda_1 < \lambda_2$ for Type I and $\lambda_2 < \lambda_1$ for Type II spectra.

The sensor responses to the Logvinenko metamer under illuminant $I(\lambda)$ can be written as

$$\varphi_i(\alpha,\lambda_1,\lambda_2) = \int_{\lambda_{min}}^{\lambda_{max}} \left((1-\alpha)x_{0.5}(\lambda) + \alpha x_{opt}(\lambda;\lambda_1,\lambda_2)\right) s_i(\lambda)I(\lambda)d\lambda \quad (4)$$

where $s_i(\lambda)$ are the three sensor sensitivities.

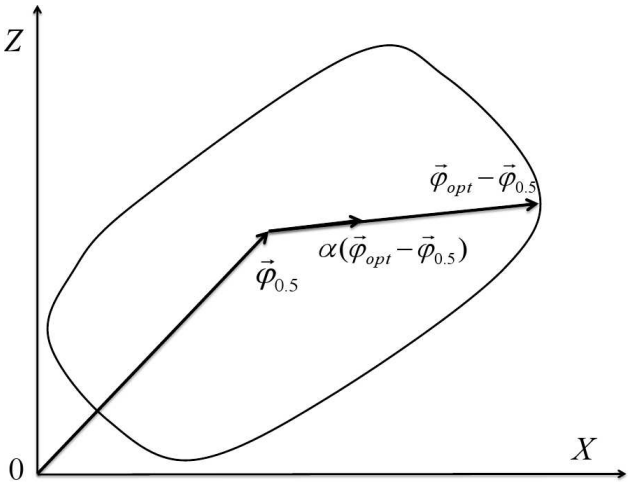

Figure 3. *A cross-section of the object-colour solid in the Y plane. $\vec{\varphi}_{0.5}$ denotes a vector of tristimulus values calculated from $x_{0.5}$ spectrum and $\vec{\varphi}_{opt}$ from x_{opt}*

This paper deals with the inverse calculation that is given sensor responses ψ_i we want to find the triplet $(\alpha,\lambda_1,\lambda_2)$ such that $\varphi_i(\alpha,\lambda_1,\lambda_2) = \psi_i$

Godau and Funt's approach to Metamer calculation

Godau and Funt approached the calculation of ADL coordinates as an optimisation problem [5]. The intuition of their approach is as follows (also see Figure 3). Suppose $\vec{\varphi}$ is the target tristimuli. We of course know the response to 50% grey (the centre of the object-colour solid) and denote it $\vec{\varphi}_{0.5}$. We can calculate the vector joining these tristimuli $\vec{\varphi}-\vec{\varphi}_{0.5}$. Clearly, if we could find the correct optimal colour on the object colour boundary, denoted $\vec{\varphi}_{1,\lambda_1,\lambda_2}$ then the vector joining this to the object-colour solid centre is in the same direction. We write this vector as $\vec{\varphi}_{1,\lambda_1,\lambda_2}-\vec{\varphi}_{0.5}$. Remembering that the cosine of the angle between two vectors can be calculated as $(\vec{a}.\vec{b})/(|\vec{a}||\vec{b}|)$ we seek λ_1 and λ_2 such that we minimise:

$$E(\alpha,\lambda_1,\lambda_2) = \arccos \frac{(\vec{\varphi}-\vec{\varphi}_{0.5}).(\vec{\varphi}_{1,\lambda_1,\lambda_2}-\vec{\varphi}_{0.5})}{|\vec{\varphi}-\vec{\varphi}_{0.5}||\vec{\varphi}_{1,\lambda_1,\lambda_2}-\vec{\varphi}_{0.5}|} \quad (5)$$

Once the optimisation is finished and hence $\vec{\varphi}_{1,\lambda_1,\lambda_2}$ is known, the calculation of the remaining parameter α is straightforward.

$$\alpha = \frac{|\vec{\varphi}-\vec{\varphi}_{0.5}|}{|\vec{\varphi}_{1,\lambda_1,\lambda_2}-\vec{\varphi}_{0.5}|} \quad (6)$$

Below we explain their method in more detail. First, (λ_1,λ_2) - the starting points for optimisation are found using the following interpolation procedure. The grey-centred sensor responses $(\vec{\varphi}-\vec{\varphi}_{0.5})$ are converted to spherical coordinates (Θ,ϕ) i.e. vector direction is encoded by the elevation and azimuth angles. Then, the authors look for a function $F:(\Theta,\phi)\rightarrow(\lambda_1,\lambda_2)$. They create only one interpolation function for Type I spectra.

Once the interpolation function is in place, we can start searching for ADL coordinates of the input sensor responses $\vec{\varphi}$. Since Type I and Type II reflectances are symmetric with respect to grey then a complementary (Type II) sensor responses can be written as

$$\vec{\varphi}^{\mathrm{II}} = \vec{\varphi}_{0.5} - (\vec{\varphi}-\vec{\varphi}_{0.5}) \quad (7)$$

Both $\vec{\varphi}^{\mathrm{II}}$ and $\vec{\varphi}$ are then converted to spherical coordinates (Θ,ϕ). Then, the aforementioned interpolation function (trained for the Type I reflectance spectra) is used for both $\vec{\varphi}^{\mathrm{II}}$ and $\vec{\varphi}$. Finally (λ_1,λ_2) are found according to the following equation:

$$(\lambda_1,\lambda_2) = \begin{cases} (\lambda_1^{\mathrm{I}},\lambda_2^{\mathrm{I}}) & \text{if } E(1,\lambda_1^{\mathrm{I}},\lambda_2^{\mathrm{I}}) \leq E(1,\lambda_1^{\mathrm{II}},\lambda_2^{\mathrm{II}}) \\ (\lambda_1^{\mathrm{II}},\lambda_2^{\mathrm{II}}) & \text{else} \end{cases} \quad (8)$$

That is if the angular error E is less or equal for $\vec{\varphi}$ than for $\vec{\varphi}^{\mathrm{II}}$, we consider the result of the interpolation a Type I spectrum and if it is greater a Type II spectrum.

The authors report that their interpolation yields very good results which often do not require further refinement using optimisation. For interpolation, the authors used the Matlab TriScatterInt and for optimisation another third party toolbox [9].

With regard to optimisation, the repeated calculations of sensor responses $\vec{\varphi}(1,\lambda_1,\lambda_2)$ are necessary. The authors note that these can be calculated efficiently as

$$\vec{\varphi}_{1,\lambda_1,\lambda_2} = \begin{cases} \vec{\varphi}(\lambda_2) - \vec{\varphi}(\lambda_1) & \text{if } \lambda_1 \leq \lambda_2 \\ \vec{\varphi}(\lambda_{max}) - (\vec{\varphi}(\lambda_1) - \vec{\varphi}(\lambda_2)) & \text{if } \lambda_1 > \lambda_2 \end{cases} \quad (9)$$

where $\vec{\varphi}(\lambda)$ denotes the antiderivative of the integrand from Equation 4 (for $\alpha = 1$). This antiderivative can be pre-calculated with a sufficiently small step, for example 0.1nm.

Godau and Funt's algorithm calculates a very precise estimate of the optimal metamer. The authors reported the maximum angular error 0.001°. The execution time was reported to be 90 seconds for the dataset of 1600 spectra [8]. This is still a significant cost (a 1 Mega pixel image would take over 15 hours to compute the Logvinenko Metamers).

Fast Combinatorial Metamer Calculation

We note that Equation 4 can be rewritten as

$$\vec{\varphi}(\alpha,\lambda_1,\lambda_2) = \vec{\varphi}_{0.5} - \alpha\vec{\varphi}_{0.5} + \alpha\vec{\varphi}_{opt} \quad (10)$$

and equate it to the given sensor responses $\vec{\psi}$

$$\vec{\varphi}_{0.5} + \alpha\left(\vec{\varphi}_{opt} - \vec{\varphi}_{0.5}\right) = \vec{\psi} \quad (11)$$

We denote the grey-centred responses $\vec{\varphi}_{opt} - \vec{\varphi}_{0.5}$ and $\vec{\psi} - \vec{\varphi}_{0.5}$ as $\vec{\xi}_{opt}$ and $\vec{\xi}$ respectively and rewrite Equation 11.

$$\alpha\vec{\xi}_{opt} = \vec{\xi} \quad (12)$$

Our method avoids expensive interpolation and optimisation and exploits the fact that one can generate the set of say k Logvinenko optimal metamers x^k_{opt} and pre-calculate their grey-centred sensor responses $\vec{\xi}^k_{opt}$.

Next, we evaluate α k times for all the candidate optimal metamers minimising the distance between $\vec{\xi}$ and $\alpha_k\vec{\xi}^k_{opt}$ that is:

$$\underset{\alpha_k}{\text{minimise}} \left|\vec{\xi} - \alpha_k\vec{\xi}^k_{opt}\right| \quad (13)$$

The solution to this minimisation is trivial and can be found as follows. We know that the orthogonal projection of $\vec{\xi}$ onto the line in the direction of $\vec{\xi}^k_{opt}$ minimises objective in Equation 13. Let θ be the angle between $\vec{\xi}$ and $\vec{\xi}^k_{opt}$. Then, the length of the projection of vector $\vec{\xi}$ onto the line in the direction of $\vec{\xi}^k_{opt}$ equals $|\vec{\xi}|\cos(\theta)$. On the other hand the length of this projection is $\alpha_k|\vec{\xi}^k_{opt}|$. Thus,

$$\alpha_k|\vec{\xi}^k_{opt}| = |\vec{\xi}|\cos(\theta) \quad (14)$$

after expanding $\cos(\theta)$ and rearranging we obtain the solution:

$$\alpha_k = \frac{\vec{\xi}^k_{opt}\cdot\vec{\xi}}{|\vec{\xi}^k_{opt}|^2} \quad (15)$$

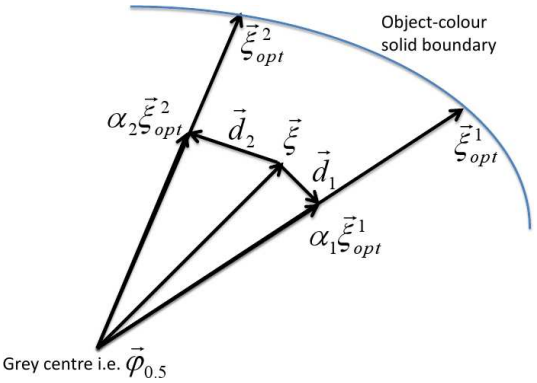

Figure 4. *Calculation of the approximate Logvinenko coordinates.*

The optimal metamer $\varphi(1,\lambda_1,\lambda_2)$ which has the smallest distance $\left|\vec{\xi} - \alpha\vec{\xi}_{opt}\right|$ among all k optimal metamers together with its optimal α give us the approximate solution to our problem - the triplet $(\alpha,\lambda_1,\lambda_2)$. See Figure 4 for geometrical intuition.

In order to perform the above calculations one has to sample the surface of the object-colour solid. This can be done by generating the sequences of zeros and ones corresponding to the optimal reflectance spectra. A reasonable starting point could be 1nm sampling and the visible spectrum range of 380-780nm. In this case, we have 401 sampling points in the visible spectrum and consequently $401\times402/2-1 = 80,600$ Type I optimal spectra and the same number of Type II optimal spectra giving us a total of $161,200$ sampling points on the surface of the object-colour solid. In general if p is the number of sampling points in the visible spectrum, there will be $(p\times(p+1)/2-1))\times2$ sampling points on the object-colour solid. In Figure 1, we can see the object colour solid sampled every 4nm from 405-650nm at 3904 points.

Unlike previous methods, our algorithm is simple and can be implemented in a few lines of Matlab. Moreover, the code can be easily vectorised removing the loop over all candidate optimal metamers and given the memory constraints also the loop over all sensor responses for which we are calculating Logvinenko coordinates.

As we shall see in the next section our method delivers good results. Yet, the reader might wonder whether we are guilty of legerdemain. We solve for α in Equation 15 yet, there is no constraint that α is between 0 and 1. Yet, to recover a Logvinenko Metamer this constraint must be enforced. Conceptually, we could enforce this interval requirement simply using the tool of quadratic programming. However, this optimisation is much more onerously expensive than the method we present. It is always much easier to carry out an unconstrained optimisation than a constrained one.

In fact (with one small caveat) it is straightforward to prove that the least-squares solution we seek will never have α outside the desired range. To see this suppose that $\vec{v}$ and $\vec{w}$ respectively denote the true RGB for which we seek a Logvinenko metamer and the RGB that the metamer our method returns actually generates. Let us now run a second optimisation where now we seek the best metamer for $\vec{w}$. We know in this case we must get the result we just computed. And, further this metamer integrates to $\vec{w}$ without error.

sampling	ΔE				$\Delta\theta$		timings [s]
	mean	med	95 pt.	max	mean	max	
10nm, 380-780	5.3	3.6	15	42	1.8	14	0.1
5nm, 380-780	2.6	1.8	7.3	24	0.9	6.5	0.4
4nm, 380-780	2.1	1.5	5.7	24	0.8	8.0	0.6
3nm, 380-780	1.5	1.0	4.4	15	0.6	5.1	1.2
2nm, 380-780	1.0	0.8	2.8	12	0.4	3.9	2.5
1nm, 380-780	0.5	0.4	1.4	5.3	0.2	1.9	10.5
1nm, 405-650	0.5	0.4	1.5	4.8	0.2	1.8	3.8

Table 1. ΔE **and angular errors for the recovered Logvinenko metamers under D65 illuminant.**

Suppose that the metamer we found has α outside the range $[0,1]$. That is, that our method returns a spectrum that is larger than 1 or has negative values. Now let us 'mix' our RGB with 50% grey and carry out the same mixing with the grey spectrum and our 'out of interval' metamer. At some point as we mix more grey into our metamer the reflectance spectrum lies on the [0,1] interval. We have just created an 'optimal colour' for an interior point of the object colour solid. Yet, we know that all optimal colours map to points on the surface of the colour solid. We have arrived at a circumstance that cannot occur. That is the original metamer we calculated must have alpha in the [0,1] range. We do not need to constrain α. The nature of the formalism of Logvinenko metamers, almost always, allows the simple least squares solution of Equation 15 to be used directly.

The above reasoning accounts for all typical Logvinenko metameric calculations. However, if our RGB is very near the surface of the object colour solid then we could have α out of range. In all our experiments with thousands of surfaces and hundreds of lights this never happened. Real reflectances are far from the shape described by the optimal colour formula.

Results

Following [6] and [5], we used the University of Eastern Finland spectral database of 1600 glossy Munsell samples [8]. According to Godau and Funt, their algorithm takes about 90 seconds to execute to recover Logvinenko Metamers for the 1600 RGBs with a maximum angular error of 0.001°. They also reported that the earlier original algorithm of Logvinenko took more than a day to calculate the ADL coordinates for the same dataset.

We tested our algorithm using different wavelength sampling resolutions, which have the obvious impact on the results (see Table 1). We also varied the limits of the integration - λ_{min} and λ_{max}. Following Logvinenko's advice [6] we shortened the spectrum range from 380-780nm to 405-650nm, which did not have the detrimental effect on the results and reduced significantly the number of optimal spectra (from 161,200 to 60,760). All the results reported were obtained for spectra under D65. We also recorded errors for illuminant A, but there was not any significant difference.

As expected, the results in Table 1 show that the precision of our algorithm cannot match the earlier described optimisation method. In the following section, we propose a further improvement that drastically decreases the execution time and also allows for a further increase in accuracy.

Fast Metamer Calculation Using K-d Tree

Another possibility for improving the algorithm execution time and/or precision would be a look-up table that would link the chromaticity coordinate of the input colour $\vec{\psi}$ to the index of optimal spectra in the 'vicinity' of that colour, whereby significantly reducing the number of candidate optimal spectra needed to be considered for each input. The implementation of this approach could be based on another good idea in the Godau and Funt's paper, which is characterising optimal spectra by the azimuth and elevation angles of their corresponding tristimulus values. Figure 5 shows the distribution of Type I and Type II spectra in these spherical coordinates. The figure was created using 5nm sampling from 405-650nm. We used illuminant D65 and the CIE 1931 CMFs.

We pursued this idea and performed the following experiment using the k-d tree structures [1]. For a fixed illumination we calculated azimuth and elevation for each of our large set of optimal colour XYZs (in the same way as Godau and Funt). The XYZs themselves are calculated by integrating our optimal spectra with the illumination of interest (D65) and the colour matching curves (CIE 1931). Thus, we can visualise all our optimal colours as points on the cartesian grid (as in Fig. 5). To find the Logvinenko metamer for a given query XYZ, it suffices to calculate its azimuth and elevation coordinates and then search, using an efficient data structure like a k-d tree, for local neighbours in our optimal azimuth and elevation point set. Intuitively a small number of neighbours will suffice to find the correct optimal spectrum with respect to which we can find the Logvinenko metamer.

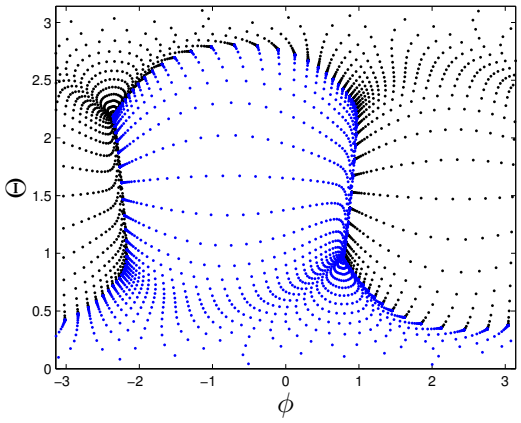

Figure 5. *Distribution of Type I (black) and Type II (blue) spectra in spherical coordinates.*

Results

In Table 2 we show the errors and timings corresponding to the identical experiments carried out in the earlier section, with

the difference being that this time we employed a k-d tree. The implementation of the kd-tree came from the Matlab Statistical Toolbox. For each of the 1600 XYZ queries, we obtained a single nearest neighbour and calculated the errors analogically as in the earlier section. Note that we give here the timings for both building and searching the k-d tree. The relevant figure for obvious reasons is the latter. As expected, the query timings are a small fraction of earlier figures from Table 1 and for the first time will allow us to calculate the Logvinenko metamers for large images in a matter of seconds. The errors for the finest resolution are relatively small and should not have much effect in most of the applications. Arguably, the precision we achieve is likely all that we can reasonably hope for. All real camera systems record RGBs that are corrupted by noise. It is unlikely that the RGB record will be known to a grater accuracy than 1 Delta E (the precision to which we calculate a vast majority of Logvinenko metamers).

Moreover, we performed a further simulation involving a hyperspectral image captured in our lab. The image (see Figure 6) was captured using a Specim VNIR hyperspectral camera (400nm - 1000nm with a 1nm resolution). The illuminant was the D65 metamer produced by the Gamma Scientific RS-5B LED illuminator [7]. The image was clipped to 490 by 350 pixels and 405-650nm range. Next, we calculated the Logvinenko metamers for all the pixels in the image using the kd-tree method. For this image size (171,500 pixels), the execution time was approximately 2s. Next, we rendered both the original hyperspectral image and the Logvinenko metamer image under two illuminants: A and F12 (fluorescent). The resulting images can be seen in Figures 7-10. It can be clearly seen that the Logvinenko metamer image rendered under illuminant A is very similar to the original (Figures 7-8), whereas this is not the case with the other pair. As to F12 illuminant, we can see that many patches were rendered with a large error and in certain areas of the image a high level of noise was introduced (see the red pepper and the adjacent red patch). The intuition behind the latter effect can be gained from Fig. 9. The figure contains the plots of two Logvinenko metamers from two adjacent pixels in the 'noisy' red patch. As one can expect, both metamers are the most reflective in the long end of the spectrum. However, we can also see that the F12 peak at 610nm coincides with high reflectance only in one of them, resulting in the two very different pixel values.

Figure 6. *Original hyperspectral image captured under D65 illuminant.*

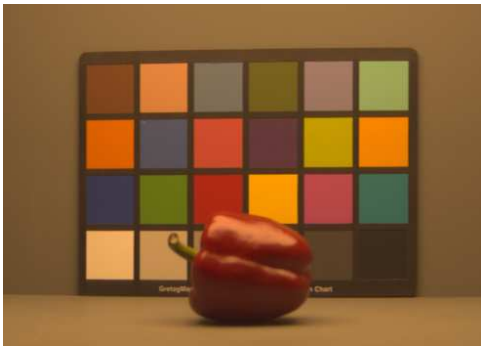

Figure 7. *Original image rendered under illuminant A.*

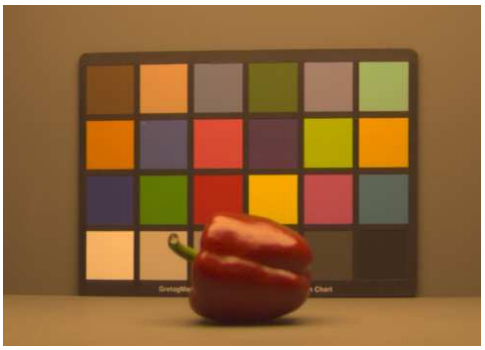

Figure 8. *Logvinenko Metamer image rendered under illuminant A.*

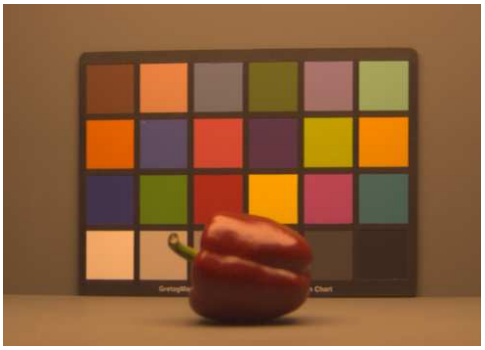

Figure 9. *Original image rendered under illuminant F12.*

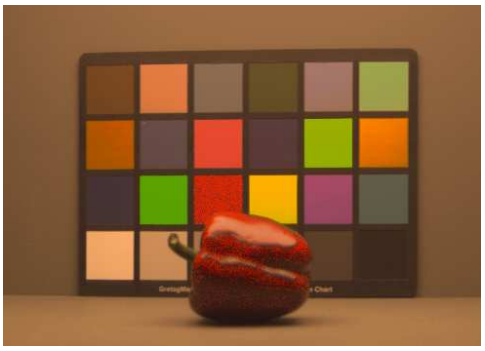

Figure 10. *Logvinenko Metamer image rendered under illuminant F12.*

sampling	ΔE				$\Delta\theta$		timings [s]	
	mean	med	95 pt.	max	mean	max	build	query
1nm, 405-650	0.5	0.4	1.5	4.3	0.2	1.8	0.2	0.01
0.5nm, 405-650	0.3	0.2	0.7	1.6	0.1	0.9	0.7	0.02

Table 2. ΔE and angular errors for the recovered Logvinenko metamers under D65 illuminant using the k-d tree indexing. The timings given are for the building and querying of the k-d tree.

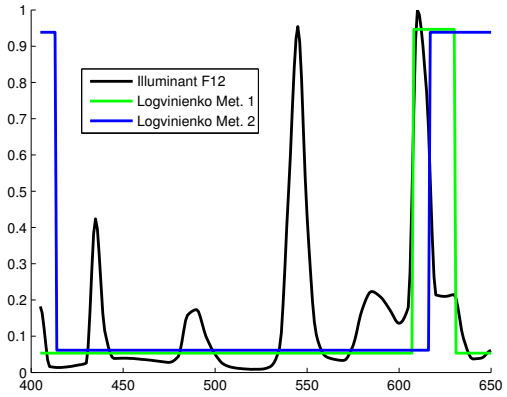

Figure 11. *Logvinenko Metemers calculated for two adjacent pixels from the 'noisy' red patch in Figure 10 and plotted together with the illuminant F12 spectrum.*

Discussion

Of course, we are in effect exploiting a similar observation to Godau and Funt. We sample the set of optimal spectra and then find the α terms as a second step. But, while we stand on their shoulders for inspiration, our approach is different. Our experiments show we do not need to interpolate between local optimal surfaces if we generate enough local surfaces in the first place. We are the first to concede that we need to generate many optimal surfaces ($> 200,000$ for the finest sampling and the best ΔE result). Yet, this simple combinatorial approach has the advantage that finding α is a simple closed form process (which uses only simple arithmetic operations). It is remarkable that our brute force solution is so much faster than the Godau and Funt's approach. And, the explanation for this is the underlying simple arithmetic form of our calculations. By making the final indexing step comprising the usage of the k-d tree, we made the Logvinenko metamer calculation, for the first time, a real practical tool useful for image analysis.

The calculations were performed on Dell laptop with Intel i7 processor. We did not use any parallel computation functionalities which are available in Matlab. In particular one could envisage that using general processing GPU could significantly speed up the calculations of the ADL coordinates (for our first method).

Finally, we remark that Finlayson et al adopted a similar combinatorial approach as we use here for the problem for finding the optimal sensor basis (a linear combination of the cones) which supports simple von Kries type adaptation [2]. In this work, sensors (linear combinations of the cones) are represented by a discrete number of points on a sphere. Candidate sharp sensors are triplets of this discrete sensor sets. Finlayson et al showed that they could, using the simplest of combinatorial optimisation, generate all possible candidate linear transforms of the cones and test their adequacy as a vehicle for von Kries adaptation. Like here, the solution to the optimisation was simple enough to be searched by a simple brute force combinatorial search. We predict that there are many other problems in the colour field that can be tackled in this combinatorial way.

Conclusions

In this paper, we proposed a simple, yet very efficient way of calculating Logvinenko coordinates for large databases. The method produces approximate results which should be sufficient for majority of applications. The method is efficient enough to be applied in general photography using desktop software.

Acknowledgements

The authors would like to thank UK EPSRC for supporting this research (grants: EP/H022236/1 and EP/H022325/1).We would like to thank Stuart Crichton from the IoN, Newcastle University for capturing the hyperspectral image used in this work.

References

[1] J.L. Bentley. Multidimensional binary search trees used for associative searching. *Commun. ACM*, 18(9):509–517, September 1975.

[2] G.D. Finlayson, M.S. Drew, and B.V. Funt. Spectral Sharpening: Sensor Transformations for Improved Color Constancy. *Journal of the Optical society of America*, 11(5):1553–1563, 1994.

[3] G.D. Finlayson and P. Morovic. Metamer Constrained Colour Correction. In *Proceedings of the 7th Color and Imaging Conference (CIC)*, 1999.

[4] G.D. Finlayson and P. Morovic. Metamer Sets. *JOSA A*, 22(5):810–819, 2005.

[5] C. Godau and B. Funt. XYZ to ADL: Calculating Logvinenko's Object Color Coordinates. In *Proceedings of the 18th Color and Imaging Conference (CIC)*, 2010.

[6] A.D. Logvinenko. An object-color space. *Journal of Vision*, 9(11):1–23, 2009.

[7] M. Mackiewicz, S. Crichton, S. Newsome, R. Gazerro, G. Finlayson, and A. Hurlbert. Spectrally tunable led illuminator for vision research. In *Proceedings of the 6th Colour in Graphics, Imaging and Vision (CGIV)*, Amsterdam, Netherlands, April 2012.

[8] University of Eastern Finland Color Group. Spectral database. online. retrieved: feb, 2012.

[9] R. Oldenhuis. Optimize for matlab. http://www.mathworks.com/matlabcentral/fileexchange/24298-optimize.

[10] E. Schrödinger. Theorie der Pigmente von größter Leuchtkraft. *Annalen der Physik*, 367(15):603–622, 1920.

[11] G. Wyszecki and W.S. Styles. *Color Science: Concepts and Methods, Quantative Data and Formulae*. John Wiley and Sons, NY, 1982.