Non–Iterative Minimum ΔE Gamut Clipping

Ján Morovic and Pei–Li Sun
Colour & Imaging Institute, University of Derby
Derby, United Kingdom

Abstract

The present paper presents a framework for implementing various minimum ΔE gamut clipping algorithms in a non–iterative way. While the approach presented here has Euclidean distance at its heart, a way of applying it to advanced colour difference formulæ (e.g. ΔE_{94}, ΔE_{2000}) is also presented. The approach is exactly applicable to ΔE equations that can be expressed as applying weights to the orthogonal coordinates of a colour space on an individual basis. A method for approximating other ΔE equations is also introduced. Having such a solution is advantageous due to being relatively fast as well as very accurate.

Introduction

Communicating colour image information between different media almost inevitably requires a way for overcoming the differences in their colour gamuts – a gamut mapping algorithm (GMA). As this communication is one that is virtually ubiquitous in the digital world, it would be beneficial to have a method that solves this issue well under a wide range of circumstances and in a speedy way. While a significant effort has been invested in researching the former aspect of the issue (i.e. the search for a universally applicable algorithm), there is far less work published on the issue of efficient implementation. The aim of this paper is therefore to propose an accurate, fast and non–iterative implementation strategy suitable for a wide range of gamut clipping algorithms. As such this paper neither intends to suggest a new solution to the gamut mapping problem, nor make claims about the suitability of existing approaches but to look at the implementation of gamut clipping from the point of view of efficiency, speed and accuracy.

Iterative Methods

A range of iterative solutions exist to the gamut clipping problem and two of the most common ones will be briefly introduced next.

Convergent Searching in 3D Colour Space

This technique was originally proposed for use in an inverse medium characterisation model – i.e. a model that predicts device–dependent values (e.g. RGB, CMYK, etc.) for given device–independent (DI) inputs (e.g. XYZ, LAB, etc.).[1] In addition to this application, it can also be used for minimum distance gamut clipping where it would consist of the following steps :

1. Input original DI_O value for which minimum ΔE reproduction value (DI_R) is to be found and provide a central point in the reproduction medium's device–dependent colour space – DD_C.
2. Choose n^k points (DD_i) surrounding DD_C with an interval of w, where k is the dimension of the DD space (i.e. 3 for RGB, 4 for CMYK) and n ($n \geq 2$) determines how many samples are to be taken from the cube (or hyper–cube) that is centred around DD_C and has sides with a length of $(n–1)w$.
3. Calculate the DI_i values corresponding to the n^k DD_is from step 2 by using the reproduction's forward characterisation model.
4. Calculate ΔEs between DI_O and each DI_i and save that pair of DI_i and DD_i values which resulted in the smallest ΔE.
5. If w is smaller than 1 (i.e. the quantisation level of the DD space is reached) then stop and the DI_i value saved in step 4 is the solution. Otherwise, make the DD_i value saved in step 4 the centre (C), reduce w by a factor of r and repeat steps 2 to 5.

This method suffers from time–consuming convergent searching as well as the possibility that the convergence is towards a region that will result in a local rather than the global minimum. The use of smaller w intervals can reduce the error but at the cost of increasing the time it takes for finding the solution. A compromise that balances both the requirement for speed and accuracy would be to ensure that $(n–1)w$ is more than twice as large as the w interval of the last convergent search – i.e. this would result in each iteration being based on more than just four points from the previous step.

The advantages of this method are that any colour difference formula can easily be used for the mapping and that DD values are obtained for the minimum ΔE colour at the same time as finding that colour itself.

Convergent Searching on Medium Gamut Boundary

Convergent searching on a medium's gamut boundary (GB) in DI space is an alternative to searching the entire gamut and it significantly reduces the number of candidate DI_is taken into account. Pre–searching techniques can be used for both these methods to speed up the process.

Proposed Non–Iterative Method

The principal idea of the non–iterative strategy for minimum ΔE gamut clipping proposed here is to calculate the intersection P_R between the gamut surface and that of its normals which passes through the colour P_O for which a reproduction (i.e. P_R) is to be found. Furthermore, it is proposed that this be implement in a way where all of the gamut's normals as well as any other P_O–independent entities are pre–computed. The advantages of such a method are a reduction in the time required for finding the solution and an increase in the accuracy thereof. To perform this idea, the gamut surface has to be defined first. Note also that CIELAB will be used as the DI colour space here and that any other DI colour space could be used instead.

Geometry of a Gamut Boundary

A medium's gamut boundary in a DI colour space (e.g. CIELAB, CIECAM97s) can be described using a range of methods, including simply the use of DI measurements of colours from the surface of the gamut in DD space as well as the segment maxima gamut boundary descriptor (SMGBD).[2] Once a set of points representing the gamut boundary are determined, they can then be triangulated (e.g. using Delaunay triangulation,[9] or some other method if the points have an inherent structure as is the case with SMGBD) to form a polyhedron (Figure 1). The boundary can therefore be described by points, line segments and triangles.

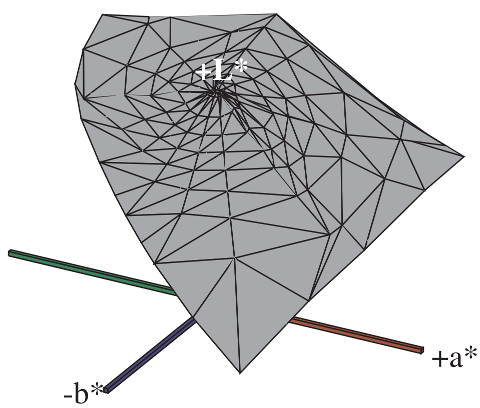

Figure 1. Medium GB triangulated in CIELAB space.

Determining Whether P_O Is Inside Reproduction Gamut

Given that clipping algorithms only change those P_O colours that are outside the reproduction's colour gamut and leave all other colours unchanged (i.e. P_R=P_O for in–gamut colours) it is first necessary to know whether the original colour P_O is outside the reproduction medium gamut. One way for determining whether a point is inside a given gamut is to first divide the whole gamut into tetrahedra determined by a triangle from the gamut's surface (as described previously) and a colour from within the gamut (C) (e.g. (L,a,b) = (50,0,0)). If the three points of the triangle P_1, P_2 and P_3, the centre C and P_O can fulfil Equation 1, then P_O is inside the tetrahedron.[3]

$$\begin{bmatrix}\alpha\\ \beta\\ \gamma\end{bmatrix}=\begin{bmatrix}L_1^*-L_C^* & L_2^*-L_C^* & L_3^*-L_C^*\\ a_1^*-a_C^* & a_2^*-a_C^* & a_3^*-a_C^*\\ b_1^*-b_C^* & b_2^*-b_C^* & b_3^*-b_C^*\end{bmatrix}^{-1}\begin{bmatrix}L_O^*-L_C^*\\ a_O^*-a_C^*\\ b_O^*-b_C^*\end{bmatrix}$$
$$\alpha\ge 0,\ \beta\ge 0,\ \gamma\ge 0,\ \text{and}\ \alpha+\beta+\gamma\le 1 \qquad (1)$$

Determining whether P_O is inside the gamut can be done by checking all of the tetrahedra that comprise it. If none of them contain P_O, it is an out–of–gamut colour and needs to be clipped. Another way of doing this is to have a line determined by a point inside the colour gamut (preferably near its centre) and P_O and to intersect this with the gamut boundary (e.g. using the FSLGB method[10]). Then if the P_O is nearer the central point than the gamut boundary, then it is inside the gamut and P_R=P_O. Further alternatives can be found in Preparata and Shamos.[9]

Mapping Geometry

Before considering the full 3D case, it is useful to look at minimum distance mapping in 2D. Here an out–of–gamut colour P_O can either be mapped to the intersection of two neighbouring line gamut boundary segments (i.e. one of the GB's vertices – a point) or the orthogonal projection of P_O onto a line gamut boundary segment (Figure 2b). Note that that there can be more than two solutions to this projection in the case of concave gamuts (Figure 2a). In 3D a similar geometry can be observed (Figure 3b) in that an out–of–gamut colour will map to one of three cases: (a) one of the vertices of the GB, (b) an orthogonal projection of P_O onto a line connecting two of the vertices of a GB triangle or (c) onto a point inside such a triangle. Hence gamut clipping can occur either onto a GB point, a GB line or a GB plane. Finding the minimum ΔE point on the GB in turn involves the calculation of minimum ΔEs for each of the three cases involves finding the points that have the smallest colour differences when considering points, lines or planes and then selecting that one of them that has the smallest ΔE. The next three sections will discuss how these three intermediate points are obtained.

Mapping Onto Points

The GB point that is closest to P_O can be obtained by calculating the colour difference between P_O and all the GB vertices using Equation 2. That GB vertex which has the smallest ΔE_V will then become the first candidate for the reproduction colour P_R. In Equation 2, (V_L, V_a, V_b) are weights which allow for different levels of importance to be given to the dimensions of the device independent colour space. Setting them to (1, 1, 1) results in the equation giving Euclidean distance in the colour space.

$$\Delta E_V=\sqrt{\left(\frac{\Delta L^*}{V_L}\right)^2+\left(\frac{\Delta a^*}{V_a}\right)^2+\left(\frac{\Delta b^*}{V_b}\right)^2} \qquad (2)$$

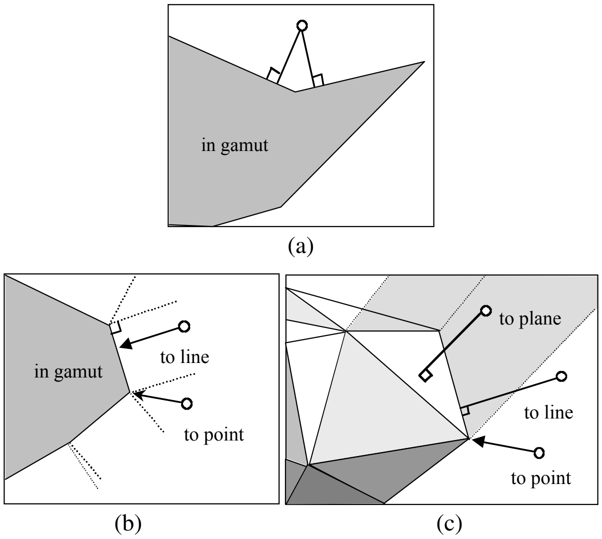

Figure 2. (a) 2D mapping geometry with two or (b) one solutions and (c) 3D mapping geometry.

Mapping Onto Lines

The second candidate for P_R is obtained by dividing the GB into a number of line segments, orthogonally projecting P_O onto these and finding that projection which has the smallest ΔE_V relative to P_O. For a projection line $\overline{P_O P_R}$ and a line segment $\overline{P_1 P_2}$ to be orthogonal, the dot product of vectors $\overrightarrow{P_O P_R}$ and $\overrightarrow{P_1 P_2}$ needs to equal zero. The P_R that satisfies this condition can be obtained using Equation 3 whereby the T parameter is obtained by substituting the right side of Equation 3 for the $[L^*_R, a^*_R, b^*_R]$ vector in Equation 4 and solving it.

$$\begin{bmatrix} L^*_R \\ a^*_R \\ b^*_R \end{bmatrix} = \begin{bmatrix} L^*_1 \\ a^*_1 \\ b^*_1 \end{bmatrix} + T \begin{bmatrix} L^*_2 - L^*_1 \\ a^*_2 - a^*_1 \\ b^*_2 - b^*_1 \end{bmatrix} \tag{3}$$

$$0 = \left[V_L \left(L^*_R - L^*_O \right) \; V_a \left(a^*_R - a^*_O \right) \; V_b \left(b^*_R - b^*_O \right) \right] \cdot \begin{bmatrix} L^*_2 - L^*_1 \\ a^*_2 - a^*_1 \\ b^*_2 - b^*_1 \end{bmatrix} \tag{4}$$

Note that the sum of the lengths of the line segments between P_1, P_2 and P_R must equal the length of the P_1P_2 line segment, i.e.,

$$\left\| \overline{P_1 P_R} \right\| + \left\| \overline{P_R P_2} \right\| = \left\| \overline{P_1 P_2} \right\|$$

for P_R to be inside the line segment and a candidate for the minimum ΔE_V solution.

Mapping Onto Planes

To calculate the third candidate for P_R, the orthogonal projections of P_O are calculated onto the planes determined by the GB's triangles and only those inside the triangles are taken into account. The function of a plane can be expressed as in Equation 5 and the orthogonal projection of P_O onto it can be obtained from Equation 6.

$$f(L^*, a^*, b^*) = c_L \cdot L^* + c_a \cdot a^* + c_b \cdot b^* + c_0 \tag{5}$$

$$\begin{bmatrix} L^*_R \\ a^*_R \\ b^*_R \end{bmatrix} = \begin{bmatrix} L^*_O \\ a^*_O \\ b^*_O \end{bmatrix} - T \begin{bmatrix} V_L c_L \\ V_a c_a \\ V_b c_b \end{bmatrix} \quad \text{where} \; T = \frac{f\left(L^*_O, a^*_O, b^*_O\right)}{V_L c_L^2 + V_a c_a^2 + V_b c_b^2} \tag{6}$$

Once the projection onto a triangle's plane is obtained, it needs to be seen whether it is inside the triangle (determined by points P_1, P_2 and P_3) as it is only such a projection that is a candidate for the solution. This can be done by calculating the area of triangle $P_1P_2P_3$ and then seeing whether the areas of triangles $P_RP_1P_2$, $P_RP_2P_3$ and $P_RP_1P_3$ add–up to the same value.

Pre–Computing Shared Entities

As could be seen from the description of how the three candidate P_Rs are obtained, there is a significant amount of computation that to take place without reference to a particular P_O. It is therefore highly advisable to pre–compute them and the following is their list:

1. *Line case*: equations of all GB lines and their normals; lengths of each GB line segment.
2. *Plane case*: c_L, c_a, c_b and c_0 coefficients of each plane; area of each triangle.

Dividing the calculations involved in obtaining the minimum ΔE_V solution into ones that are dependent on P_O and ones that are not and then pre–computing the latter results in a significant increase in the speed of finding the solution.

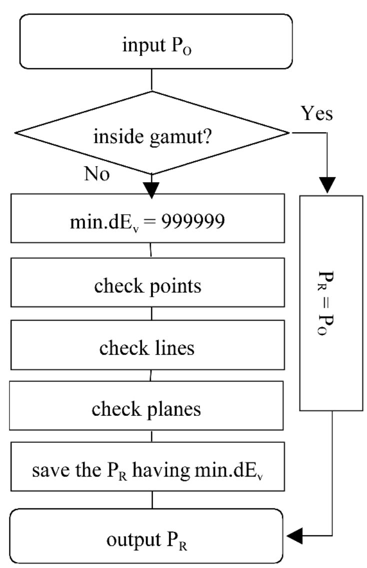

Figure 3. Flow chart of proposed algorithm.

Work Flow

The work flow of the proposed algorithm is illustrated in Figure 3. A very large initial value is chosen for the minimum ΔE_V variable and ΔE_Vs obtained by checking the three cases (points, lines and planes) are compared to it. Whenever a smaller ΔE_V value is encountered, P_R is set to the corresponding colour and the minimum ΔE_V variable is updated.

Note, that it is possible to find more than one GB triangle containing a candidate P_R and this is due to the possibility of candidates coming from other side of the gamut or due to concavities of the gamut boundary. In order to speed up the process, one can check only a part of the points, lines and planes of the GB by using ΔEs obtained from them as parameters to a termination condition.

Using Advanced Colour Difference Formulæ

In the pervious sections, all the equations used for finding P_R were geared towards finding the minimum Euclidean distance from P_O. Recent studies, however, show that gamut clipping using weighted ΔE formulae[4] including CIE ΔE_{94} gives more accurate reproductions. It is therefore necessary to understand whether it is possible to use these weighted ΔE formulæ in the framework outlined above. In essence, the framework is designed for Euclidean distance but other metrics can be used if they can be expressed in terms of scaling the individual orthogonal dimensions of the DI colour space used. The following sections will suggest ways of how to do this for some chosen ΔE formulæ.

ΔE_{wt}(1:2:2)

The generic weighted ΔE formula for gamut clipping is shown in Equation 7 and studies[5,6] suggest that the most accurate reproductions can be obtained when the $(K_L{:}K_C{:}K_H)$ coefficients are set to (1:2:1) or (1:2:2). Referring to Equations 2 to 6, if $K_C=K_H$, the present framework can be used by determining the (V_L, V_a, V_b) weights from Equation 2 as follows: $K_L=V_L$ and $K_C=V_a=V_b$. When V_L is not equal to V_a and V_b, the normal vectors of line segments and triangle planes will simply be perturbed and Euclidean distance is calculated in the perturbed colour space. In this case the conversion between (K_C,K_H) and (V_a,V_b) is easy, however, if K_C is not equal to K_H, the conversion becomes very difficult as ΔH^* cannot be as an overall scaling of the orthogonal dimensions.

$$\Delta E_{wt} = \sqrt{\left(\frac{\Delta L^*}{K_L}\right)^2 + \left(\frac{\Delta C^*}{K_C}\right)^2 + \left(\frac{\Delta H^*}{K_H}\right)^2} \qquad (7)$$

Simulated CIEDE94 and CIEDE2000

Because the $[K_C,K_H]$ weights of CIE ΔE_{94}[7] and CIE ΔE_{2000}[8] are difficult to convert to $[V_a,V_b]$, a least squares error technique can be used to optimise the $[V_a,V_b]$ weights so as to approximate these advanced ΔE formulae. One way of doing this is using the following steps:

1. Let $V_a = 1 + X \cdot \overline{C^*}$ and $V_b = 1 + Y \cdot \overline{C^*}$ where X and Y are unknown parameters to be optimized.
2. Implement one of the iterative minimum ΔE GMAs that can provide reliable results using the given colour difference formula.
3. Sample colour space outside the reproduction gamut with equal L*a*b* intervals, gamut-map the samples using the GMA from step 2 and record L*a*b* values of each (P_O, P_R) pair.
4. Use the least squares error technique to minimize the sum squared errors between the actual and our simulated ΔEs for each sample pair by optimizing the X and Y parameters.

To improve the accuracy of the simulated ΔEs, the X and Y parameters can be optimised individually for individual hue regions. The final hue dependent parameters, X_h and Y_h, can be obtained by interpolating between the optimised parameters of the two neighbouring hue regions nearest to the $\overline{h}$ of the (P_O, P_R) pair. The simulated ΔE_{94}(1:1:1) formula can therefore be expressed as Equation 8.

$$\Delta E_{94} \cong \sqrt{\left(\frac{\Delta L^*}{1}\right)^2 + \left(\frac{\Delta a^*}{1+X_h\overline{C^*}}\right)^2 + \left(\frac{\Delta b^*}{1+Y_h\overline{C^*}}\right)^2} \qquad (8)$$

The (V_L,V_a,V_c) vector for Equations 2 to 6 is therefore:

$$[V_L, V_a, V_b] = \left[1, \left(1+X_h\overline{C^*}\right), \left(1+Y_h\overline{C^*}\right)\right] \qquad (9)$$

In these formulae, $\overline{C^*}$ is defined as the mean of C^* values for P_O and P_R. However, as P_R is unknown when checking line segments and triangle planes, a solution is to use the mean C^* of the vertices determining the line segment or the triangle. Note that ΔE_{94} uses the geometric mean whereas ΔE_{2000}(1:1:1) uses the arithmetic mean and applies it to $\overline{L^*}$ and $\overline{C^*}$. Otherwise ΔE_{2000}(1:1:1) parameters can be calculated in an analogous way.

Although the structure of CIE ΔE_{2000} is more complex than that of ΔE_{94}, the same method can be used for simulating its use. The simulated ΔE_{2000} colour difference formula is shown in Equation 10.

$$\Delta E_{00} \cong \sqrt{\left(\frac{\Delta L^*}{S_L}\right)^2 + \left(\frac{(1+G)\Delta a^*}{1+X_h\overline{C^*}}\right)^2 + \left(\frac{\Delta b^*}{1+Y_h\overline{C^*}}\right)^2}$$

$$\text{where } S_L = 1 + 0.015\left(\overline{L^*}-50\right)^2 + \sqrt{20+\left(\overline{L^*}-50\right)^2} \qquad (10)$$

$$G = 0.5\left(1 - \sqrt{\overline{C^*}^7 \div \left(\overline{C^*}^7 + 25^7\right)}\right)$$

The (V_L,V_a,V_c) vector for ΔE_{2000} in Equations 2 to 6 is therefore:

$$[V_L, V_a, V_b] = \left[S_L, \left(\left(1+X_h\overline{C^*}\right) \div (1+G)\right), \left(1+Y_h\overline{C^*}\right)\right] \qquad (11)$$

At this stage it is important to know the magnitude of the difference between using either the actual ΔE formulae or their simulations. A test has therefore been conducted for the mapping from CRT to prints under D65 illumination and this was done in the following way:

1. Send colours with 10 LAB unit intervals covering both the CRT and printer gamuts to the gamut clipping algorithm using a simulated colour difference formula.
2. Record P_O and the P_R for out-of-gamut colours only.
3. Calculate the colour differences using both simulated and actual colour difference formulae and compare the results between the two data sets.

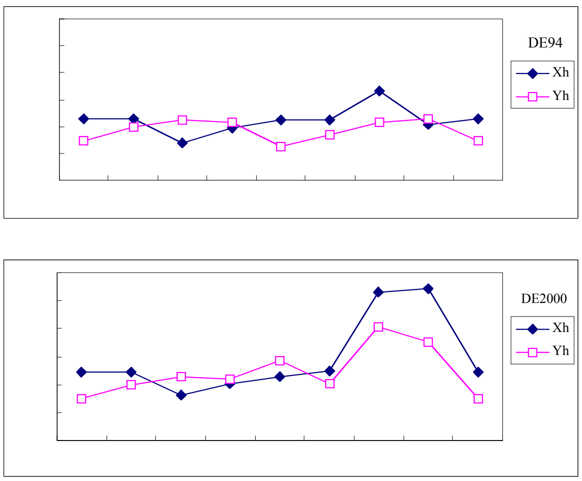

Figure 4. Optimised X_h and Y_h parameters for ΔE_{94} and ΔE_{2000}.

Both the global and hue dependent methods for optimizing (X,Y) parameters were used. In the case of the global method, the (X,Y) parameters were (0.042,0.044) and (0.083,0.064) for ΔE_{94} and ΔE_{2000} respectively. In the hue dependent case, colour space was divided into 8 hue regions with 45 degree intervals. The optimized parameters-(X_h,Y_h) are shown in Figure 4. As can be seen, a strong hue dependent effect – particularly in the blue region – was shown for the simulated ΔE_{2000} and this correlates with one of the missions of the ΔE_{2000} development which was to correct the errors in that region.

Table 1. Differences between simulated (sim.) and actual (act.) ΔE for out–of–gamut colours.

colour difference		ΔE_{94}			ΔE_{2000}		
		mean	95th	max.	mean	95th	max.
act.		9.66	21.04	30.81	8.39	16.61	23.12
|act.-sim.|	global	0.20	0.66	1.26	1.70	5.11	6.47
	hue dep.	0.15	0.52	1.07	0.84	3.63	5.69

The results of the error test are shown in Table 1. In that table, the 'act.' row shows the statistics of colour differences for the out-of-gamut colour mapping. As can be seen (when compared with the magnitude of 'act.'), both the global and hue dependent methods worked well for ΔE_{94} but not so well for ΔE_{2000}. The possible reason is that ΔE_{2000} deals with region dependent rotation of colour difference ellipses and as a result it is difficult to apply simple parameters to (a^*,b^*) co-ordinates for the simulation.

Performances of the Iterative and Non-Iterative Methods

A test was conducted to evaluate the performances of the iterative and non-iterative methods mentioned previously. Since the methods involve many different parameters and as they cannot be compared directly, they will not be introduced in detail. Only the number of candidates with which ΔE was computed before finding the minimum will be shown as a reference.

Convergent searching in 3D colour space (3D conv.) was tested at three levels of speed. The slowest one involved 2×10^6 candidates for each input colour and started with a search interval of about 1 LAB unit. Compared with the slowest one, the fastest method for 3D conv. started with an interval of about 25 LAB units and required the checking of 875 candidates before completing the search.

Convergent searching on the medium gamut boundary (GB conv.) was tested at three speed levels too. The slowest one involved 8,311 candidates and started with a search interval of about 2.5 LAB units and the fastest GB conv. started with an interval of about 25 LAB units and required 212 candidates for completing the search.

The non–iterative method used a 16 × 16 GBD and it is based on this that the number of candidate colours was determined. It equals the sum of the vertices and the P_R candidates from projecting P_O onto the GBD's line segments and triangles – this number is therefore variable and has a minimum equal to the number of GBD colours (in this case 256).

Next, the comparison of the performances of the various clipping techniques used is not a straight–forward task as each of them has a different gamut boundary descriptor (sometimes implicit). Differences between the techniques would therefore also include this rather than only the differences in finding minimum ΔEs. To see how big the differences are between the slowest version of the iterative techniques and the non–iterative technique, Table 2 shows the statistics of the differences between the P_R values given by the various methods. The colours that are inside a CRT gamut but outside the gamut of prints under D65 illumination with an interval of 10 LAB units were regarded as the input data for the test.

Table 2. Comparison of Iterative and Non-Iterative Methods.

ΔE	mean	std.	95th	max.
3D v GB	2.12	2.28	6.19	23.98
GB v non	0.99	1.39	3.81	16.09
3D v non	1.93	2.06	5.55	23.98

As can be seen the majority of differences between the various techniques are not very big even though their maxima are quite substantial.

To take a closer look at the iterative techniques, the errors due to increasing the starting search interval were also evaluated by calculating ΔEs between the slowest and the faster versions. The speed of calculation on a 400 MHz Pentium II PC (in terms of colours per second) was also determined and both these results can be seen in Table 3. In terms of speed, it's important to understand that it depends on the implementation of the algorithms. For instance, the speed for the non-iterative method is inversely proportional to the LUT size of the gamut boundary descriptor. As can also be seen, iterative methods have a risk of providing quite different results when increasing the starting interval in order to speed the process. This is due to them sometimes converging on a local rather than the global minimum. The non-iterative method is therefore recommended for the minimum ΔE gamut clipping as it is both relatively fast and as it always results in the global minimum ΔE.

Table 3. Speed and Influence of Starting Search Interval.

method	No. of candidates	Speed (colours/s)	error (ΔE)		
			mean	std.	max.
3D conv.	2.1×10^6	0.09	-	-	-
	3.6×10^4	5.26	0.03	0.45	8.28
	875	217.39	0.26	1.79	15.77
GB conv.	8,311	0.44	-	-	-
	653	5.26	0.20	1.48	22.92
	212	15.87	0.20	1.49	22.92
non-iter.	<300	47.62	-	-	-

Hue–Preserving Minimum ΔE Version

A hue–preserving version of the proposed algorithm can be achieved by only using the 2D mapping referred to in Figures 2a and 2b. In that algorithm, only points and 2D line segments are checked and they can be obtained by intersecting the GB with a plane that has the hue angle of P_o. The reasons for introducing a hue–preserving version are that the human visual system is more sensitive to hue shifts (this is also supported by the fact that preferred ΔE weights normally give most importance to ΔH^*) and also that it is faster.

Conclusions

As gamut clipping is a technique that is a key element of cross–media colour image reproduction, it is important to have an implementation of it that is efficient, accurate and fast. The non–iterative gamut clipping framework proposed in this paper is intended to make the use of this popular gamut mapping approach less time consuming and more accurate as it always results in the global minimum ΔE between a given out–of–gamut colour and the colours of a target gamut. A method for using this non–iterative framework for advanced colour difference formulæ has also been introduced and results were shown of the speed and accuracy of various iterative and non–iterative methods. Using the non–iterative method described here will result in a relatively fast and very accurate calculation of minimum ΔE gamut clipping results.

References

1. M.C. Stone, W. B. Cowan and J. C. Beaty (1988) Color Gamut Mapping and the Printing of Color Images, *ACM Trans. on Graphics*, 246–292.
2. J. Morovic and M. R. Luo (1999) Developing Algorithms for Universal Colour Gamut Mapping, *Colour Imaging: Vision and Technology*, L. MacDonald and M. R. Luo (eds.), John Wiley & Sons Ltd., 253–283.
3. P. C. Hung (1993) Colorimetric Calibration in Electronic Imaging Devices Using a Look–Up–Table Model and Interpolations, *Journal of Electronic Imaging*, **2/**1:53–61.
4. M. Ito and N. Katoh (1999) Three–dimensional Gamut Mapping Using Various Color Difference Formulae and Color Spaces, *Proc. of SPIE & IS&T Electronic Imaging '99 Conf.*
5. N. Katoh and M. Ito (1996) Gamut Mapping for Computer Generated Images (II), *Proc. IS&T/SID 4th Color Imaging Conf*, 126–129.
6. R. Y. C. Wei, M. J. Shyu and P. L. Sun (1997) A New Gamut Mapping Approach Involving Lightness, Chroma and Hue Adjustment, *TAGA Proc.*, 685–702.
7. CIE (1995) *Industrial Colour–Difference Evaluation*, CIE 116–1995..
8. M. R. Luo, G. Cui and B. Rigg (2001) The Development of the CIE2000 Colour Difference Formula: CIEDE 2000, *Color Res. Appl.*, **26**:000-000.
9. F. P. Preparata and M. I. Shamos (1985) *Computational Geometry An Introduction*, Texts and Monographs in Computer Science, Springer Verlag, 209-211.
10. J. Morovic and M. R. Luo (2000) Calculating Medium and Image Gamut Boundaries for Gamut Mapping, *Color Res. Appl.*, **25**:394-401.