Gernot Hoffmann

Gamut Visualizations and Out-of-gamut Distances

Contents

Settings for Acrobat

Edit / Preferences / General / Page Display (since version 6)
Custom Resolution 72 dpi / View by zoom 100% or 200%

Edit / Preferences / General / Color Management (full version only)
sRGB
Euroscale Coated or ISOCoated or SWOP
Gray Gamma 2.2

1. Introduction

A gamut is the whole set of colors which can be shown by a device. An RGB space and a CMY space can be visualized topologically by the same cube. Such a cube lacks physical relevance. Obviously there are many different RGB and CMY spaces. The cube can be mapped topologically as a somewhat distorted volume into the color space CIELab. ‚Topologically' means, there is certainly a well defined relation between each of the source spaces and the destination space CIELab. Such a relation does not exist between CMYK and CIELab. Black ink K introduces a fourth dimension for the source space.
The boundary surface of the mapped RGB or CMY cube in CIELab confines the gamut volume in physical coordinates, almost always for reference white D50, because print products have to be checked under D50.
The really executed mapping is here called functional mapping. An RGB cube can be mapped with arbitrary accuracy. A CMY cube is mapped using huge tables in ICC profiles. Based on interpolation, the resolution can be arbitrary, but the accuracy is limited by the stored data.
For CMYK the situation is different. The gamut consists in CIELab of e.g. 65536 discrete samples, which can be considered as a point cloud. Finding a plausible non-convex hull for such a cloud is the major goal of this investigation.
Once an arbitrary color is given in CIELab, it is possible to tell whether it is out-of-gamut, and to estimate then the out-of-gamut distance. This distance depends on an assumed gamut mapping algorithm – how to convert out-of-gamut colors into colors which can be shown by the device. In this sense there is not just one true out-of-gamut distance.
The demand for the calculation of out-of-gamut distances came during the evaluation of the authors ‚The Digital Munsell', a swatch book [24] for the well-known Munsell colors [19]. Each swatch has assigned values for CIELab, sRGB and out-of-gamut distances for sRGB and one common CMYK process, as defined by an ICC profile.
The sRGB out-of-gamut distance tells the viewer at the monitor, how reliable the color reproduction might be, if his monitor is calibrated for white point D65, gamma 2.2, and if the primaries are near to those of the sRGB standard.
The CMYK out-of-gamut distance tells the reader, how reliable the color reproduction might be in a printed version.

Note: AdobeRGB(1998) is abbreviated by aRGB

References by categories

Color Science
[1] to [8], [27]

Computer graphics, computer vision
[9], [10], [26]

Standard RGB (sRGB)
[11], [12]

ICC profiles, Offset printing
[13] to [15], [23]

ICC Profile Inspector
[16]

Black Point Compensation
[17]

Munsell colors system
[18], [19], [24]

CIE colorimetry, by the author
[20], [21]

PostScript graphics
[25]

2. Topological and functional mapping

The color cube has eight vertices, twelve edges and six faces. For an RGB working space like sRGB or Adobe RGB (1998) = aRGB, the vertices represent the primaries R,G,B, the secondaries C,M,Y, absolute black O and absolute white W. The color temperatures of all grays from black to white are defined by daylight D65 for sRGB and aRGB. This can be different for other spaces.
For a CMYK space like ISOCoated_v2_eci.icc the vertices represent the primaries or ink colors C,M,Y on the paper, the secondaries R,G,B, the darkest printable CMY black O and paper white W. So far, the black ink K is not involved.
Mapping either RGB or CMY (without K) by concept into another space, here CIELab, is called topological mapping. The cube appears by a new shape.
The transformation is executed by functional mapping. Getting from any RGB space to CIELab is a mathematical transformation. One does not need ICC profiles.
The transformation from CMY to CIELab is defined by huge multidimensional look-up tables (CLUTs) in ICC profiles, together with a couple of onedimensional functions [14], [23].
The CLUTs contain as well the ink K, but the influence of K cannot be explained by topological mapping. Nevertheless, an approximation can be found by a different kind of functional mapping, see chapter 4.5.
Thus we have for RGB or CMY either analytical functions or functions by tables. The topological mapping can be executed by functional mapping. It is then the question how to map the six faces of the cube from the source space to the six curved faces in the destination space CIELab. The volume is not convex but ‚weakly concave'. Therefore, each face has to be represented by n·n small squares in the source space, which leads to n·n almost flat quadrilaterals in CIELab. The number n is arbitrary for RGB spaces, for instance 25. It is predefined in CMYK ICC profiles by the number of gridpoints per channel CMY, for instance n=15 for 16 gridpoints.
Each destination quadrilateral is divided into two triangles. Each triangle has a normal which has to point generally into outside direction. The scene is observed by a synthetical camera. A triangle is visible in the graphic, if its normal points towards the camera. Otherwise it is invisible. This is called backface culling [9].
All illustrations were programmed by PostScript as vector graphics [25]. The visibility check by backface culling is theoretically correct only for (single) convex objects, but here it works mostly satisfying despite the concaveness. The comfortable z-buffer algorithm [9] for raster graphics cannot be used for vector graphics.
Each face has two edges which are handled by parameters p1, p2. For the face ORYGO in Fig. 2.1 one can use p_1 in OG direction **a** and p_2 in OR direction **b.**
The vector product **n**=**a**x**b** points into outside direction. This has to be defined for each face individually.

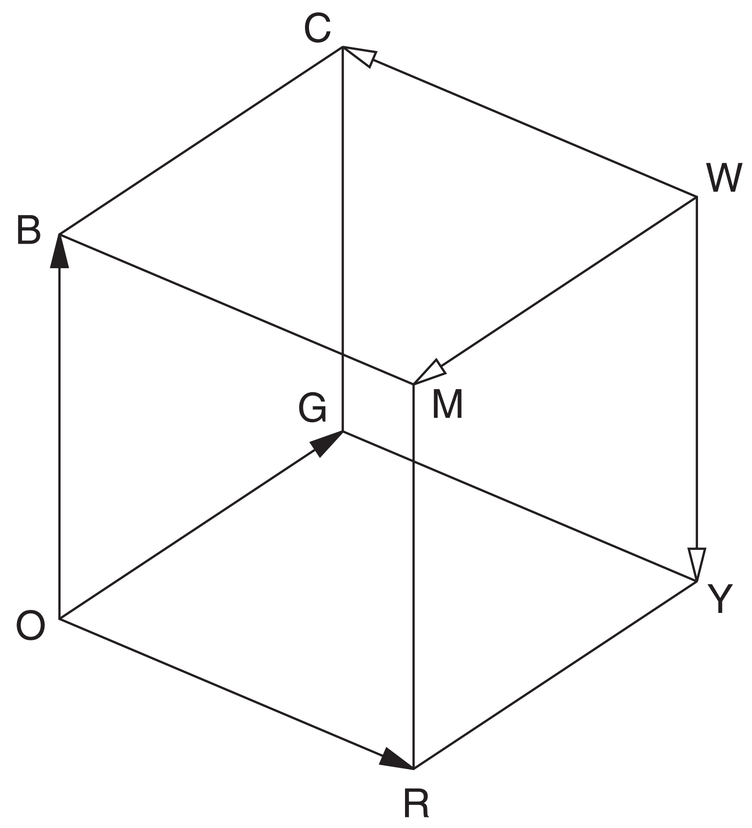

Figure 2.1 RGB and CMY color cube

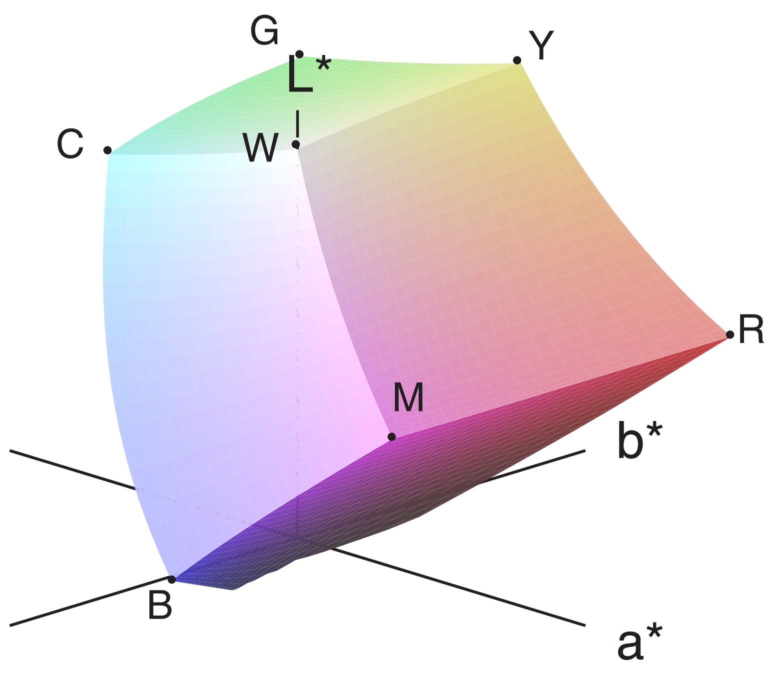

Figure 2.2 RGB cube in CIELab

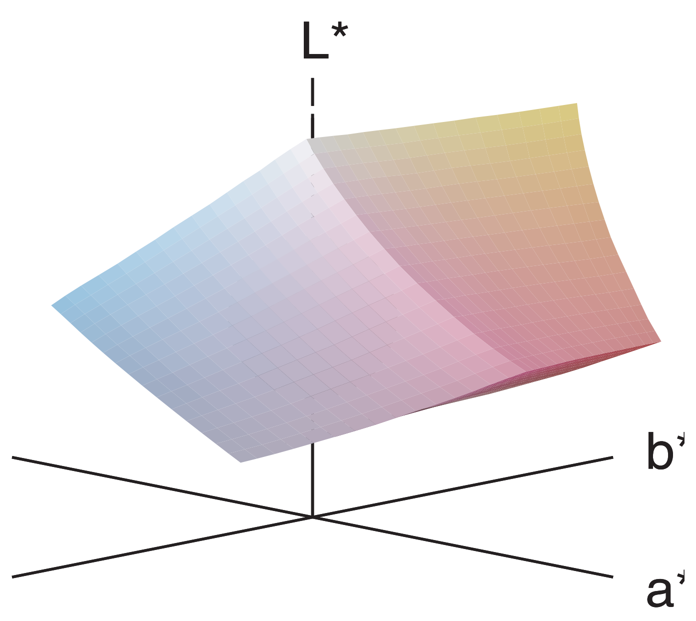

Figure 2.3 CMY cube in CIELab (no K)

3.1 Mapping RGB to CIELab / Concept

This is the complete transform from sRGB or other RGB spaces to CIELab. All the details, for instance the matrices $\mathbf{C}_{xr}$ and **B,** are found in [21]. The reference white for CIELab is typically D50. It is assumed that a print is viewed under D50 and that the observer is adapted to D50. The chromatic adaptation transform is required if the working space white is not D50. The spaces sRGB and aRGB have D65.

Generic gamma encoding, g=2.2, C=R,G,B

$$C = C'^{g}$$

sRGB gamma encoding, C=R,G,B

$$C = \begin{Bmatrix} C'/12.92 & \text{if } C' \leq 0.03928 \\ ((0.055 + C')/1.055)^{2.4} & \text{else} \end{Bmatrix}$$

RGB to XYZ column matrices (same white point D65, source)

$$\mathbf{X}_s = \mathbf{C}_{xr}\,\mathbf{R}$$

RGB to XYZ (new white point D50, destination)

Chromatic adaptation transform (CAT), use linearized Bradford transform by matrix **B** :

$$\mathbf{X}_d = \mathbf{B}\,\mathbf{X}_s$$

$\mathbf{X}_d$ to L*a*b* (reference white matrix $\mathbf{R}_w^{-1}$ for $\mathbf{X}_{w50}$)

$$\mathbf{X}_c = \mathbf{R}_w^{-1}\,\mathbf{X}_d = \begin{bmatrix} 1/X_{w50} & 0 & 0 \\ 0 & 1/Y_{w50} & 0 \\ 0 & 0 & 1/Z_{w50} \end{bmatrix} \mathbf{X}_d$$

$$X_1 = \begin{Bmatrix} X_c^{1/3} & \text{if } X_c > 0.008856 \\ 7.787\,X_c + 16/116 & \text{else} \end{Bmatrix}$$

$$Y_1 = \begin{Bmatrix} Y_c^{1/3} & \text{if } Y_c > 0.008856 \\ 7.787\,Y_c + 16/116 & \text{else} \end{Bmatrix}$$

$$Z_1 = \begin{Bmatrix} Z_c^{1/3} & \text{if } Z_c > 0.008856 \\ 7.787\,Z_c + 16/116 & \text{else} \end{Bmatrix}$$

$$\begin{aligned} L^* &= 116\,Y_1 - 16 \\ a^* &= 500\,(X_1 - Y_1) \\ b^* &= 200\,(Y_1 - Z_1) \end{aligned}$$

The six faces of the RGB cube are subdivided into 25x25 grid points which delivers 24x24 small squares in the source space. It turned out that the colors should be assigned by a square law to the indices in order to achieve a better spacing in CIELab. Additionally, the dark end of the volume should be rendered with higher resolution.

The next page shows eight views to the color spaces sRGB (small) and aRGB (large) in CIELab. The original graphics are vector graphics. The overlay was created as a raster graphic by Photoshop with 144ppi.

3.2 Mapping RGB to CIELab / sRGB and aRGB

Eight views to the color spaces sRGB (small) and aRGB (large) in CIELab. Optimal view by 72ppi with zoom 200%, see page 1 for the Acrobat settings.

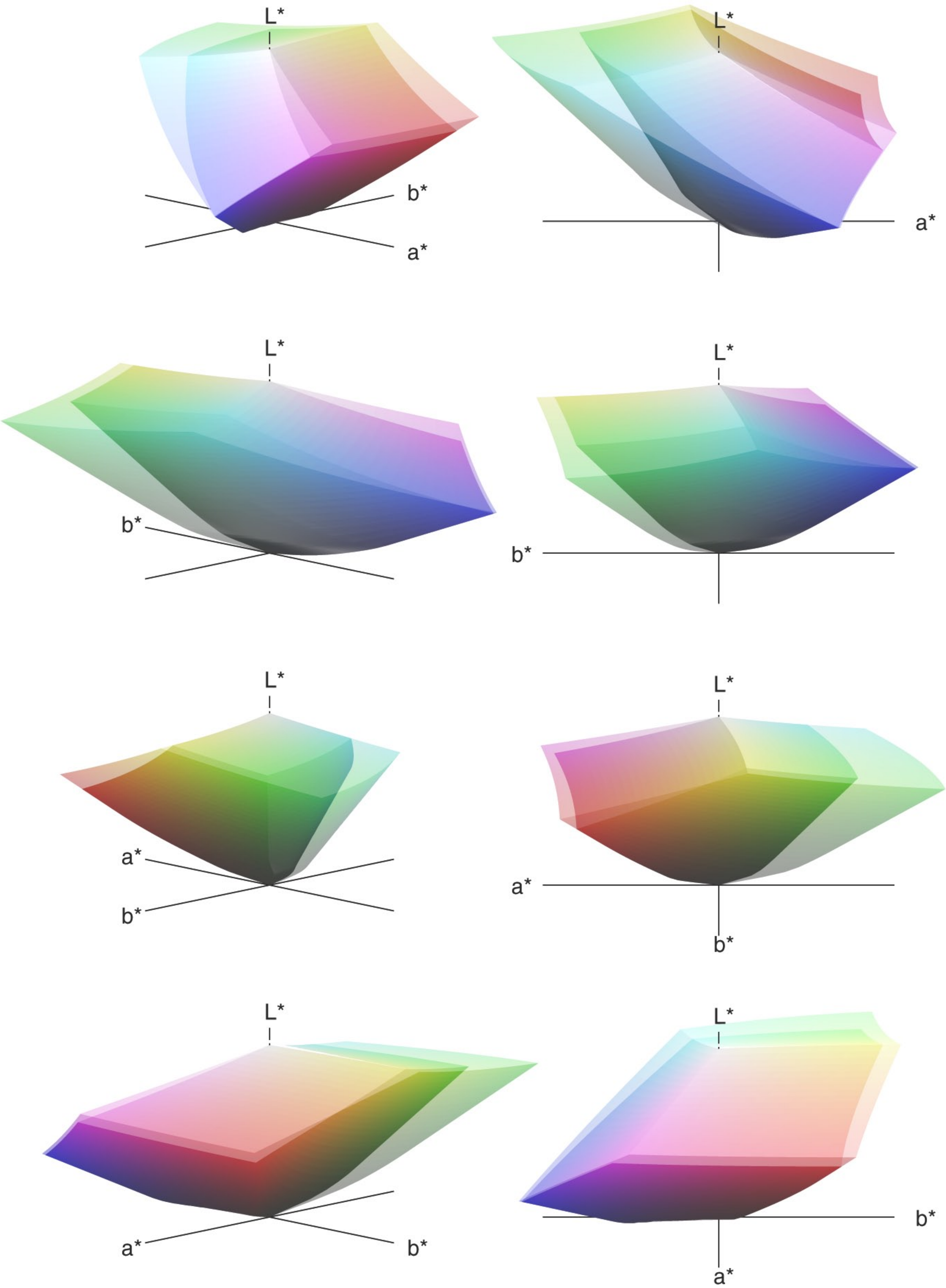

Figure 3.1 Gamuts for sRGB and aRGB in

3.3 Mapping RGB to CIELab / Out-of-gamut distances

A color is out-of-gamut for a certain RGB space, if at least one channel value C of R,G,B is less 0 or greater 255 (for 8 bits per channel). Out-of-gamut values C are normally clipped for [0...255] in advance to the application of gamma encoding. A power function like $C' = \mathrm{Round}(255\cdot(C/255)^{(1/g)})$ does not exist for negative values C.

An out-of-gamut distance is the Euclidian distance of an out-of-gamut color $\mathbf{C}_0$ from the gamut boundary, expressed in CIELab units. $\mathbf{C}_0$ is a color, as represented by three components L^*_0, a^*_0, b^*_0, whereas the value C, as used above, means R,G or B.
Out-of-gamut distances are useful for demonstrating whether a monitor preview of spot colors or Munsell colors is reliable. Calibrated standard monitors are near to sRGB. The distance calculation is not unique, but probably better than just the information ‚out-of-gamut‘.
Defining the distance from a gamut boundary requires besides the metric a gamut mapping strategy. Here we proceed as follows:

Draw a straight line from $\mathbf{C}_0$ to the point $\mathbf{P}_0$ at L*=50 on the lightness axis.
If $\mathbf{C}_0$ is out-of-gamut, then find a point $\mathbf{C}_1$ in-gamut by a coarse search along the line $\mathbf{C}_0\mathbf{P}_0$.
Then find the gamut boundary at $\mathbf{C}_2$ by a binary subdivision of the length $\mathbf{C}_1\mathbf{C}_0$.
The length of the line segment $\mathbf{C}_2\mathbf{C}_0$ is the out-of-gamut distance dE.
This search happens in a plane of constant CIELab hue. The illustration shows for sRGB the plane H=0 (that is along the axis a*).
The illustration shows as well the plane for H=180 on the other side.

More details and gamut boundaries for three RGB spaces can be found in [26].

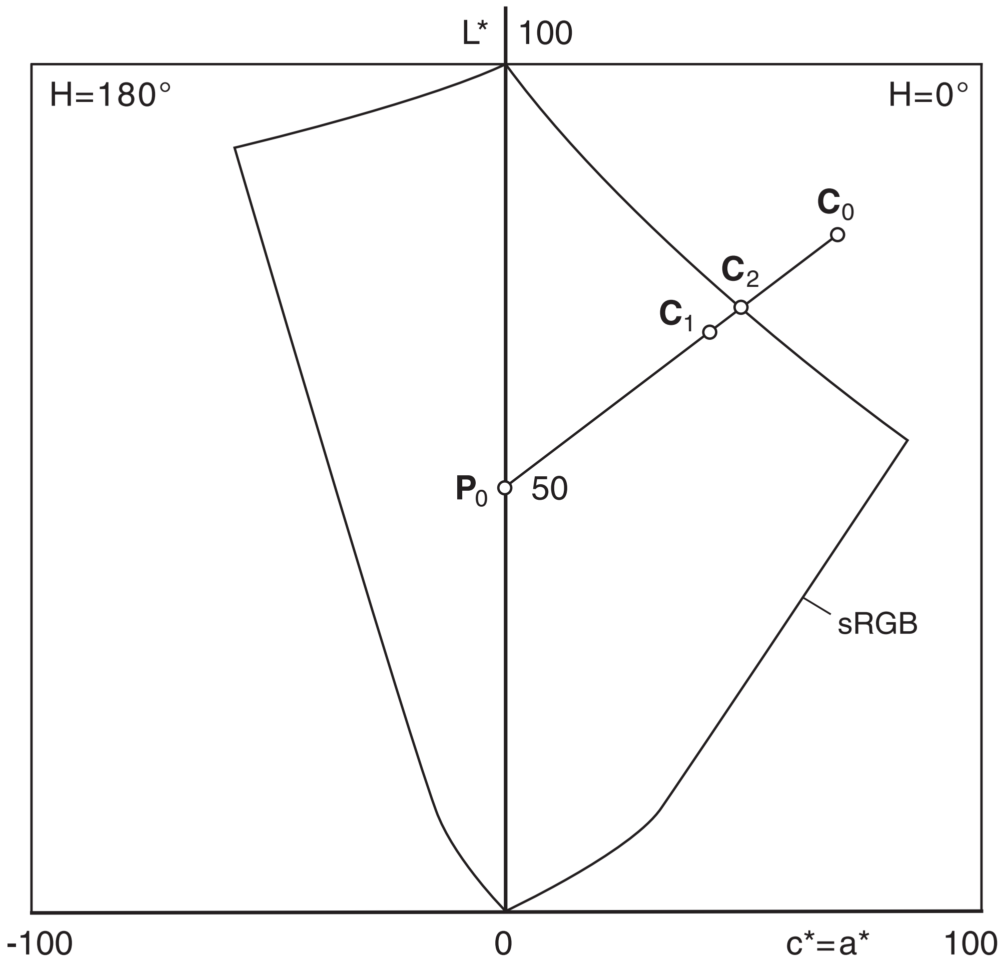

Figure 3.2 Out-of-gamut distance for sRGB

4.1 Mapping CMYK to CIELab / Concept

ICC profiles for CMYK printers are built by printing and measuring a target. Target values are defined by CMYK numbers. The well-known target IT8.7/3 for the spectrophotometer Eye-One Pro contains 1120 patches, and ECI2002 contains 1780 patches. The target is printed without any color management by sending the CMYK numbers to the printer. The CIELab values of the printed patches are measured by the respective instrument.
Based on these data, a signal flow from CMYK to CIELab is established, as shown in Fig.4.1. This can be found in an ICC profile as AtoB1 tag. ‚AtoB‘ means output (CMYK) to Profile Connection Space (CIELab). ‚1‘ means rendering intent Media-Relative Colorimetric, or simply RelCol. It is a rather bad habit to call such a tag A2B1, where ‚2‘ is expected to be interpreted as ‚to‘.

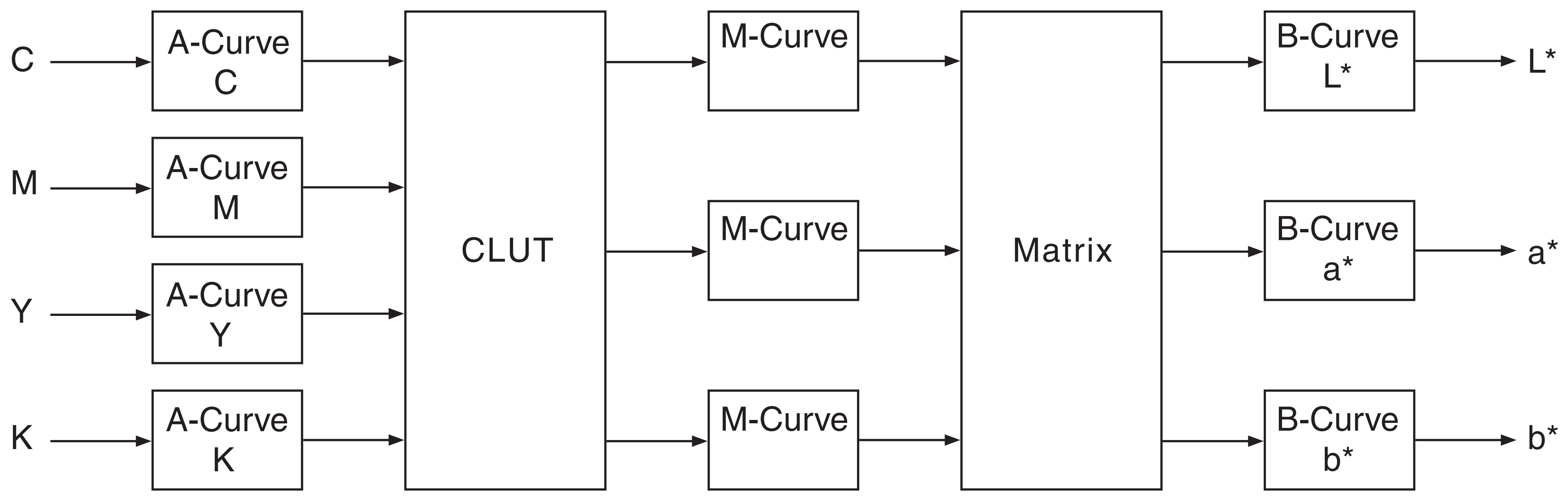

Figure 4.1 ICC signal flow for AtoB1

Each channel CMYK is input for a one-dimensional A-Curve. The input range is 0 to 1 and the output range as well. A multidimensional CLUT (color look-up table) converts each set of four input channels into a set of three output channels. These can be fed through three one-dimensional M-Curves, a Matrix with 12 elements (3 for an offset) and three one-dimensional B-Curves.
The author does not interpret the ICC profiles directly. He uses ICC Profile Inspector [16] and generates dumps of all tables and the header. Here we find the number of gridpoints. This is for instance n=9 or n=16.
For n=16 (the actual example) we have 16 values for each input channel, but the spacing is 1/15, normalized range 0 to 1 assumed. Altogether we have 16^4=65536 sets CMYK. The CLUT delivers the same number of output sets Lab for L*,a* and b*.
This huge number of sets is generated by interpolating the original 1120 or 1780 sets.
The content of the A-Curves can look like a slightly nonlinear function, but as long as everything is smooth and regular for the input and output range 0 to 1, this nonlinearity does not affect the boundary of the CIELab gamut volume. According to ICC Profile Inspector, the investigated profiles do not have M-Curves and M is an identity matrix (the offset terms are anyway not shown, probably zero). The B-Curve is always an identity function with input and output range 0 to 1. For the actual task it is sufficient to investigate the 65536 output sets of the CLUT.
The first set of the CLUT output for AtoB1 for the index 0 means always L*=100, a*=0 and b*=0. It does *not* represent paper white.
The respective integer values of the binary coded numbers are found for three different examples:

N1	N2	N3	N4			
Index	LutMax	LutZero	LutZero			
0	65535	32896	32896	ICC.1:2004-10	New standard	for version 4 profiles
0	65280	32768	32768	ICC.1:2004-10	Legacy mode	ISOCoated_v2-eci.icc
0	65535	32768	32768	ICC.1:2004-10	Not defined	ProfileMaker5 inkjet profile

The discrepancy is obvious. Therefore the CIELab values are extracted by this formula, using N2,N3,N4 and LutMax, LutZero for indices N1=0 to 65535:

$$L^* = 100 \cdot (N2)/LutMax$$
$$a^* = 128 \cdot (N3-LutZero)/LutZero$$
$$b^* = 128 \cdot (N4-LutZero)/LutZero$$

The application of a profile for printing requires an inverse signal flow, for instance BtoA1 for RelCol. Inputs are CIELab and outputs are CMYK. This is here not discussed.

4.2 Mapping CMYK to CIELab / White point mapping

For the rendering intent RelCol the CLUT white is mapped to the medium white, and all other colors similarly. The dump by ICC Profile Inspector delivers as well the medium white point (MediaWhitePointTag). This is always given in CIE XYZ. It is rather strange that the medium white point is slightly different for all ISOCoated profiles. For ISOCoated_v2_eci.icc we get:

X_{mw} = 0.84552
Y_{mw} = 0.87683
Z_{mw} = 0.74716

Furtheron one needs the Reference White D50 for CIELab:

X_n = 0.96420
Y_n = 1.00000
Z_n = 0.82491

The CIELab value, as extracted on the previous page, are mapped to medium white:

LABtoXYZ (L*,a*,b*,X_1,Y_1,Z_1)

$X_2 = X_1 \cdot X_{mw}/X_n$
$Y_2 = Y_1 \cdot Y_{mw}/Y_n$
$Z_2 = Z_1 \cdot Z_{mw}/Z_n$
XYZtoLAB (X_2,Y_2,Z_2,L*,a*,b*)

4.3 Mapping CMYK to CIELab / Extracting table values

For some applications it is necessary to extract from the CLUT not only the CIELab values but also the respective CMYK values simultaneously. This pseudocode explains the concept:

```
n   =16     % number of gridpoints
n1  =n-1
dc  =1/n1   % increment for normalized values CMYK
in  =0      % index for row in CLUT data file
For c=0 to n1 do
Begin
    C=c·dc
    For m=0 to n1 do
    Begin
        M=m·dc
        For y=0 to n1 do
        Begin
            Y=y·dc
            For k=0 to n1 do
            Begin
                K=k·dc
                % Read one row ‚in‘ of CLUT and extract L*,a*,b*
                % Apply white point mapping
                % Do anything with C,M,Y,K,L*,a*,b*
                in=in+1
            End
        End
    End
End
```

Multiplications like C=c·dc are replaced by incrementing the color values by dc (perhaps less understandable). It is possible to fill tables for all faces of the color cube CMY (not CMYK) by choosing in the innermost loop only values K=0. The faces are calculated for six cases (C=0; M,Y), (C=1; Y,M), (M=0; Y,C), (M=1; C,Y), (Y=0; C,M), (Y=1; M,C). For instance (C=1; Y,M) means: if C=1 then use Y as first parameter and M as second. This guarantees in further calculations that the normal vector points into outside direction.

4.4 Mapping CMYK to CIELab / Non-convex hull

The graphic shows the gamut boundary for sRGB for L*=70 (black contour) and CLUT values for the CMYK profile ISOCoated_v2-eci.icc in the range L*=69 to L*=71. The task in this tutorial example is now: find the CMYK gamut boundary for the sample points.The red line is the convex hull. If the black dots were nails, then the convex hull could be simulated by a rubber band. The convex hull is unique and it can be found mathematically without significant difficulties.

Unfortunately it does not describe what we want. We need a non-convex hull, as shown for the three-dimensional case by Figure 2.3. In other words: the hull can be weakly concave. These difficulties were extensively discussed by J.Morovič [8]. One practical solution is the method by ‚Alpha Shapes':

Roll a wheel with a sufficiently large radius along the contour of the ‚nails'. Mark the trajectory of the center and construct the hull by going orthogonally from the trajectory into inside direction by distance ‚radius'. For the three-dimensional case use a sphere instead of the wheel.

A powerful alternative, the ‚Segment Maxima Algorithm', was developed by J.Morovič, see next chapter.

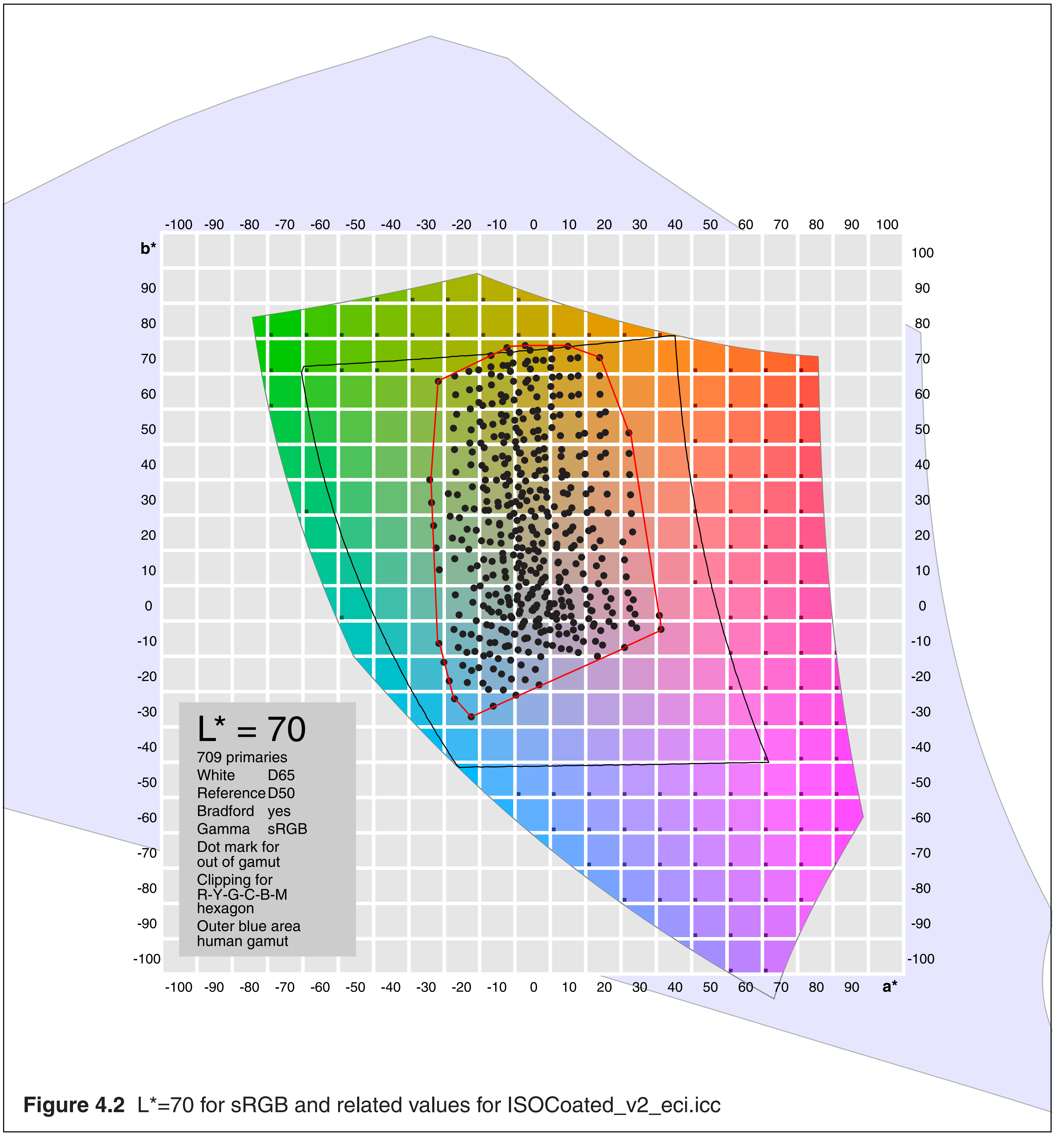

Figure 4.2 L*=70 for sRGB and related values for ISOCoated_v2_eci.icc

4.5 Mapping CMYK to CIELab / Segment maxima algorithm

Following the concept of J.Morovič [8], the gamut boundary will be described in sphere coordinates, which are centered in the point L*=50, a*=0, b*=0.
The graphic shows a subdivision by n=20 azimuth/longitude angles ψ_i and m=10 elevation/latitude angles θ_j. The actual numbers are somewhat arbitrary, for instance n=24 and m=18 for profiles with 16 gridpoints per channel.

There are three twodimensional tables *Psi*, *The* and *Rad* with entries for i=0 to n and j=0 to m. *Psi* contains the original azimuth angles ψ_i=0° to 360° with increments dψ=360°/n. *The* contains the original elevation angles θ_j=-90° to +90° with increments dθ=180°/m. *Rad* is initialized by r=10 CIELab units. The illustration shows r=50 (a hypothetical ideal gamut boundary).

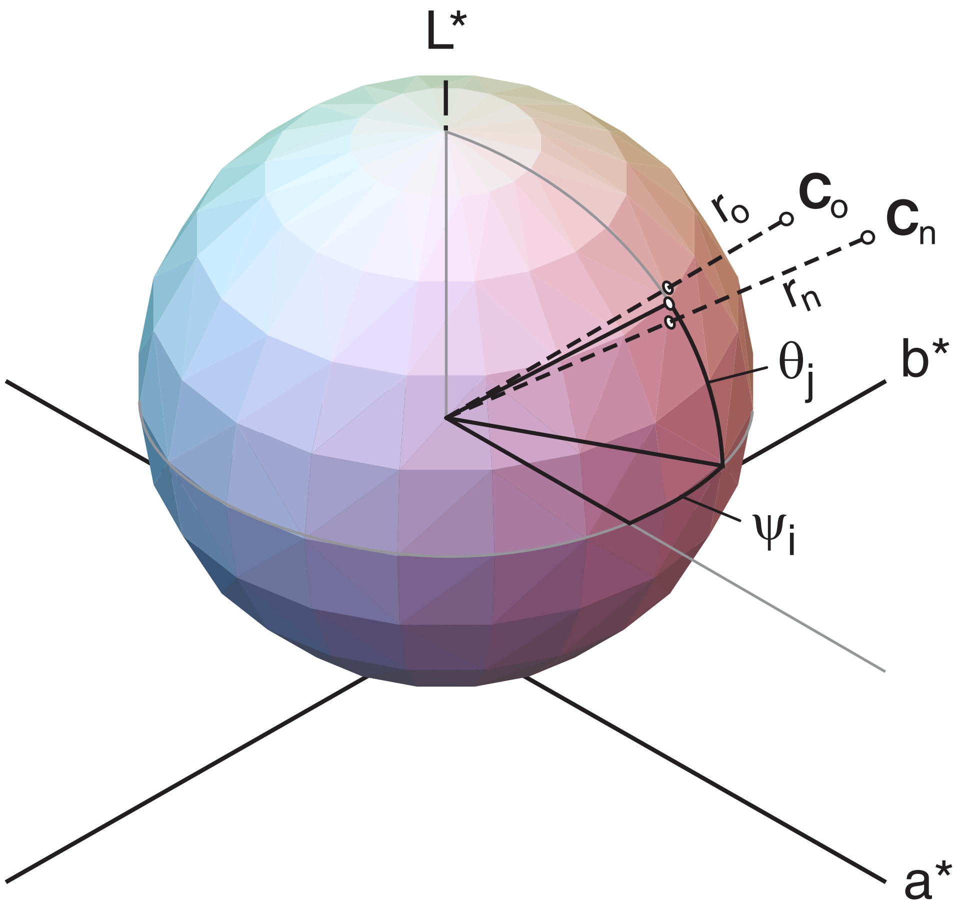

Figure 4.3 Sphere coordinates

For 16 gridpoints per channel we have a CLUT with 65536 sets L*,a*,b*. These values are scanned once and the white point transform is applied immediately (the white point is then at paper white).
For each new set $\mathbf{C}_n$ the actual angles ψ_n,θ_n and the actual radius r_n are calculated. From ψ_n,θ_n one can find the indices i and j. For instance ψ_n is in a strip $\psi_i \pm d\psi/2$. Old values $\mathbf{C}_o$ are available, either by initialization or from previous steps. If the new radius is larger than the old one, $r_n > r_o$, then the old angles and the radius are overwritten by new values. This results in a growing gamut volume. Finally, the tables contain the angles and the radius for the boundary.

The sphere coordinates can be converted into cartesian coordinates a*,b* and L*, but one needs mostly some tweaking in advance. At the poles the points on the sphere are rather dense. Even for 65536 data sets it happens that not enough large radii are found – the surface has holes or valleys. Near to the white point this problem was solved by closing the contour by pieces of planes through existing good points and the white point itself.
It turned out that such a correction was not necessary at the black point.
It is probably obvious that ψ=0° and ψ=360° mean the same azimuth. The introduction of 360° was necessary in order to find the actual indices for a color $\mathbf{C}_n$ by simple means (division of ψ by dψ and θ by dθ, followed by rounding) and for drawing closed surfaces. All radii for 0° and 360° are compared, and the entries with the greater one are finally used for both angles.

Unfortunately one cannot use arbitrarily large numbers n and m. If the angle grid is too dense then there cannot be found enough relevant maxima, the same situation as for the white point region.
Thus, one has to find appropriate numbers n and m, which depend loosely on the number of gridpoints per channel.

The method has a couple of advantages: sharp edges are preserved and the gamut boundary is valid not only for CMY but for CMYK. Drawing the gamut boundary for planes of constant hue or constant lightness is simple.

The 3D graphic can be improved by overdrawing the upper three faces by mapping only CMY (chapter 4.1 to 4.3). It will be shown that black ink K shifts the gamut always towards less L* and less chroma. Therefore the upper three faces are not affected by black ink K.

The problems at the poles could be avoided if the generic spherical subdivision would be replaced by icosahedron subdivision as explained in [22]. Such a subdivided icosahedron consists of almost equal triangles, whereas the generic subdivision results in very small quadrilaterals or triangles at the poles.

4.6 Mapping CMYK to CIELab / Contribution of black ink K

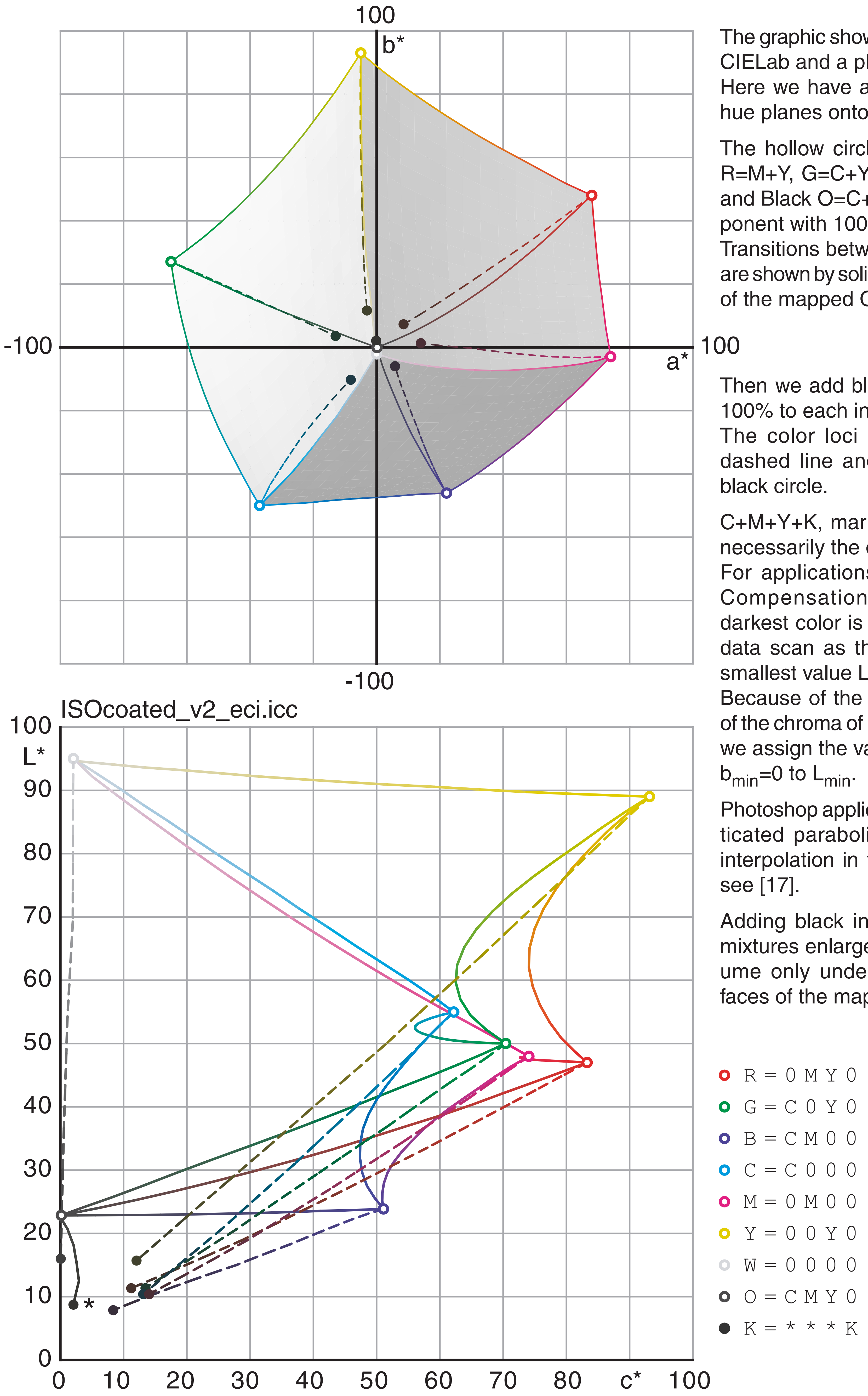

Figure 4.4 Contribution of ink K

The graphic shows a top view onto CIELab and a plane L*,c*. Here we have a projection of all hue planes onto one.

The hollow circles mean C,M,Y, R=M+Y, G=C+Y, B=C+M, White, and Black O=C+M+Y, each component with 100%. Transitions between these colors are shown by solid lines–the edges of the mapped CMY cube.

Then we add black ink K=0% to 100% to each ink or ink mixture. The color loci move along the dashed line and end in a filled black circle.

C+M+Y+K, marked by (*), is not necessarily the darkest color. For applications of Black Point Compensation BPC [17], the darkest color is found during the data scan as the color with the smallest value L_{min}=min(L*). Because of the uncertain nature of the chroma of rather dark colors we assign the values a_{min}=0 and b_{min}=0 to L_{min}.

Photoshop applies a more sophisticated parabolic least squares interpolation in the black region, see [17].

Adding black ink K to other ink mixtures enlarges the gamut volume only under the three dark faces of the mapped CMY cube.

4.7 Mapping CMYK to CIELab / Out-of-gamut distances

Figure 4.5 shows all azimuth or hue planes projected onto one. A color $\mathbf{C}_0$ is given in CIELab coordinates. It is assumed that the generating space reaches from darkness L*=0 to maximal lightness L*=100. This is true for RGB working spaces and for the Munsell system. Printing such a color with rendering intent Media-Relative Colorimetric (RelCol) requires a mapping of L*=100 to paper white. According to the ICC specifications there is no black point mapping for RelCol (but it is applied for rendering intent Perceptual). Therefore Adobe had introduced in Photoshop Black Point Compensation (BPC), which is highly recommended.
The medium white point is given by the MediaWhitePointTag in CIE XYZ: X_{mw},Y_{mw},Z_{mw}. The darkest black is here calculated by finding L_{min} in the CLUT (approximately) and setting a_{min}=0, b_{min}=0. The index ‚pb' means profile black point, which contains the information about the darkest printable black, but with assumed chroma zero (see point * in Figure 4.4). The index ‚n' means reference white for CIELab, which is D50.

LABtoXYZ (L_{min},0,0,X_{pb},Y_{pb},Z_{pb})
LABtoXYZ (L_0,a_0,b_0,X_0,Y_0,Z_0)
$X_1 = X_{pb} + (X_{mw} - X_{pb}) \cdot X_0/X_n$
$Y_1 = Y_{pb} + (Y_{mw} - Y_{pb}) \cdot Y_0/Y_n$
$Z_1 = Z_{pb} + (Z_{mw} - Z_{pb}) \cdot Z_0/Z_n$
XYZtoLAB (X_1,Y_1,Z_1,L_1,a_1,b_1)

The graphic shows symbolically that $\mathbf{C}_0$ (values L_0,a_0,b_0) is shifted to $\mathbf{C}_1$ (values L_1,a_1,b_1). The azimuth and elevation angles for $\mathbf{C}_1$, divided by $d\psi$ and $d\theta$, followed by rounding, deliver the indices i and j for the the color $\mathbf{C}_2$, mainly the table value r_2 in *Rad*. If the radius r_1 is larger than the table value r_2, then the color is out-of-gamut. Then the vector $\mathbf{C}_2\mathbf{C}_1$ is projected onto the line $\mathbf{P}_0\mathbf{C}_1$, and the length of $\mathbf{C}_3\mathbf{C}_1$ in CIELab units is considered as out-of-gamut distance dE.

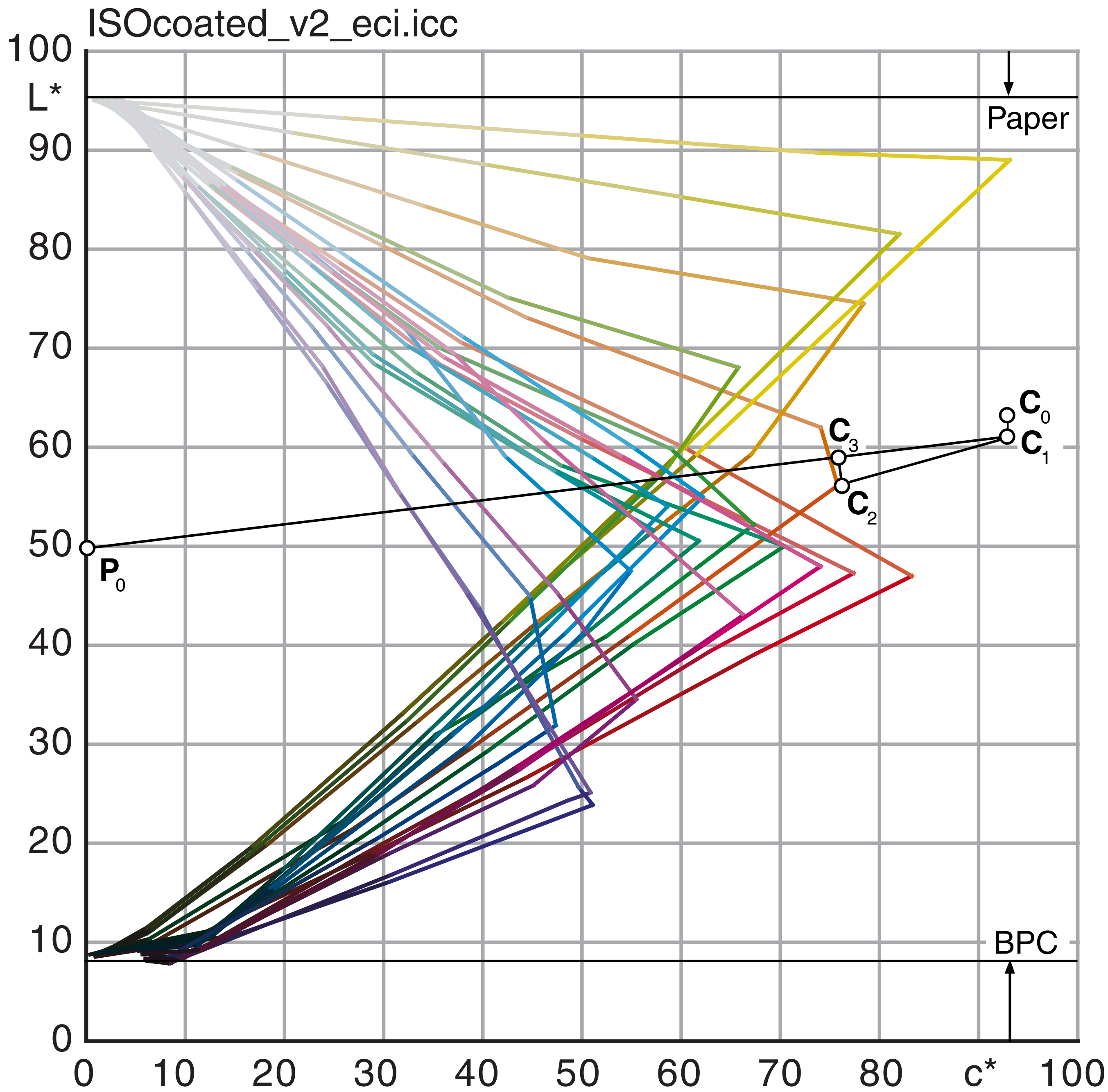

Figure 4.5 Out-of-gamut distance for CMYK

5.1 Examples / CIELab 3D: aRGB and ISOCoated_v2_eci.icc

Figure 5.1 shows that the CMYK gamut for ISOCoated is inside the gamut for aRGB.

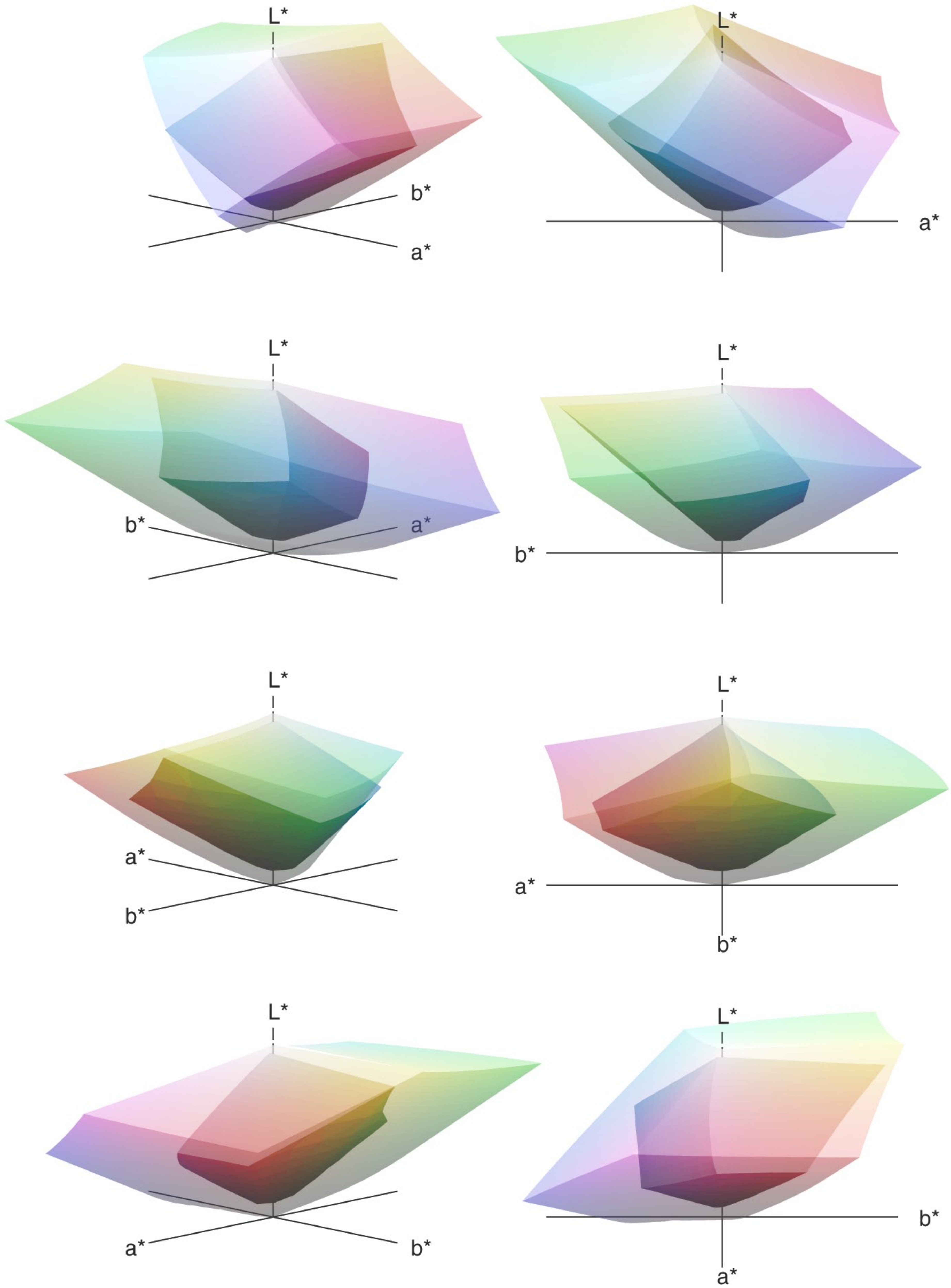

Figure 5.1 Gamuts for aRGB and ISOCoated_v2-eci.icc

5.2 Examples / CIELab 3D: ISOCoated_v2_eci.icc / Contrib. of black ink K

Figure 5.2 shows the important contribution of black ink K to the gamut volume (gray regions).

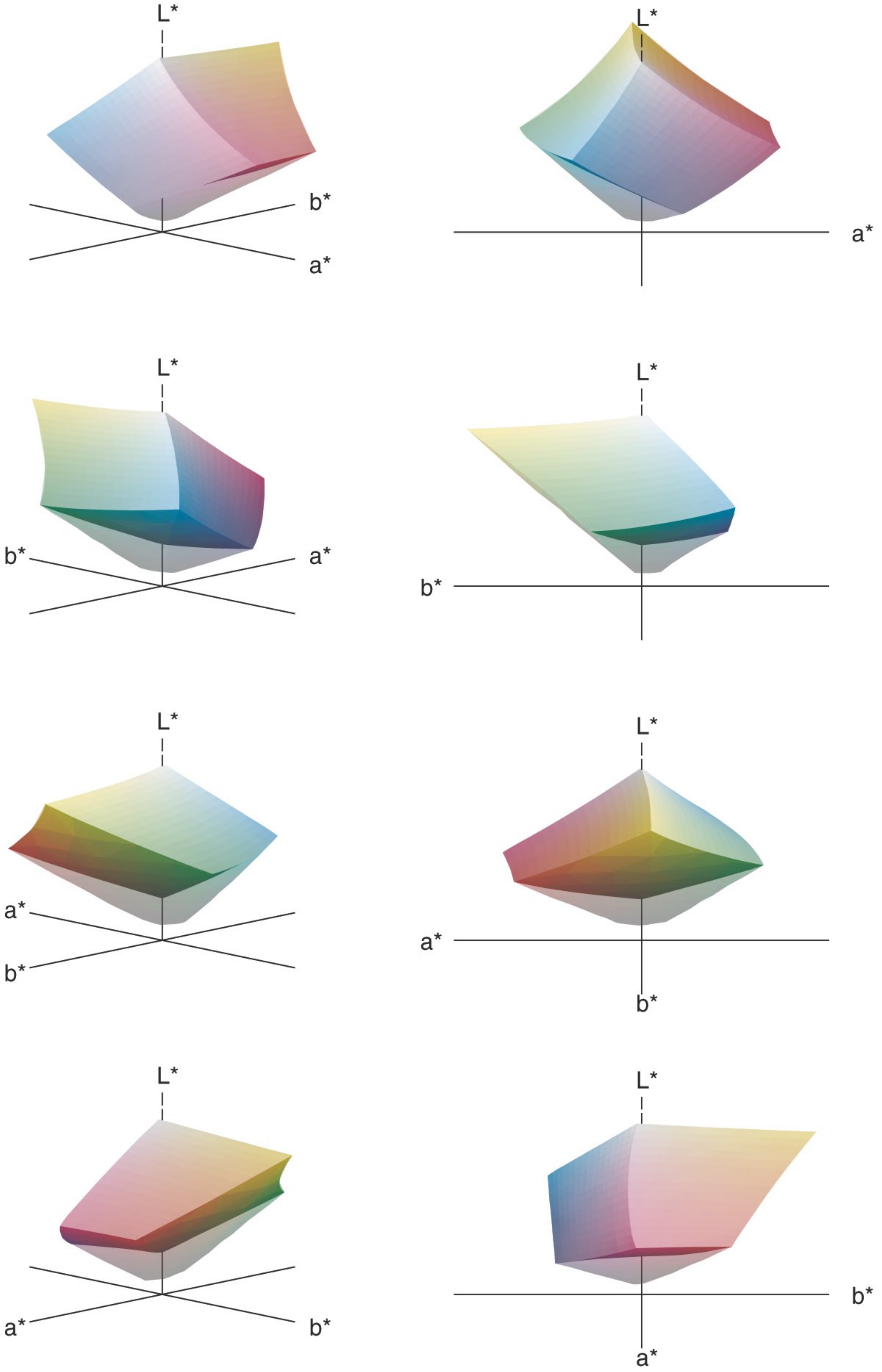

Figure 5.2 Gamut for ISOCoated_v2-eci.icc without and with black ink K

5.3 Examples / CIELab 3D: aRGB and Inkjet Mutoh 6100 (190S)

Figure 5.2 3 shows that the Inkjet gamut for Mutoh 6100 is generally smaller than the gamut for aRGB, nevertheless partly outside.

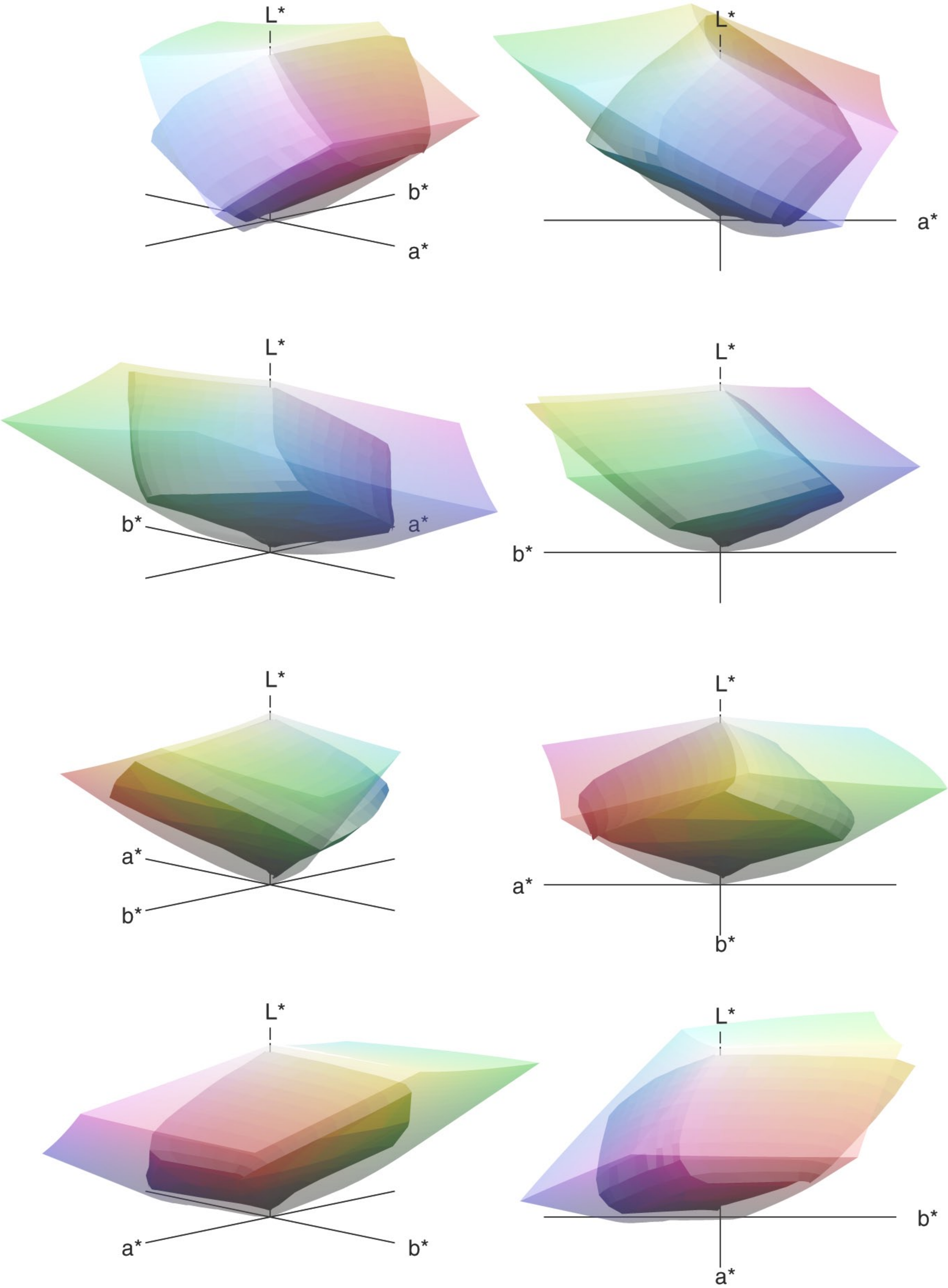

Figure 5.3 Gamuts for aRGB and Inkjet Mutoh 6100 (190S)

5.4 Examples / CIELab 3D: Inkjet Mutoh 6100 (190S) / Contrib. of black ink K

Figure 5.4 shows the rather small contribution of black ink K to the inkjet gamut volume (gray regions).

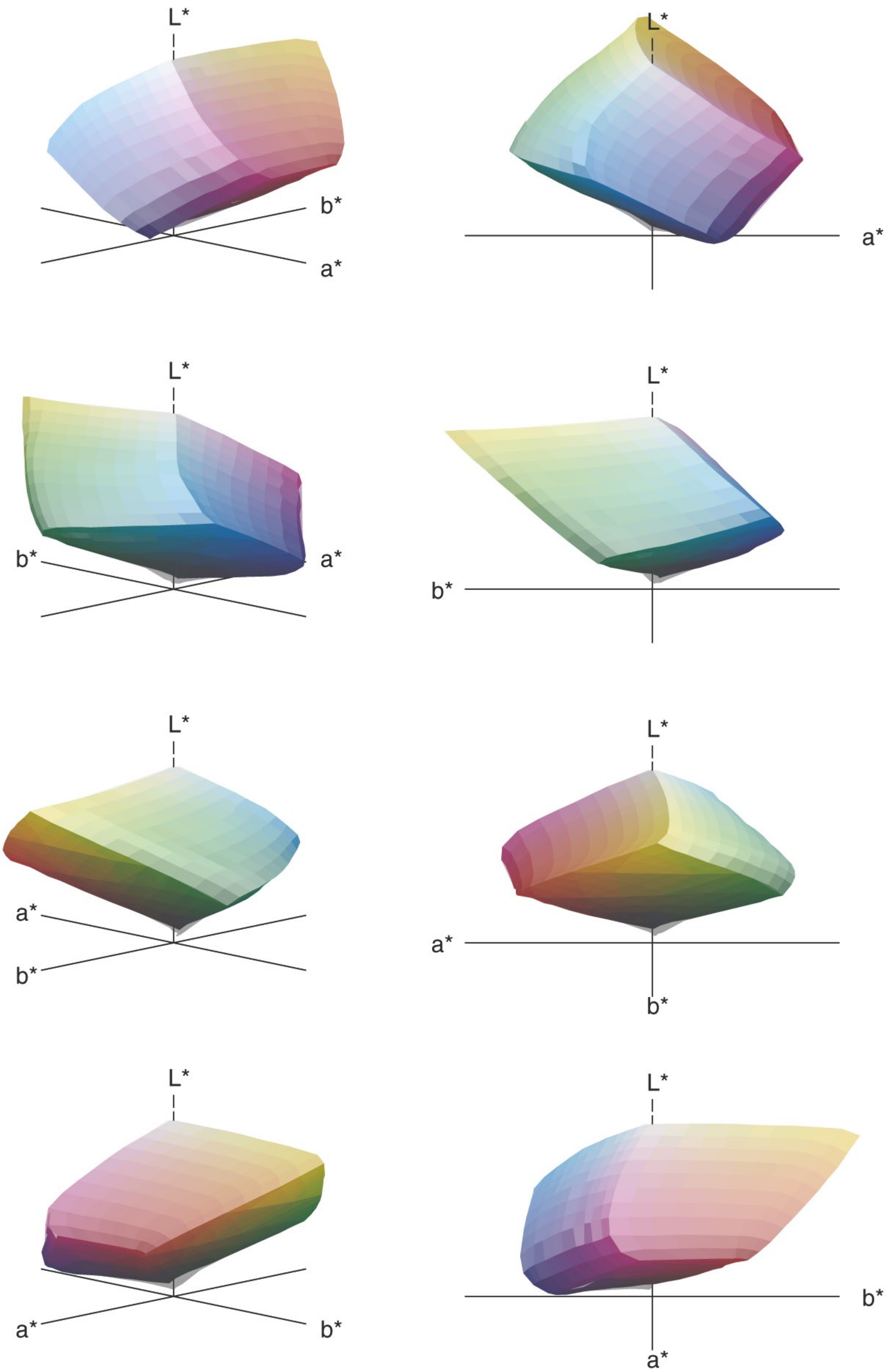

Figure 5.4 Gamuts for Inkjet Mutoh 6100 (190S) without and with black ink K

5.5 Examples / CIELab 2D: sRGB and ISOCoated_v2_eci.icc

Figure 5.5 shows that sRGB is not sufficient for offset printing by ISOCoated. The lightness values are different because of white point and black point mapping for RelCol.

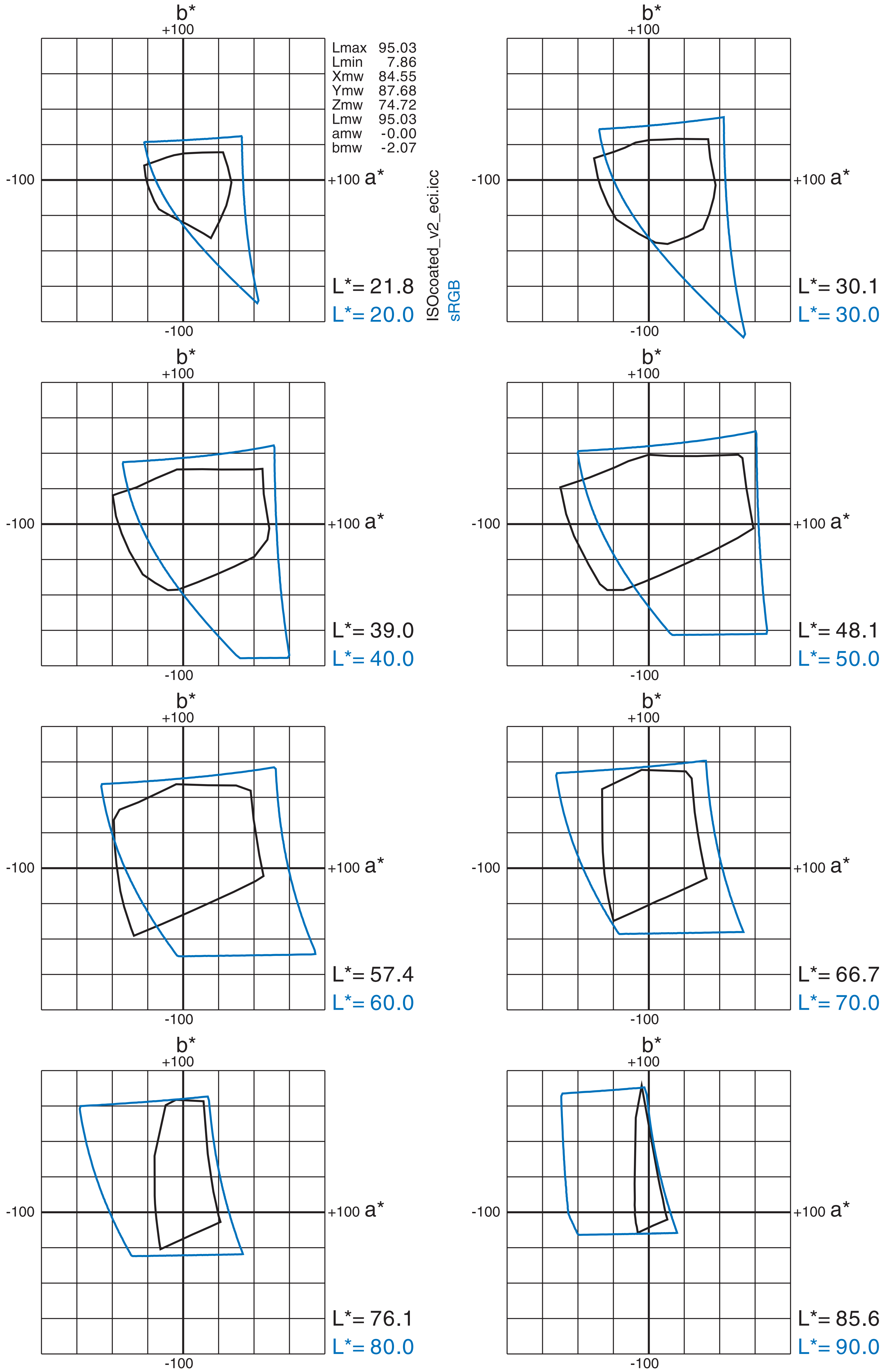

Figure 5.5 Gamuts for sRGB and ISOCoated_v2_eci.icc

5.6 Examples / CIELab 2D: aRGB and ISOCoated_v2_eci.icc

Figure 5.6 shows that aRGB is sufficient for offset printing by ISOCoated. The lightness values are different because of white point and black point mapping for RelCol.

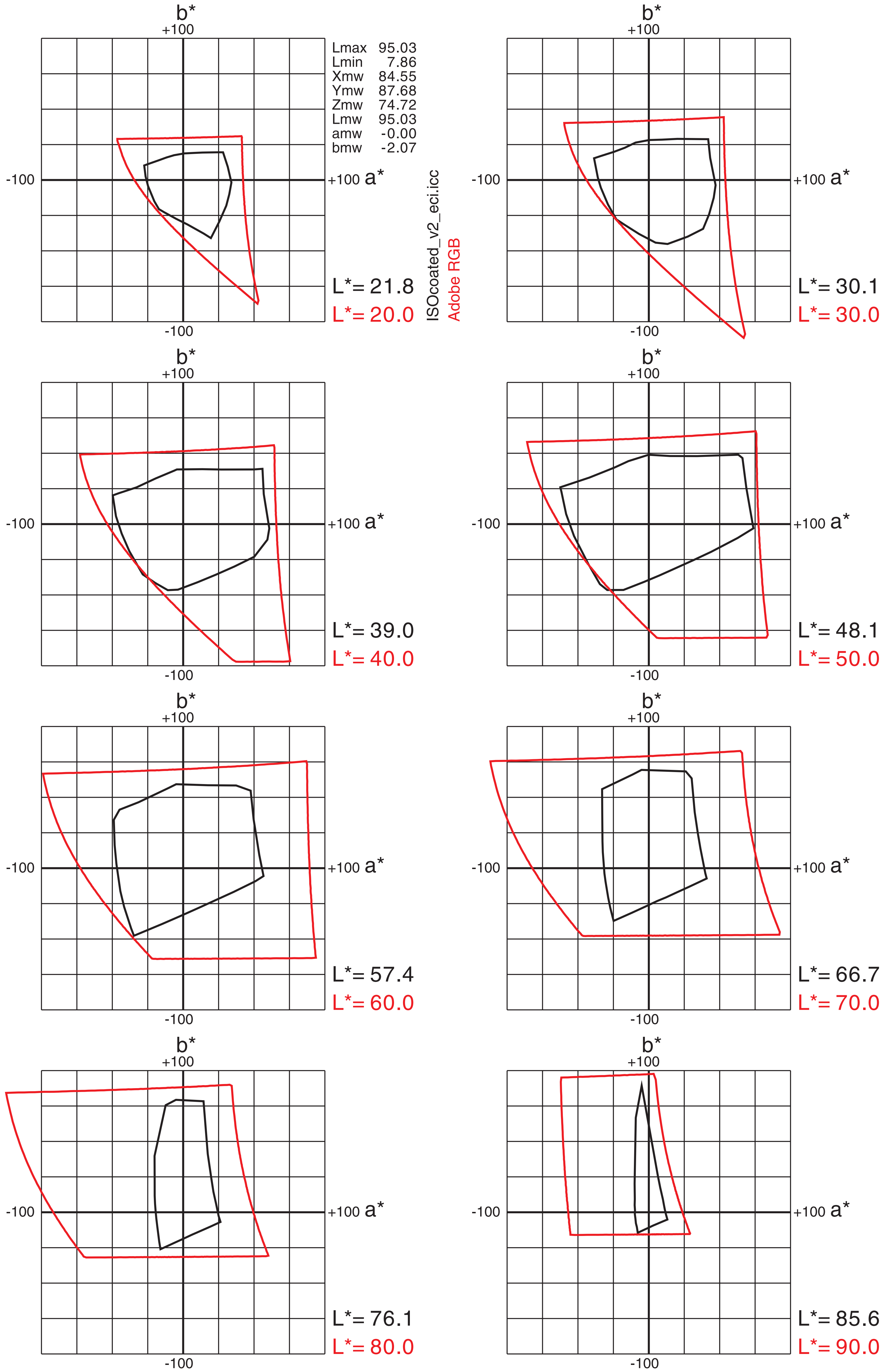

Figure 5.6 Gamuts for aRGB and ISOCoated_v2_eci.icc

5.7 Examples / CIELab 2D: aRGB, ISOCoated_v2..., Inkjet Mutoh 6100 (190S)

Figure 5.5 shows that the inkjet gamut is sufficiently large for proof printing ISOCoated. The lightness values are different because of white point and black point mapping for RelCol.

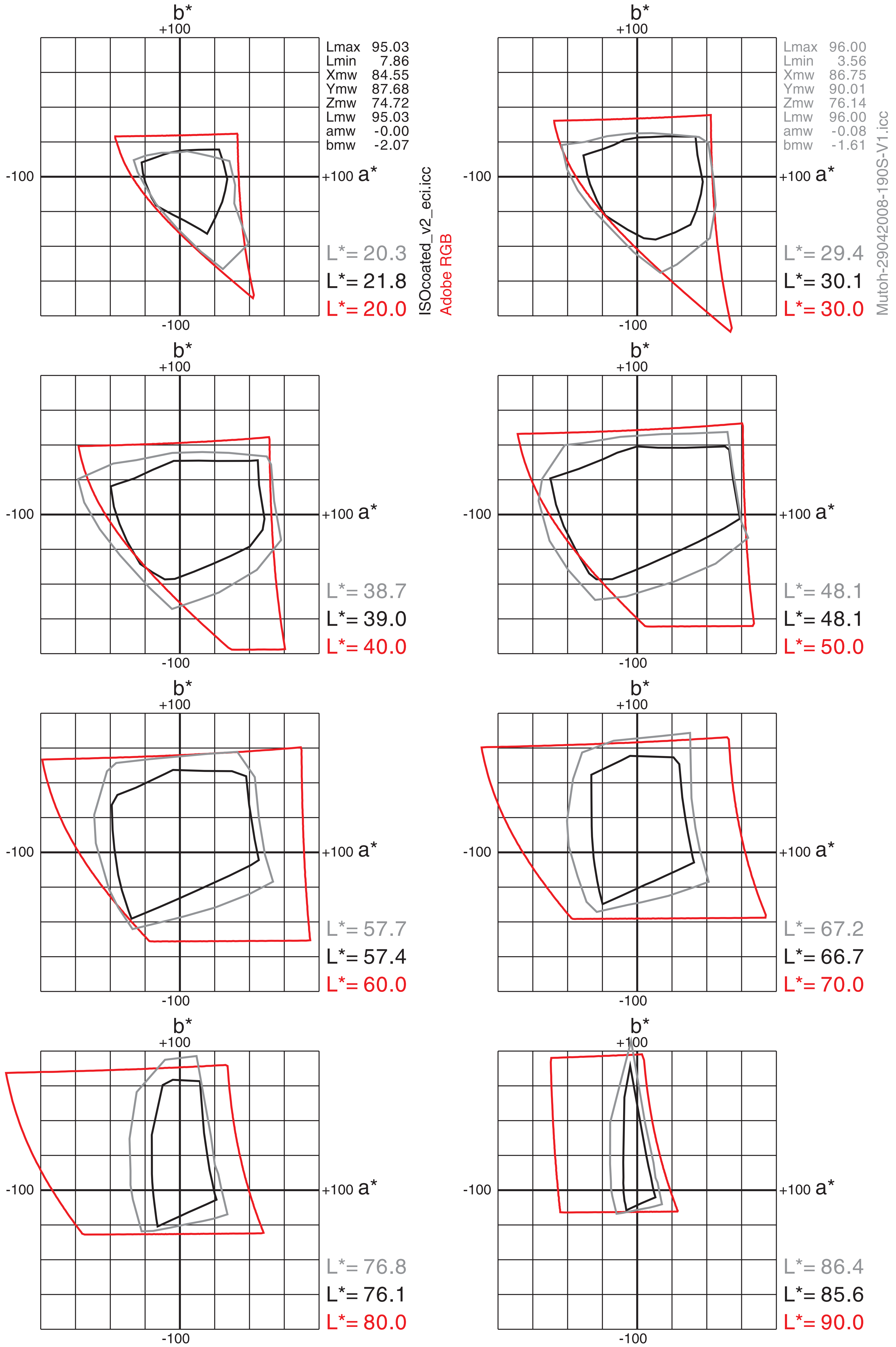

Figure 5.7 Gamuts for aRGB, ISOCoated_v2_eci.icc, Inkjet Mutoh 6100

6.1 References

[1] G.Wyszecki + W.S.Stiles
Color Science
John Wiley & Sons, New York ,..., 1982

[2] R.W.G.Hunt
Measuring Colour
Fountain Press, England, 1998

[3] R.W.G.Hunt
The Reproduction of Colour, Sixth edition
John Wiley & Sons, Chichester, 2004

[4] E.J.Giorgianni + Th.E.Madden
Digital Color Management
Addison-Wesley, Reading Massachusetts ,..., 1998

[5] R.S.Berns
Billmeyer and Saltzman's Principles of Color Technology, Third edition
John Wiley & Sons, New York ,..., 2000

[6] M.D.Fairchild
Color Appearance Models, Second Edition
John Wiley & Sons, Chichester, 2005

[7] H.R.Kang
Computational Color Technology
SPIE Press, Bellingham, 2006

[8] J.Morovič
Color Gamut Mapping
John Wiley & Sons, Ltd, Chichester, 2008

[9] J.D.Foley + A.van Dam + St.K.Feiner + J.F.Hughes
Computer Graphics
Addison-Wesley, Reading Massachusetts,...,1993

[10] C.H.Chen + L.F.Pau + P.S.P.Wang
Handbook of Pattern recognition and Computer Vision
World Scientific, Singapore, ..., 1995

[11] M.Stokes + M.Anderson + S.Chandrasekar + R.Motta
A Standard Default Color Space for the Internet - sRGB
http://www.w3.org/graphics/color/sRGB.html
1996

[12] M.Nielsen + M.Stokes
The Creation of the sRGB ICC Profile
Year unknown, after 1998
http://www.srgb.com/c55.pdf (dead link)
Search by title or use this link
http://tinyurl.com/d7kwym

[13] International Color Consortium
http://www.color.org

[14] Specification ICC.1:2004-10
http://www.color.org/icc_specs2.xalter

[15] European Color Initiative: Offset profiles
http://www.eci.org/doku.php?id=en:start

6.2 References

[16] ICC Profile Inspector
http://www.color.org/profileinspector.xalter

[17] Adobe System‘s Implementation of Black Point Compensation
http://www.color.org/AdobeBPC.pdf

[18] Munsell Color Science Laboratory
http://www.cis.rit.edu/mcsl/online/munsell.php

[19] Adobe Systems: The Munsell Color System
http://web.archive.org/web/20030813092028/www.adobe.com/support/techguides/color/colormodels/munsell.html
Use short URL:
http://tinyurl.com/dg9ixx

[20] G.Hoffmann
CIE (1931) Color Space
http://docs-hoffmann.de/ciexyz29082000.pdf

[21] G.Hoffmann
CIELab Color Space
http://docs-hoffmann.de/cielab03022003.pdf

[22] G.Hoffmann
Sphere Tessellation by Icosahedron Subdivision
http://docs-hoffmann.de/ikos27042002.pdf

[23] G.Hoffmann
Color Management by ICC Profiles
http://docs-hoffmann.de/cmsicc08102003.pdf

[24] G.Hoffmann
The Digital Munsell
http://docs-hoffmann.de/munsell15052009.pdf

[25] G.Hoffmann
PostScript Tutor
http://docs-hoffmann.de/pstutor22112002.pdf

[26] G.Hoffmann
Spot Swatch Book
http://docs-hoffmann.de/swatch16032005.pdf

[27] Everything about Color and Computers
http://www.efg2.com

[28] G.Hoffmann
Color Management signal flow diagrams / CMS flow charts
http://docs-hoffmann.de/cmsflow03072013.pdf

This doc:
http://docs-hoffmann.de/gamshow15052009.pdf
Last revision October 07 / 2018

Gernot Hoffmann

February 01 / 2013 – October 07 / 2018
Website
Load browser / Click here