

# 注：

---

这是对**Foundation of 3D Computer Graphics**第19章的翻译，本章讲解了CG中很关键的色彩概念。本书内容仍在不断的学习中，因此本文内容会不断的改进。若有任何建议，请不吝赐教[ninetymiles@icloud.com](mailto:ninetymiles@icloud.com)

## Color

---

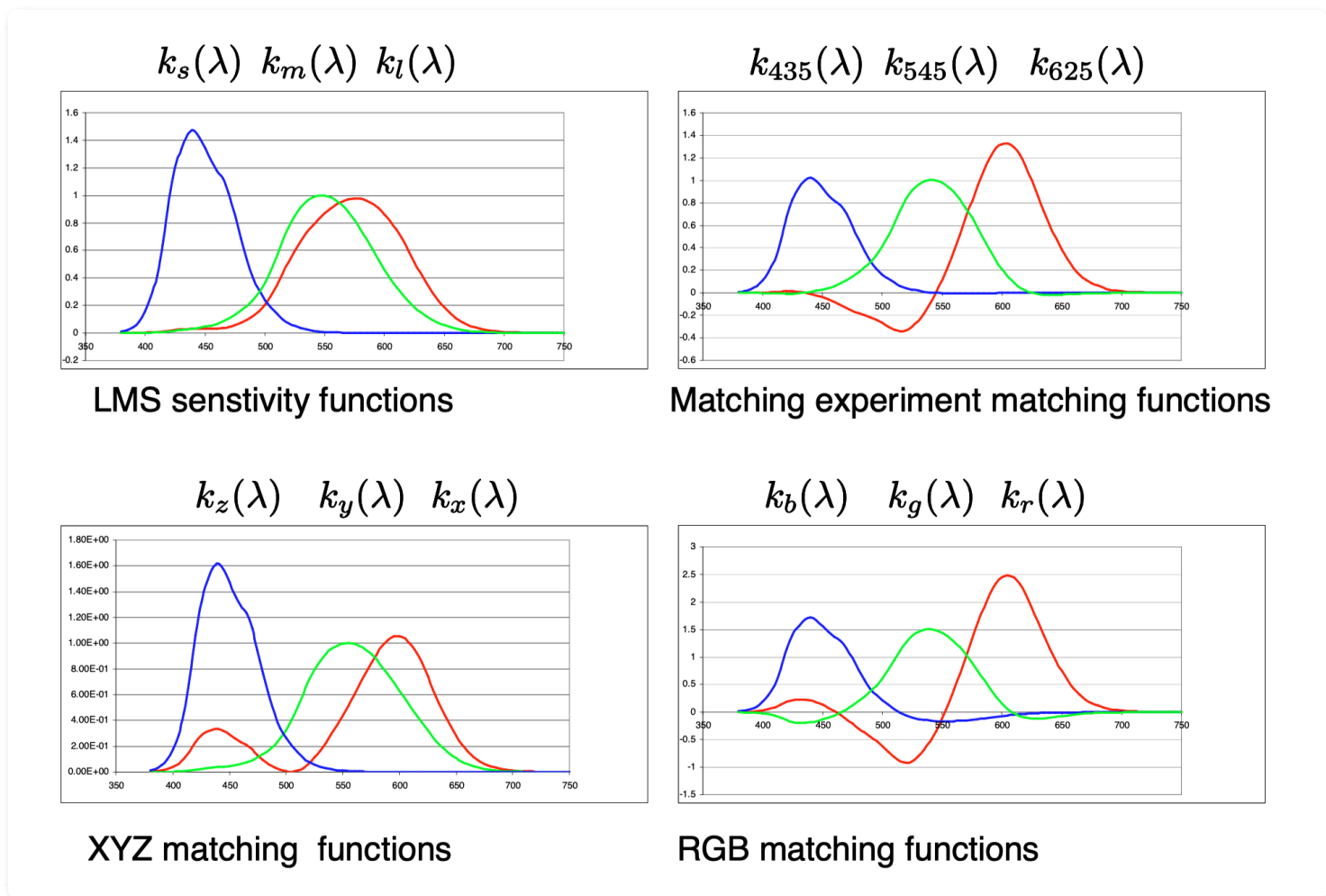
本节，我们将探索色彩的基本概念。同时还将讨论色彩是什么，以及多种表达色彩的方式。色彩在研究领域有着丰富的主题，并且在人类色彩感知领域尚有很多未解之谜。我们会在这个主题上专门投入时间，因为我们发现这个主题非常有趣，同时还因为其重要性不仅涉及到计算机成像方面同时还关联到电子（数码）设备成像领域。

事实上，色彩是一个被重载过的术语，色彩意味着很多东西。当光束照射在视网膜上，存在某种由锥体细胞（cone-cells）所感受的最初的神经反射（反应），这种反应独立地出现在每个锥体细胞上。我们称呼这种反应为视网膜色彩（retinal color）。视网膜色彩随后被以一种整体风格在整个视觉域上被处理，而正是这种处理导致了我们可以实际感受到并且在其之上进行判断的（被）感知色彩（perceived color）。感知色彩经常又被关联到我们正在观察的物体（object）上，这又被称为物体色彩（object color）。

在所有这些色彩阶段，对于两种实际的色彩我们可以判别的最简单的内容就是他们相同还是不同。而这种判别正是我们经常记录和量化的东西，且是我们在本章处理色彩的主要方式。然而在感知色彩级别，很明显存在色彩的意识体验，这点要以实验和正规方式来处理就要难得多。

最后，在我们（人类）如何将色彩感知组织为命名色彩方面（比如像红色和绿色这样的词汇）有更多待解问题。

在本章中，我们将主要聚焦在视网膜色彩（正常会丢掉视网膜这个（前缀）术语）。视网膜色彩相对容易被理解，为理解其它色彩概念的第一步。现在，我们首先将从被良好证实的生理（生物物理学）基础上描述视网膜色彩。之后，我们将直接从感知实验中重新推导出相同的色彩模型。我们还将讨论计算机图形中的一些最常见的色彩表达。



**Figure 19.1:** 感光/匹配 (sensitivity/matching) 函数。

## 19.1 简单生理模型 (Bio-Physical Model)

可见光是波长大致落入  $380 < \lambda < 770$  区间的电磁辐射，以纳米计量。（可以将每种波长当作光的一种不同的“物理口味”）。我们将讨论两种光束。纯光束  $l_\lambda$  是拥有一个“单位”的指定波长为  $\lambda$  的光（以辐照度单位来计量）。混合光束  $l(\lambda)$  拥有不同数量的多种波长。这些数量通过函数  $l(\cdot) : R \rightarrow R+$  来确定,同时以频谱辐照度单位来计量。光的数值总是非负，这是因为不存在“负光”。

人眼在视网膜上有多种光敏感细胞。锥体细胞给了我们色彩的感知。（非色盲）人类拥有3种不同的锥体细胞，我们称它们为短波锥体（short），中波锥体（medium）和长波锥体（long）（依据它们最敏感的光的波长命名）。为这三类锥体细胞关联3种感光函数  $k_s(\lambda)$ ,  $k_m(\lambda)$ ,  $k_l(\lambda)$ 。响应函数描述了一种锥体“响应”不同波长的纯光束有多强烈。例如， $k_s(\lambda)$  告诉我们短波敏感锥体在多大程度上响应纯光束  $l_\lambda$ 。（参看图示Figure 19.1左上角）。

因为每种纯光束导致视网膜上的3种响应值，针对每一种类型的锥体，我们可以可视化这种响应（值）为一个3D空间中的点。我们来定义一个3D线性空间，使用坐标  $[S, M, L]^t$ 。那么针对一个固定波长,我们可以绘制视网膜反应为一个拥有坐标  $[k_s(\lambda), k_m(\lambda), k_l(\lambda)]^t$  的矢量。当我们让  $\lambda$  变化，这种矢量会描摹出一个拉索曲线(lasso curve)（参看图示Figure 19.7的最顶部行）。拉索曲线（lasso curve）被  $\lambda$  参数化。拉索曲线（lasso curve）完全位于正（值）象限，因为所有的响应

值是正的。曲线开始和结束都在原点，因为这些极端的波长位于可视区域的边缘，超越边缘后锥体细胞的响应值就是0。曲线短暂停留于S轴（以淡蓝色显示）并且最终向L轴靠近（以红色表示）。曲线绝不会靠近M轴，因为没有光（包括混合光束）可以独立刺激这种锥体（细胞）。

在这种简单模型中，我们把由光束所产生的 $[S, M, L]^t$ 坐标认为是在描述由其所引起的（视网膜）色彩感光反应。我们使用符号 $\vec{c}$ 去表示一个色彩自身，现在，对于色彩我们将其和一个视网膜事件对等。马上我们也将严格的定义色彩。因此，在图示**Figure 19.7**，我们能够把每个3D矢量潜在地当作是表示某种色彩。那么在拉索曲线(lasso curve)上的矢量就是各种纯光束的实际色彩。

在某种强度范围内，（相关对应的）锥体（细胞）线性地响应照摄在它们之上的光。如此，对于混合光束 $l(\lambda)$ ，3个响应坐标值 $[S, M, L]^t$ 可用如下公式计算：

$$S = \int_{\Omega} d\lambda l(\lambda) k_s(\lambda) \quad (19.1)$$

$$M = \int_{\Omega} d\lambda l(\lambda) k_m(\lambda) \quad (19.2)$$

$$L = \int_{\Omega} d\lambda l(\lambda) k_l(\lambda) \quad (19.3)$$

此处 $\Omega = [380..770]$ 。

当我们考察所有可能的混合光束 $l(\lambda)$ ，它们形成的 $[S, M, L]^t$ 色彩坐标形成了（搜寻出了）3D空间中的某种矢量集合。因为 $l$ 可能为任意正值函数，搜索的集合由拉索曲线（lasso cuve）上的一些矢量的所有正值线性组合构成。因此，搜索出的集合是拉索曲线（lasso cuve）上的凸状锥体，我们称它为色彩锥体（color cone）。锥体内的矢量表示了实际可获取的色彩感知。锥体外的矢量，诸如垂直轴（M轴）不能成呈现为任何实际光束的感知，无论是纯光束还是混合光束。

为了帮助可视化色彩锥体（color cone），我们已将图示**Figure 19.7**的绘制分开（展开？）为一系列步骤。在第二行，我们已标准化拉索曲线（lasso cuve）- 标准化是通过缩放每个矢量以便满足 $S+M+L=K$ （ $K$ 为某种常量）。这样一个被缩放过的拉索曲线（lasso cuve）被称为马靴曲线（horseshoe curve）。我们还添加尾状线（射线）连接这个马靴曲线到原点。第三行中，我们增加了从原点到马靴曲线（horseshoe curve）的线条。目的是为了尝试给出针对色彩锥体（color cone）形状的更佳感觉。最后，在第四行中，我们放置一个不透明平面，用以展示色彩锥体的切片。在这个平面上，我们还绘制了位于这个切片上的真实色彩，这些色彩由R, G, B（红、绿、蓝监视器元素）的线性组合产生。（这种RGB色彩空间随后会讨论）。要绘制出最亮的色彩，受制于这种要求的限制，我们特意选择了在 $S+M+L=K$ 中的 $K$ 值，以便色彩切片包含由RGB坐标 $[1, 0, 0]^t$ 表示的色彩。在线框图（wireframe）中，我们展示了RGB-立方体（RGB-cube），它由通过红绿蓝的组合（系数在 $[0..1]$ 范围）获得的色彩集合组成。

拉索曲线（lasso cuve）由无数的矢量（纯光束响应值）构成，当然会超过3个！因此，对于严格位于色彩锥体之内的矢量，借助拉索曲线（lasso cuve）上一些矢量的正值线性组合，会存在生成某种固定 $[S, M, L]^t$ 色彩坐标的很多方式。每一个都等价于产生这种固定响应值的某种光束。如此，一定有很多真实可区分的光束，它们拥有不同数量的波长，产生相同的色彩感知。我们称任何两种这样的光束为条件等色。

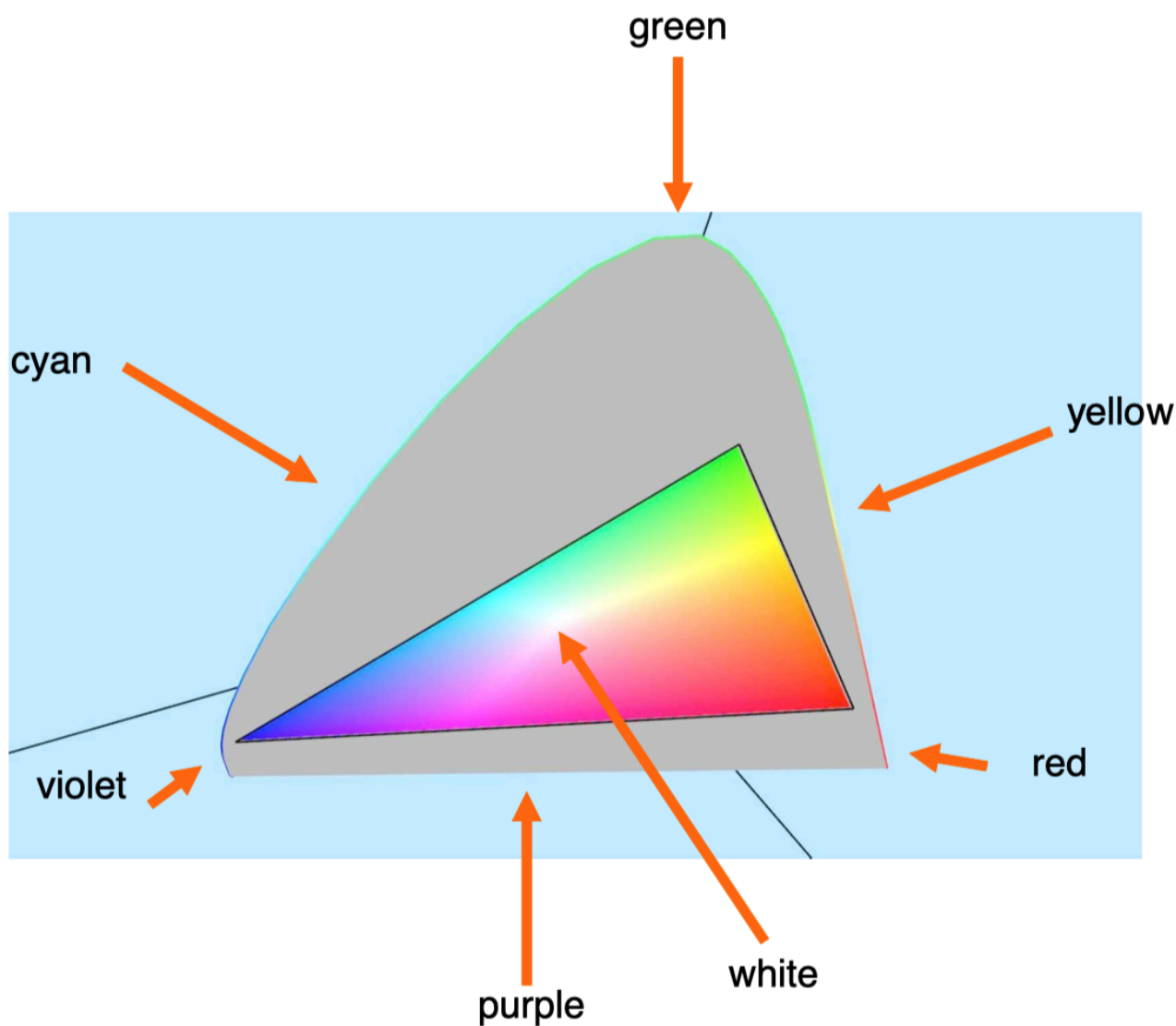
此处，我们总结已经看到过的一些有区别的数据类型，同时也有几个类型，我们随后将会见到。

- 一束纯光束被表示为 $l_\lambda$ 。一束混合光束被表示为 $l(\lambda)$ 。
- 光敏感（光感知）函数被表示为 $k(\lambda)$ 。我们随后也称之为匹配函数。
- 色彩的视网膜感知被表示为 $\tilde{c}$ 。之后，我们会使用三个这样的色彩去构成一个色彩空间的基(basis)。
- 色彩被表示为3个坐标值，诸如 $[S, M, L]^t$ 。一束光被看到的色彩坐标借助方程式（19.1）中的匹配函数来计算。
- 随后，我们会看到反射函数 $r(\lambda)$ ，它描述了出现在某种真实材料上的每个波长的反射部分。

### 19.1.1 色彩空间的映射（Map of Color Space）

目前，我们已经有足够信息来粗略映射出色彩锥体（color cone）（的形状）。

色彩锥体（color cone）中矢量（vectors）的缩放对应于我们可察觉到的色彩感知中的亮度变化，那么这种行为就不是很有趣。（虽然，在我们将一束橙色的光模糊后，我们实际会感知到棕色。）因而通过缩放（scale）来标准化色彩图是方便的，借助这种方式从而获得一个2D图示（2D drawing）（如图示**Figure 19.2**）。在本图示中，我们以图**Figure 19.7**中的色彩锥体的切片为初始起点。切片中所有被展示的色彩被缩放以便R，G或者B中的一个坐标值在完全的强度（处于最大值1）上。凭借显示器R、G和B显示元件（phosphor）的正值线性组合方式，我们无法展示出图示中被绘制的彩色三角形范围之外的色彩。我们就说这样的色彩位于显示器展示范围（gamut）之外。



**Figure 19.2:** 二维色彩图示。三角形范围外的色彩超越了计算机显示器的显示范围。

顺着色彩锥体（color cone）边缘的色彩是生动鲜明的（vivid）且被感知为“饱和的”。始于L轴，同时沿着曲线部分移动，我们顺着彩虹的色彩从红到绿再到紫罗兰色。这些色彩仅能通过纯光束被获得。另外，色彩锥体的边缘有一个平面的楔形（在2D图示中以一条线段表示）。位于这个楔形上的色彩是紫色和粉色系色彩。这些色彩不会显示在彩虹中，并且只能通过恰当地组合红色和紫罗兰（violet）色光束来获得。当我们围绕锥体边缘转圈，我们会经过不同的色调（hues）。

当我们从边缘向色彩锥体的中央区域移动进去，那些色彩，我们同时保持他们的色调，随着饱和度不断减小，色彩变浅并且最终发灰或发白。（但是在我们的处理中我们将不需要特意正式地选定一个色彩作为白色）。

这种通用描述可以被数字化方式准确表达在所谓的用以明确色彩的色调饱和度评估系统。

## 19.2 数学模型（Mathematical Model）

刚刚在19.1节描述的模型实际上在19世纪已经被推导出来，仅仅借助了几个感知实验。他们当时

无法使用研究眼睛细胞所需的技术。所以这是一个很神奇的伟业。此处，我们跟从这个原始的推理线索并且解释我们的色彩模型如何可以从头开始被推导出来，仅借助正确种类的感知实验。这会让我们更仔细深入的理解如何定义色彩，并且这也让我们可以使用所有线性代数的工具处理色彩空间，并无需借助任何种类的神经反应。

我们从来自物理学的基础知识作为起始点，也即是说光束可以被描述为波长分布 $l(\lambda)$ ，同时借助一种大略的观察-这种观察可以区分出光分布对于人类观察者有时候无法区分。为了仔细研究这种条件等色现象，同时特意去避免任何可能出现在人观察复杂场景时的效应，我们设计了一个实验设备，就如在图示Figure 19.3中所示。这允许我们为一个观察者呈现两束拥有已知波长分布的光束。我们随后就可以询问观察者是否这些光束看起来完全相同还是不同。

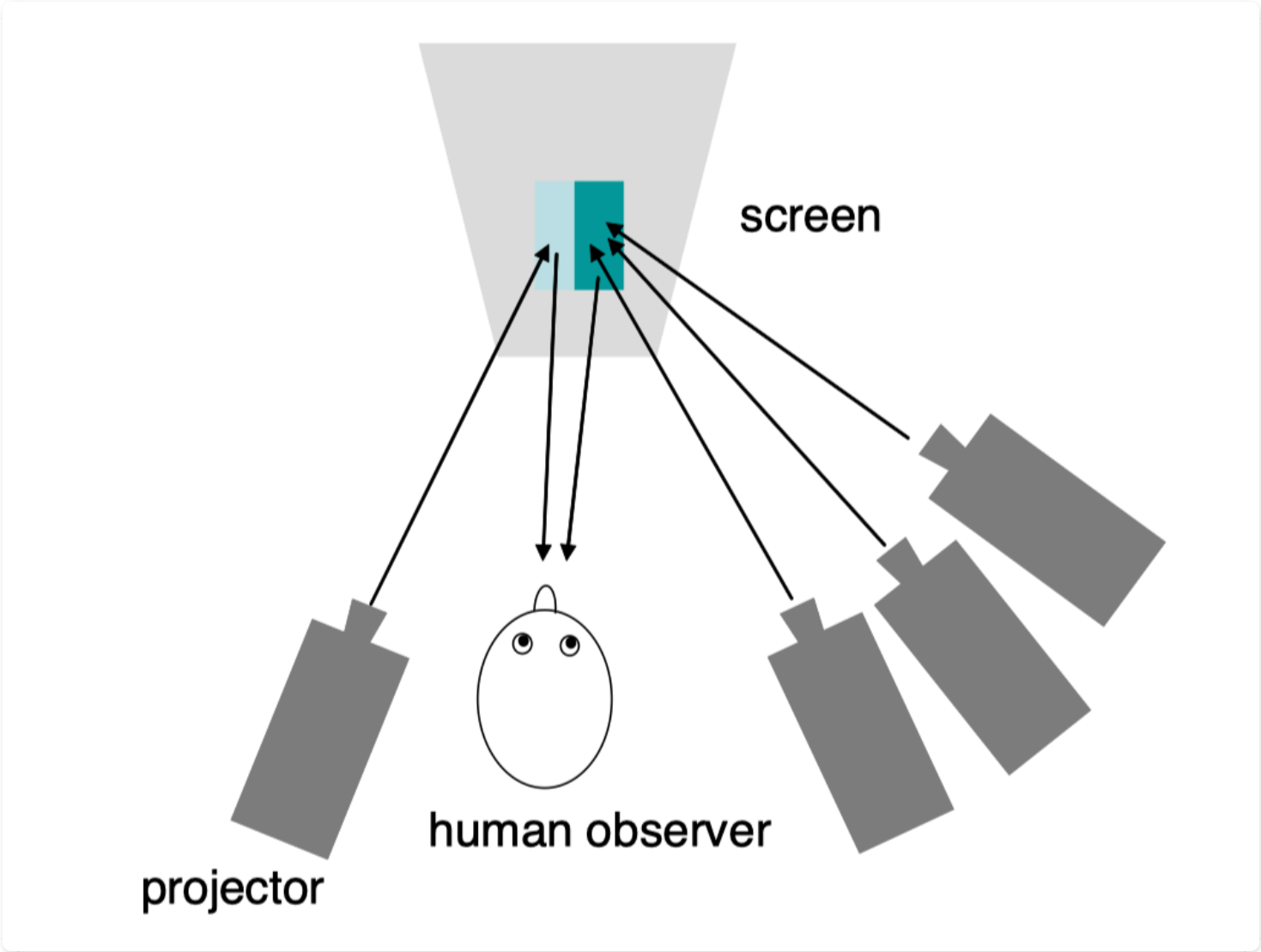


Figure 19.3: 基本色彩匹配实验装置。光投影仪使用多种可变光分布，将光束聚焦在一块大的单色屏幕上，借助特定方式形成两种色块，这种方式让每个光束都由受控制的波长分布构成。多个投影仪被放置在右边，以方便我们也能够检测到当多种光分布被汇总在一起会发生什么。人类观察者会被问两个色块能否通过色彩来区分或者不能。

就是在第一个实验中，我们验证了条件等色关系是可传递的（此处我们忽略刚刚可注意到的差异问题，以及阈值效应）。实际上，我们发现如果 $l_1(\lambda)$ 对于 $l'_1(\lambda)$ 不可区分，同时 $l'_1(\lambda)$ 对于 $l''_1(\lambda)$ 不可区分，那么 $l_1(\lambda)$ 也总是对于 $l''_1(\lambda)$ 不可区分。

由于这种传递性，我们实际上定义 $\tilde{c}(l_1(\lambda))$  - “光束 $l_1(\lambda)$ 的色彩”为对于人类观察者无法从 $l_1(\lambda)$ 区



分的光束的集合。所以在我们的案例中，我们会有 $\vec{c}(l_1(\lambda)) = \vec{c}(l'_1(\lambda)) = \vec{c}(l''_1(\lambda))$ 。因此在我们的数学模型中，一种（视网膜）色彩就是一类等效（一整类具有相同效果的）的光束（也就是说具有相同效果的每个光束都是同一种色彩）。

最终，我们意欲将色彩空间处理为一个线性矢量空间。例如，这会允许我们轻松借助坐标矢量（coordinate vectors）就可以表示色彩，同时也会告诉我们怎样通过将几种“主色彩”混合在一起以产生所想要的色彩。

然后，我们的下一步是指出怎样将两种色彩加在一起。我们从物理学中得知当两个光束， $l_1(\lambda)$ 和 $l_2(\lambda)$ 被加在一起，它们只是以光分布 $l_1(\lambda) + l_2(\lambda)$ 的特征组成了一个合并光束。如此，我们尝试定义两个色彩的加法，其结果为两个光束的加法（和）所对应的色彩。

$$\vec{c}(l_1(\lambda)) + \vec{c}(l_2(\lambda)) := \vec{c}(l_1(\lambda) + l_2(\lambda))$$

要让这个公式被正确定义，必须以实验方式验证我们选择哪个光束作为这种色彩的表达并没有什么不同。实际上，如果 $\vec{c}(l_1(\lambda)) = \vec{c}(l'_1(\lambda))$ ，那么我们必须验证（再次借助图示Figure 19.3的装置）：对于所有的 $l_2(\lambda)$ ，我们具有如下关系 $\vec{c}(l_1(\lambda)) + \vec{c}(l_2(\lambda)) := \vec{c}(l'_1(\lambda)) + \vec{c}(l_2(\lambda))$ ，也即，我们必须验证光束 $l_1(\lambda) + l_2(\lambda)$ 对于 $l'_1(\lambda) + l_2(\lambda)$ 不可区分。这个属性确实可被实验证实。

我们的下一步是尝试定义用非负实数 $\alpha$ 乘以一个色彩坐标意味着什么。再次，因为我们可以用一个正数（scalar）乘以一个光束，尝试如下定义：

$$\alpha \vec{c}(l_1(\lambda)) := \vec{c}(\alpha l_1(\lambda)) \quad (19.4)$$

再一次，我们需要验证这个操作的行为不依赖我们对光束的选择。因此当 $\vec{c}(l_1(\lambda)) = \vec{c}(l'_1(\lambda))$ ，我们必须验证对于所有的 $\alpha$ 值我们也具有 $\vec{c}(\alpha l_1(\lambda)) = \vec{c}(\alpha l'_1(\lambda))$ ，也即是说，我们必须验证光束 $\alpha l_1(\lambda)$ 对于 $\alpha l'_1(\lambda)$ 是无法区分的。这个属性也可以通过实验证实。

## 19.3 色彩匹配（Color Matching）

色彩匹配实验的目标是要证实色彩空间的维度是3维。另外，它会给出我们一种计算方式（类似于方程式（19.1）），这种方式专门用于映射光束到其关于一个特定基(basis)的色彩坐标。

使用图示Figure 19.3的装置，观察者看到两个屏幕。在屏幕的左侧会给他们展示一个纯的测试光束 $l_\lambda$ ，拥有某种固定波长。在屏幕的右侧，他们会观察到一束光，由3个纯匹配光束的正值线性组合构成，波长分别为435，545和625纳米。观察者的目标是调整右侧的3个旋钮，控制匹配光束的强度，以便加权过的3种匹配光束对于测试光束是不可区分的。针对固定的波长 $\lambda$ ，同时将旋钮设置值看可作 $k_{435}(\lambda), k_{545}(\lambda)$ 和 $k_{625}(\lambda)$ ，目标便是设置这些旋钮以便光束 $k_{435}(\lambda)l_{435} + k_{545}(\lambda)l_{545} + k_{625}(\lambda)l_{625}$ 为 $l_\lambda$ 的条件等色。如果用户不能成功，那么他们被允许从右侧移动一个或者多个匹配光束到左侧，同时将他们合并到测试光束并代替测试光束。在扩展色彩空间的数学里，这等同于允许某个数量值 $k(\lambda)$ 变动为负值。

这个处理在可见光范围内针对所有的波长 $\lambda$ 被重复。当匹配实验被执行，我们发现用户确实可以成功获得针对所有可见光波长的匹配。

更进一步，实验给出了我们3个所谓的匹配函数 $k_{435}(\lambda)$ ,  $k_{545}(\lambda)$ 和 $k_{625}(\lambda)$ ，这些函数被图形化展示在图示\testFigure19.1。注意，在3种波长435, 545, and 625的每一个之上，若其匹配函数被设置为1，那么其余两个会被设置为0。

我们将实验的结果提炼为下面公式：

$$\vec{c}(l_\lambda) = [\vec{c}(l_{435}) \ \vec{c}(l_{545}) \ \vec{c}(l_{625})] \begin{bmatrix} k_{435}(\lambda) \\ k_{545}(\lambda) \\ k_{625}(\lambda) \end{bmatrix}$$

根据关于对 $\vec{c}$ 的线性映射的某种合理连续性假设,我们可以升级这个方程式，且应用到所有的混合光束。这么做之后，我们获得下面公式：

$$\vec{c}(l_\lambda) = [\vec{c}(l_{435}) \ \vec{c}(l_{545}) \ \vec{c}(l_{625})] \begin{bmatrix} \int_{\Omega} d\lambda \ l(\lambda) \ k_{435}(\lambda) \\ \int_{\Omega} d\lambda \ l(\lambda) \ k_{545}(\lambda) \\ \int_{\Omega} d\lambda \ l(\lambda) \ k_{625}(\lambda) \end{bmatrix} \quad (19.5)$$

非正式地说，这个方程式对应了一种思路，就是每种混合光束实际只是一种（数不清的）纯光束的线性组合。

根据前面内容我们汇总如下：

\* 色彩空间是3维的。

\*  $[\vec{c}(l_{435}) \ \vec{c}(l_{545}) \ \vec{c}(l_{625})]$ 构成了一个色彩空间的基(basis)。

\* 匹配函数可以被用于相关于这个基(basis)给出我们任何光分布的坐标。

正如我们对LMS线性空间所做的，我们可以在图示Figure 19.8中可视化这种线性空间。注意，在这种情形中，拉索曲线（lasso cuve）依次经过了（基(basis)上的）每个轴，因为我们的基(basis)（坐标对应的）色彩是单色的（纯色的）。仍然要注意，在这种基(basis)中，拉索曲线（lasso cuve）确实离开了第一象限。

## 19.4 各种色彩空间基(Bases)

正如各种矢量空间，色彩空间可以借助很多不同的基（bases）被描述。以方程（19.5）为基础，我们可以插入任何（非奇点） $3 \times 3$  矩阵M和它的反转矩阵以获得如下的表达：

$$\begin{aligned} \vec{c}(l_\lambda) &= ([\vec{c}(l_{435}) \ \vec{c}(l_{545}) \ \vec{c}(l_{625})] M^{-1}) \left( M \begin{bmatrix} \int_{\Omega} d\lambda \ l(\lambda) \ k_{435}(\lambda) \\ \int_{\Omega} d\lambda \ l(\lambda) \ k_{545}(\lambda) \\ \int_{\Omega} d\lambda \ l(\lambda) \ k_{625}(\lambda) \end{bmatrix} \right) \\ &= [\vec{c}_1 \ \vec{c}_2 \ \vec{c}_3] \begin{bmatrix} \int_{\Omega} d\lambda \ l(\lambda) \ k_1(\lambda) \\ \int_{\Omega} d\lambda \ l(\lambda) \ k_2(\lambda) \\ \int_{\Omega} d\lambda \ l(\lambda) \ k_3(\lambda) \end{bmatrix} \quad (19.6) \end{aligned}$$



此处 $\vec{c}_i$ 描述了一个新的色彩基(basis), 定义如下:

$$[\vec{c}_1 \ \vec{c}_2 \ \vec{c}_3] = [\vec{c}(l_{435}) \ \vec{c}(l_{545}) \ \vec{c}(l_{625})]M^{-1}$$

还有 $k(\lambda)$ 函数可以组成新的关联匹配函数, 定义如下:

$$\begin{bmatrix} k_1(\lambda) \\ k_2(\lambda) \\ k_3(\lambda) \end{bmatrix} = M \begin{bmatrix} k_{435}(\lambda) \\ k_{545}(\lambda) \\ k_{625}(\lambda) \end{bmatrix} \quad (19.7)$$

以此类推, 可知存在3种主要的概念方式, 可凭借其来指定一个色彩空间的基(basis):

- 以用于任何固定(特定)色彩空间的基(basis)为初始, 比如 $[\vec{c}(l_{435}) \ \vec{c}(l_{545}) \ \vec{c}(l_{625})]$ , 我们就可以关联这个固定基(basis), 通过制定一个反转的3x3矩阵 $M$ 来描述一个新基(basis)。
- 我们可以指定3种(非同平面)实际色彩 $\vec{c}_i$ 。每个这样的 $\vec{c}_i$ 可以通过某种光束 $l_i(\lambda)$ 所生成(指定)。(然后我们能插入每个这样的 $l_i(\lambda)$ 到方程(19.5)的右边, 来获得关联于 $[\vec{c}(l_{435}) \ \vec{c}(l_{545}) \ \vec{c}(l_{625})]$ 基(basis)的色彩坐标。这种方式完全确定了基(basis)变换矩阵 $M$ 。
- 我们可以直接指定3个新匹配函数。要成为有效匹配函数, 这些函数必须从一个类似方程(19.6)的基(basis)变换公式中产生, 就如方程(19.7), 就是说每个匹配函数必须是 $k_{435}(\lambda)$ ,  $k_{545}(\lambda)$ ,  $k_{625}(\lambda)$ 函数的线性组合。若我们尝试使用不是这种形式的匹配函数, (它们-这些方程)就不能保留条件等色关系; 对于人类不可区分的光束可能被映射到不同的坐标矢量, 同样反之亦然。理想地, 数码相机色彩传感器也应遵从这种形式, 使得相机可以真正地捕获色彩, 也就是说, 是条件等色相关的。另外, 被造出的相机的感知函数一定也是完全(处处)非负的。

在 $[\vec{c}(l_{435}) \ \vec{c}(l_{545}) \ \vec{c}(l_{625})]$ 之外, 我们已经看到另一种用于色彩空间的基(basis)。实际上, 方程(19.1)中的匹配函数也描述了一种色彩空间基(basis), 这里色彩的坐标被称为 $[S, M, L]^t$ 。实际的基(basis)由3种色彩构成, 我们称之为 $[\vec{c}_s, \vec{c}_m, \vec{c}_l]$ 。色彩 $\vec{c}_m$ 事实上是一种想象色, 因为在LMS色彩坐标中, 不存在真实的光束其坐标为 $[0, 1, 0]^t$

### 19.4.1 Gamut

假设我们想让一种基(basis), 其上所有的实际色彩都有非负坐标, 如此则其上拉索曲线(lasso cuve)决不会离开第一象限。那么我们发现定义这个象限的至少一个基矢量(basis vectors)一定位于实际色彩锥体的外部。这样一个基矢量一定是一种想象色。这种现象仅是由于拉索曲线(lasso cuve)本身的形状所导致; 我们不能找到3个矢量, 它们即都触碰到拉索曲线(lasso cuve)同时还在它的正数区间(象限)中包含整个拉索曲线(lasso cuve)。

反之, 如果所有我们的基矢量都是实际色彩, 如此在色彩锥体内, 一定存在某种实际的色彩不能用这种基的非负坐标记录。我们就说这些色彩位于这个色彩空间的显示范围(gamut)之外。

### 19.4.2 特定的色彩基(basis) (Specific Bases)

用于色彩空间的中央标准基（central standard basis）被称作 $XYZ$ 基。这个基被3个匹配函数 $k_x(\lambda), k_y(\lambda), k_z(\lambda)$ 所指定，被图示在Figure 19.1中左下角。对于关联于这个基(basis)的某种色彩的坐标通过一个被称作 $[X, Y, Z]^t$ 形式的坐标矢量被给出。这种3D色彩基被展示在图示Figure 19.9。其中最下边一行展示了色彩锥体的 $X + Y + Z = K$ 平面。这个平面是用于可视化色彩空间的典型的2D图示。

3种实际匹配函数被特意选择以保证它们总是正值，同时便于色彩的“Y坐标”表达色彩的整体被感知“亮度”。如此，Y总是被用作色彩的黑色或白色表达。关联的基 $[\vec{c}_x, \vec{c}_y, \vec{c}_z]^t$ 由3种想象色构成；这些轴在图示Figure 19.9中位于色彩锥体的外部。

整本书中，我们都在使用RGB坐标来描述色彩。事实上，存在多种不同的色彩空间都使用了这个名称。当前正在使用的被标准组织指定的RGB色彩空间为Rec.709 RGB色彩空间。（参考图示Figure 19.10）。

这种情形中基(basis) $[\vec{c}_r, \vec{c}_g, \vec{c}_b]^t$ 由3种意图匹配理想监视器/电视的3种Phosphors(监视器的基本显示元件)的实际色彩构成。拥有非负RGB坐标的色彩可以在一台监视器上被产生，同时可被说成是位于这个色彩空间的范围之内。这些色彩实际位于图示的第一象限之内。但是类似于基 $[\vec{c}(l_{435}), \vec{c}(l_{545}), \vec{c}(l_{625})]$ ，存在有用某种负RGB坐标表示的实际色彩。这样的色彩不能在监视器上被产生。另外，在一台监视器上，每个Phosphor的最大值为“1”，这也限制了可获取的色彩输出。

一个具有显示器可显示范围之外色彩的图像一定要某种程度上被映射到显示范围之内。对于这种情形的最简单方案只是将所有的负值限制到0。也存在用于显示范围映射更成熟的方法，只不过那超过了我们的学习范围。

在小节19.7.2中，我们会描述另一种常常碰到的色彩空间，被称作 $sRGB$ 。正如我们将看到的，其不是线性色彩空间。

## 19.5 反射建模（Reflection Modeling）

当来自一个光源的一束光 $i(\lambda)$ 照射到一个表面上，那个光的一部分被吸收，还有一部分被反射。被反射的光的部分依赖于表面材料的物理属性。让我们借助一个反射函数 $r(\lambda)$ 指定每个波长有多少被反射。在这种情形中，我们可以用每波长乘法建模从表面反射出来的光束

$$l(\lambda) = i(\lambda)r(\lambda)$$

（注意：这种方式没有建模光和表面之间的所有类型的交互，例如光闪耀（florescence）。另外，这个讨论中，我们没有让我们注意 $r(\lambda)$ 的对于光进入和逃离（出射）角度的依赖，这点将会在第21章完成。）这种乘法发生在每波长的基础上，同时在3D线性空间中不能被精确模拟。确实，两种可能在一种光源下反射出条件等色光束的材料，但是在另一种光源下却可能产生两种有区别的光束。

$$\vec{c}(i_1(\lambda)r_a(\lambda)) = \vec{c}(i_1(\lambda)r_b(\lambda)) \not\Rightarrow \vec{c}(i_2(\lambda)r_a(\lambda)) = \vec{c}(i_2(\lambda)r_b(\lambda))$$

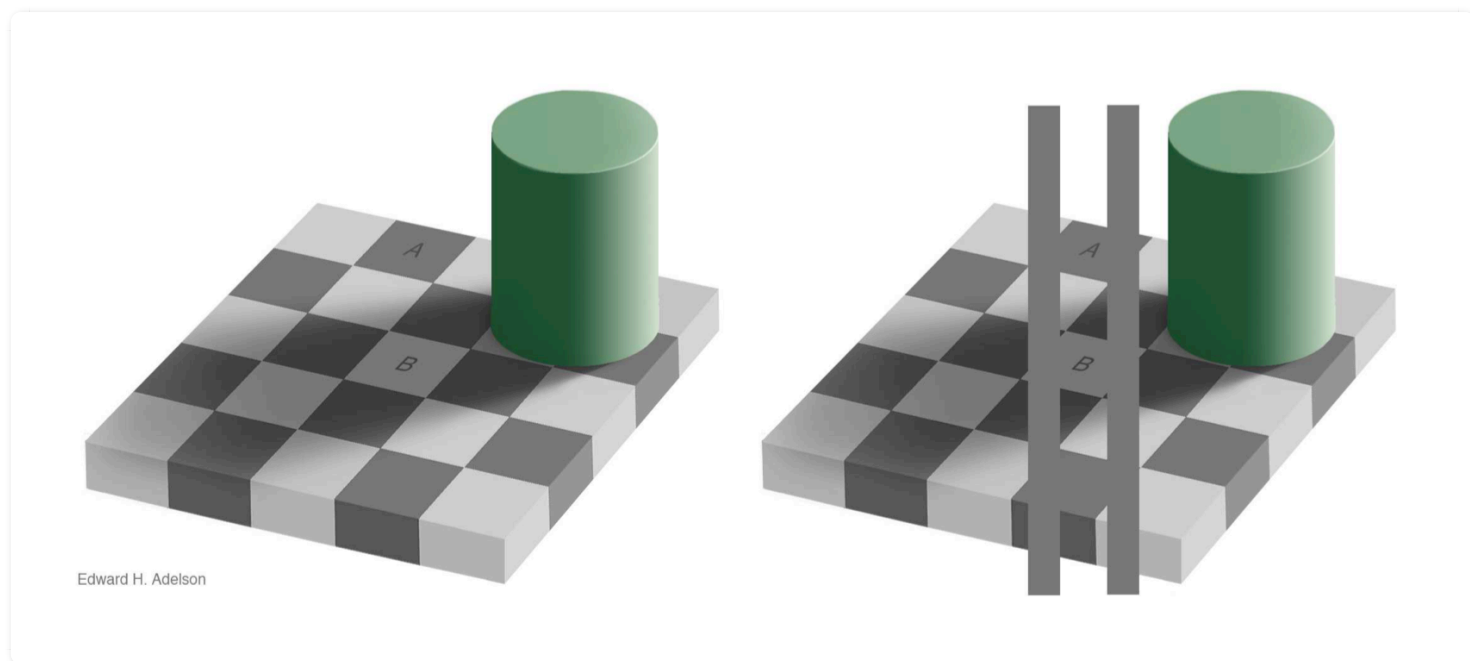
因此，在某些渲染情形中，在反射中建模光谱依赖就是重要的因素。但是更典型的情形是，我们

会忽略这个问题，而是通过比方说3个RGB色彩坐标（丢弃掉关于 $i(\lambda)$ 的频谱信息），同样再借助3个反射“因子”来建模表面的反射属性（很多关于光的渲染模型都是基于这个假设，比如Gauraud的Phong Shading）。

## 19.5.1 白平衡

假定在一个固定场景中，如果我们改变光源（照明物），那么一个图像中的色彩将会同时改变。例如，如果我们从荧光灯切换到白炽灯，相机（或观察者）所观察的色彩会整体移向黄色端。经常地，我们希望调整图像色彩以便于近似到这样的图像，就是说这个图像是在“特定的”经典光源下（比方说白天）拍摄。这个处理被称为“白平衡化”。这不是一个基(basis)变化，而是应用在所有色彩上的变换(transformation)。最简单的这种变换允许用户使用3个增益因子(gain factor)独立的缩放(scale)调整R, G, B坐标。

就如刚才描述的，我们不能期望总是可以成功产生场景在经典光源下的真实图像，因为当我们生成最初的图像时就已经丢失了频谱信息。确实，有些在经典光源下呈现出不同外观的物体在当前光源下可能是条件等色，从而在当前图像中拥有完全相同的色彩坐标。没有确切量级的简单白平衡化操作可以复原这种情形。



**Figure 19.4:** 被标记位A和B的方块事实上是相同的灰色视网膜着色，但是被有差异地感知到，这是由于我们（人类）的视觉处理和本地适应行为所导致。参考[1], © Edward c Adelson。

## 19.6 感官适应性 (Adaption)

来自视网膜的色彩数据在视觉系统中经历了明显的处理，同时人类没有途径可以直接感知一个被观察光束最原始的视网膜色彩坐标。这个处理导致了大量的标准化（流程）；可以适应于横跨整个视域的全局和本地变化趋势。

当光源变化时，比方说从烈日当空到漫天乌云，视网膜上每个直接被观察到的色彩坐标可能经历

剧烈的变化。但是这些剧烈的变化最终没有被人类感知到，同时针对每个目标物体的色彩很大程度上保持“恒定”。例如，一个令人恐惧的老虎会在极为广大范围的光源下被感知为黄色（激励人类逃跑）。这个现象被称为色彩恒定性（color constancy）。根据我们对白平衡的讨论，不能期望这种色彩稳定（恒定）机制是完美的，因为太多的频谱信息在从输入光谱光束到转换为视网膜中锥体细胞3元响应数值的处理中被丢失。但是在很大的程度上，人类视觉系统的这种处理可以保持正常工作，因而就允许我们将一种材料（老虎皮毛）当作实际处理一种色彩（令人惊慌的橙色）。

甚至当一个视域中的局部区域经历了一个亮度改变（比如说某部分视域进入阴影中），我们的视觉处理过程可能在这个区域也会有不同程度的适应，再一次保持最终被感知的色彩紧密的贴近实际观察的材料。（看例子图示Figure 19.4）。这个处理（机制）仍未被我们完全理解。

当我们在某个光源下拍摄一张照片，但是随后观看这张照片在一个不同的环绕光源下，观看者的适应状态既会被来自图片的光线也会被来自周围屋子的光线所影响。由于房间光线的效应，照片中的色彩可能最终“看起来是错的”。这也是为什么我们需要做上面所讲述的白平衡化的部分原因。

## 19.7 非线性色彩（non linear color）

我们已见到过（已经知道）视网膜色彩能够被建模为一个三维的线性空间。在本部分，我们会看到还存在使用一个不同的视网膜色彩表达集合的很多理由，这些表达被非线性关联到（之前的）色彩（线性）坐标。

### 19.7.1 可感知的（色彩）距离（Perceptual Distance）

任何线性色彩空间中，根据一个人类观察者对观察两种色彩会显得多大程度上的“不同”，那么两个色彩值的欧几里得距离不是一个准确的预估参考（predictor）。比如，人类对暗色变化的感知相比于亮色变化要敏感得多。已经被设计的多种色彩表达就提供了一个针对人的可感知色彩距离的更好匹配。从一个线性色彩空间到这种色彩表达的映射是非线性的。即便如此，我们仍会称呼这种表达为“色彩坐标”。

例如，一个这种坐标集合被称为 $L^*ab$ 坐标（全称CIELAB坐标）。 $L^*$ 坐标被称作“亮度”，同时被按照如下公式计算（除去非常小的值）：

$$L^* = 116 \left( \frac{Y}{Y_n} \right)^{\frac{1}{3}} - 16 \quad (19.8)$$

此处Y是XYZ基中第二个坐标，同时 $Y_n$ 为某种标准化因子。我们不会研究（讨论）这种坐标中a和b坐标的计算。

这个色彩空间有很多用处。实际上，如果我们在使用每坐标部件值8位或者更少位数的固定点表达（fix point representation，比如字节类型），我们最好存储我们的数据在一个可感知的平均空间中。当一个Y值的连续集合（continuum）被装进256个平均间隔的数位桶（bit bucket）中，在暗色区间会有显著的视觉缝隙（visual gap，指色彩的过渡不平滑）。在 $L^*ab$ 坐标中，为了解决这

个问题，更紧密（数量多）的数位桶（bit bucket）会被用于暗色区域。对应的在亮色区域需要较少的桶，但是亮色间的这种缝隙却不会被（人类视觉系统）感知到。

### 19.7.2 Gamma修正（Gamma Correction）

Gamma修正涉及一个类似于方程（19.8）中的幂运算符的变换。最初被用于在CRT设备中解决非线性问题，但是一直沿用至今，其中有部分理由是因为在固定点表达（fixed point representation）中比较好用。

**Gamma的本意：**在以前的时候，计算机图像显示在阴极射线管（cathode ray tubes - CRTs）之上。这种显示设备上每个像素被三个电压值所驱动，比方说 $(R', G', B')$ 。要使得从这个像素中输出的光拥有色彩坐标 $[R, G, B]^t$ ，这些输出的色彩的坐标与电压大约有如下的关系：

$$\begin{aligned} R &= (R')^{\frac{1}{.45}} \\ G &= (G')^{\frac{1}{.45}} \\ B &= (B')^{\frac{1}{.45}} \end{aligned}$$

如此，如果我们想从一个像素获得某种特定 $[R, G, B]^t$ 色彩坐标输出，我们需要用下面的电压值来驱动：

$$\begin{aligned} R' &= (R)^{.45} \\ G' &= (G)^{.45} \\ B' &= (B)^{.45} \end{aligned}$$

这种 $[R', G', B']^t$ 坐标值被称作（一种）色彩的Gamma修正的RGB坐标。标记(')表示这些是非线性色彩坐标。

**Gamma坐标的当前用法：**类似于 $L^*ab$ 色彩坐标，Gamma修正的色彩坐标相比线性色彩坐标具有更佳的感知一致性，因而对于电子色彩表达非常有用（见图示19.5）。实际中，流行的图像压缩技术，就像JPEG，就是始于 $[R', G', B']^t$  (Gamma修正的色彩坐标)，随后应用一个线性变换以获得一个新种类的色彩坐标称之为 $[Y', C'_B, C'_R]^t$ 。（注意此处 $Y'$ 不是通过一个简单幂等式被关联到 $Y$ ）。

对于 $[R, G, B]^t$ 坐标还可以应用一种相关的但更复杂的非线性变换，替换通过方程（19.9），以获得sRGB色彩坐标，被称为 $[R_{srgb}', G_{srgb}', B_{srgb}']^t$ 。现代LCD显示器设备被预设接受以这种坐标表示的输入。



**Figure 19.5:** 上部图片中的数据为一个线性坡度表示的色彩变化，以这种方式在一台监视器上展示了在 $[R', G', B']^t$ 上等分间隔的坐标数位桶（bit bucket）。下部图片中，在存储前色彩的线性坡度已经被Gamma修正。如此，在一台监视器上显示了等分间隔的 $[R, G, B]^t$ 坐标。这个图片中的色彩看起来快速离开黑色区域的控制，同时占用（耗费）了更多的坐标数位桶在亮色区域色彩值上。

### 19.7.3 数值近似（Quantization）

数值位于实数 $[0..1]$ 范围的sRGB色彩坐标必须以数值方式表达。（比如说在一个framebuffer或者文件格式中）这种表达经常用数值范围 $[0..255]$ 的固定点表达来实现。在C语言中，这可以用一个unsigned char类型来实现。我们能够指定这种数值近似的值和实数色彩坐标数值（比方说红色的坐标）之间的转换关系通过如下公式：

$$\begin{aligned} \text{byteR} &= \text{round}(\text{realR} * 255); \\ \text{realR} &= \text{byteR} / 255.0; \end{aligned}$$

注意，对于任意byteR的设置，如果我们变换到实数表达并且随后在变换到字节（固定点）表达，我们可以取回我们最开始的值。一个同样满足这种属性的替换性的转换关系可以使用下面公式来应用：

$$\begin{aligned} \text{byteR} &= \text{round}(f \geq 1.0 ? 255 : (\text{realR} * 256) - .5); \\ \text{realR} &= (\text{byteR} + .5) / 256.0; \end{aligned}$$

在此种表示中，和上面的表示不同，真实的被近似到字节（固定点表达）数值的数位桶（bit bucket）都是相同尺寸的。但是数值为0和255的字节值不能被对应地映射到0和1。（参考图示Figure 19.6）。





**Figure 19.6:** 在实数值和字节值这两种不同的映射中。从实数值变化到字节值，我们数量近似每个实数范围到被展示出来的整数值。从字节值到实数值我们使用青色的小箭头图示。

## 19.7.4 Gamma和图形（处理）（Gamma and Graphics）

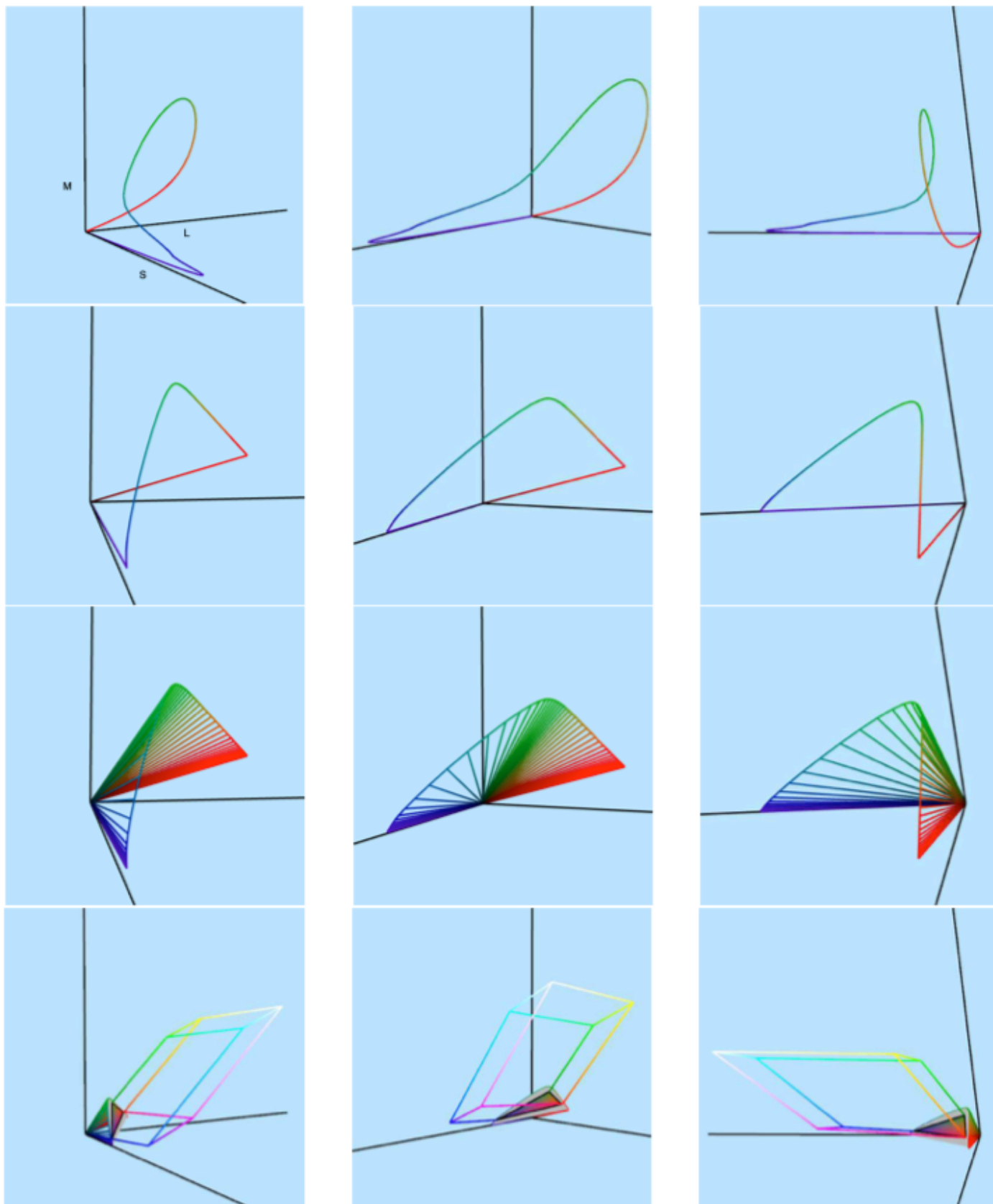
一方面，图像通常被存储在Gamma修正的坐标中，而监视器预期以Gamma修正的坐标来展示色彩。另一方面，计算机图形技术以线性模拟关联于光束的处理。如此，多数计算机图形计算应该以线性色彩表达的方式（被）完成。例如 $[R, G, B]^t$ 空间。举个例子，我们可以在 $[R, G, B]^t$ 空间近似地建模反射。其它的渲染步骤，比如给透明度建模，同时还有用于反锯齿目的的色彩值的混合，还有建模在光束中的线性处理，因此应该用线性色彩坐标被完成。在电子设备成像中，白平衡典型情况下在线性色彩空间执行。这种不一致性多年来已经成为困扰和乏味的根源。

这种情形最近已经得到改善。OpenGL的当前版本中，借助 `glEnable(GL_FRAMEBUFFER_SRGB)` 我们可以请求sRGB framebuffer。然后我们就能从fragment shader中传出 $[R, G, B]^t$ 坐标值，随后在发送到屏幕之前，他们会被在sRGB格式中被Gamma修正。

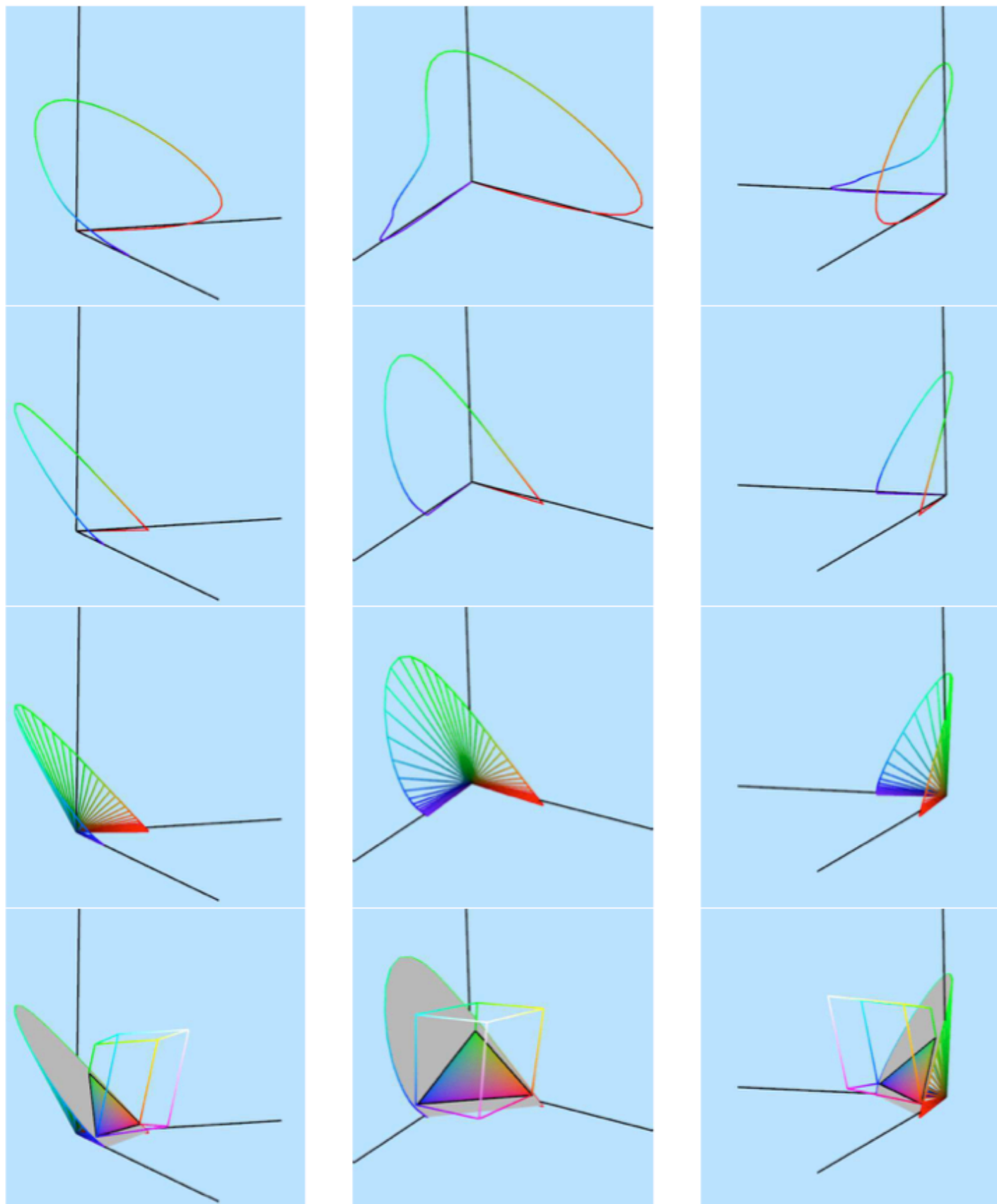
另外，对于纹理映射，我们可以指定输入图像为一个sRGB格式的纹理。借助调用

```
glTexImage2D(GL_TEXTURE_2D, 0, GL_SRGB, twidth, theight, 0, GL_RGB, GL_UNSIGNED_BYTE, pixdata)
```

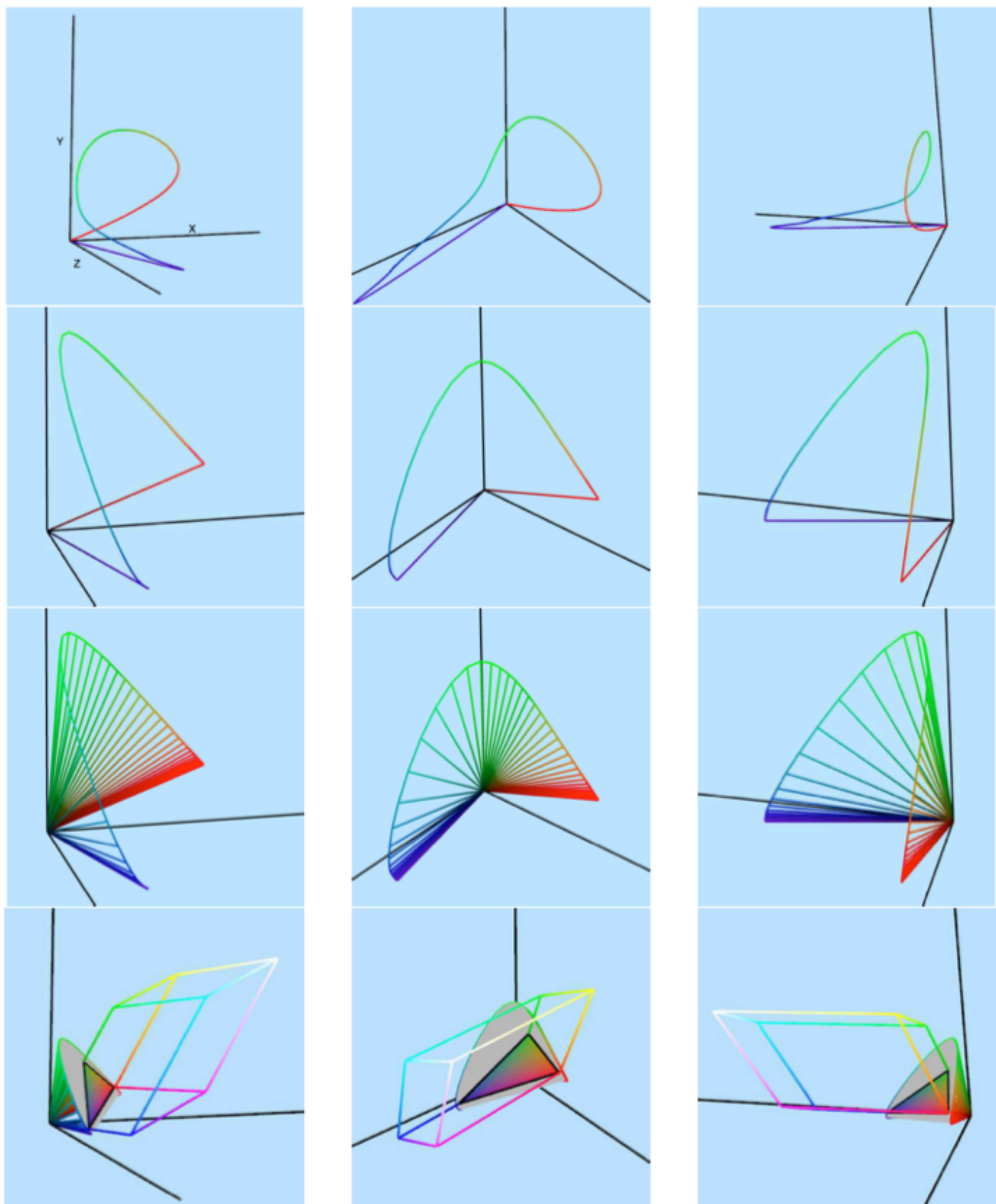
来完成。不论何时这个纹理在fragment shader中被访问，数据被shader使用前会首先被转化为线性 $[R, G, B]^t$ 坐标值。



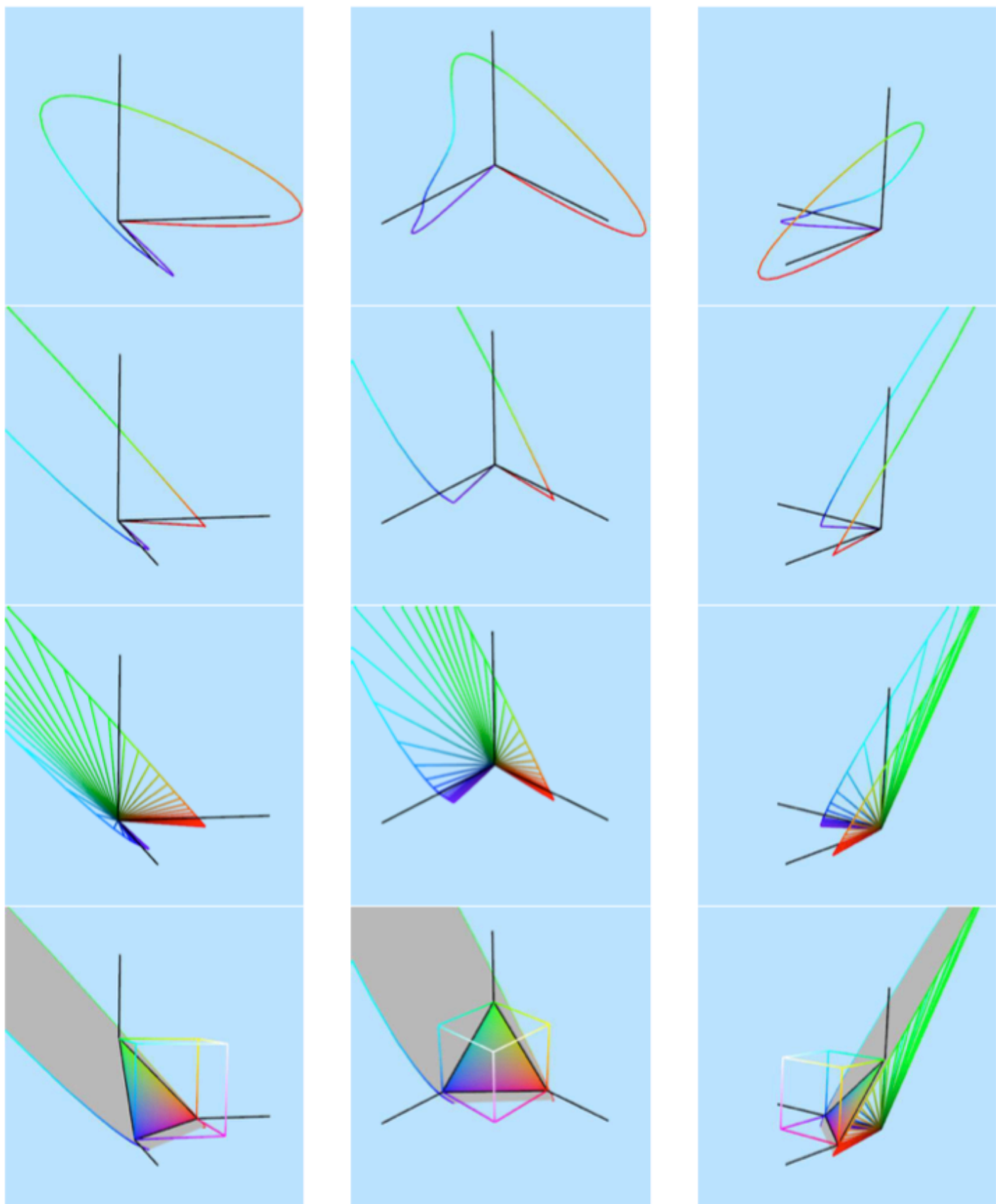
**Figure 19.7:** LMS色彩空间 (LMS color space) : 每行展示了一个不同的视角。第一行: 拉索曲线 (lasso curve) 在LMS坐标内被绘制。第二行: 被“标准化”的拉索曲线是一个马靴曲线 (horseshoe curve) 。第三行: 很多射线将马靴曲线连接到原点。第四行: 一个位于拉索曲线上的凸状色彩锥体 (convex cone) 。其中的三角形展示了在这个切片上的实际色彩。这些色彩被表达为监视器色彩 R, G, B的正值之和。RGB色彩立方体 (RGB color cube) 被展示在线框图中。



**Figure 19.8:** 由匹配实验生成的色彩空间。



**Figure 19.9:** XYZ色彩空间（XYZ color space）。



**Figure 19.10:** RGB色彩空间（RGB color space）。