Animated Color Cube

Asahi Sugiyama[1*] and *Michael* Cohen[1**]

[1]Spatial Media Group; University of Aizu; Tsuruga, Ikki-machi; Aizu-Wakamatsu 965-8580; Japan

Abstract. The RGB color model is framed by black, white, three additive primary colors, and three subtractive secondary colors, but there are many hues between them. We used a color cube for representing dimensional bases and intermediate hues and animated it to visualize color interpolation processes.

1 Introduction

The goal of this project is to make an animated color cube, showing how color is determined in RGB and CMY color models by interpolating between primary color dimensions.

1.1 RGB and CMY

A color model is an abstract mathematical model describing the way colors can be represented as tuples of numbers, typically as three or four values or color components [1]. Three-value color model examples are RGB: red, green, and blue, CMY: cyan, magenta, and yellow, and RYB: red, yellow, blue. Four-value color model examples are CMYK (where "K" stands for "key," or black) and RGBA (where A stands for "alpha," or transparency).

1.1.1 The RGB color cube

As seen in Fig. 1, the RGB color model is a type of color expression, based on three additive primary colors: red, green, and blue.

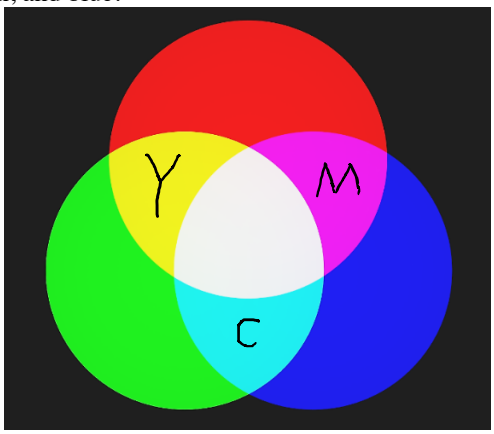

Fig. 1. The RGB color model yields CMY as secondary colors.

The RGB color model is used for sensing, representing, and displaying images in emissive (additive) systems. tan

1.1.2 The CMY color cube

As seen in Fig. 2, the CMY color model is based on three subtractive primary colors: cyan, magenta and yellow, which become secondary colors in the RGB color model.

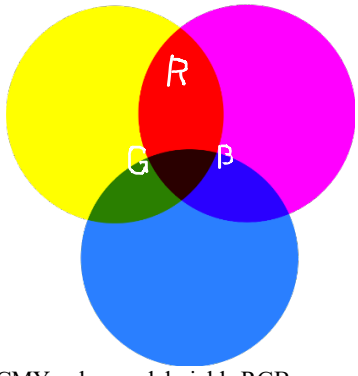

Fig. 2. The CMY color model yields RGB as secondary colors.

The CMY color model, often augmented with black, is used in subtractive processes such as printing.

1.2 Color cube

The color cube is a three-dimensional physical model wich can be used to represent the gamut or set of colors reproducible in a given medium, by combining three primary colors atop a base color [2]. The basic color cube is framed by red, green, and blue; cyan, magenta, and yellow; white and black; and the colors between them, expressing the gamut of displayable colors.

* sugi1250157@gmail.com, ** mcohen@u-aizu.ac.jp

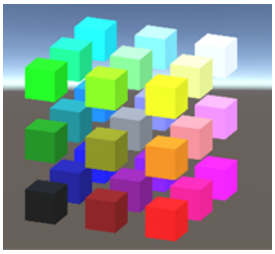

Fig. 3. A 3^3 color cube

If black is set as base color and RGB is established as three primary colors, the color cube is framed by red, green, blue. Such as color cube, as shown in Fig. 3, shows us how the color was determined in RGB in more detail. It is identical, except for rotation, to a CMY color cube, as shown in Fig. 4.

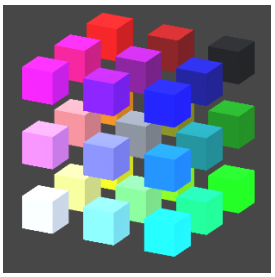

Fig. 4. A 3^3 CMY color cube

2 Implementation

The first version of our project used 6 primary colors and black and white atoms, with which we could combine pairs of colors to make intermediate (secondary) colors by interpolating or mixing those two. At each checkpoint, determined by sampling frequency, first-order atoms ("cubies") of primary color have intermediate saturation and value, split to travel in orthogonal direction as well as along the original axes. When atoms from separate axes meet, they combine to instantiate a composed color atom. These second-order atoms are also animated, propelled along a third dimension to meet passes of its second-order atoms, whereupon they spawn third-order atoms in play but continue along to keep rendezvous.
The animation starts from black corner cubie ("0G") which spawns red, green and blue cubies ("1G") along axes of primary colors, which make secondary ("2G") colors, which make tertiary ("3G") colors before terminating at white at the diagonally opposite corner.

2.1 Edge and Vertices

The number of vertices on a side can be designated n, the number of nodes along each side of regular (equilateral) rectangular prism. An n^3 color cube has $(n-1)$ nodes on each side.

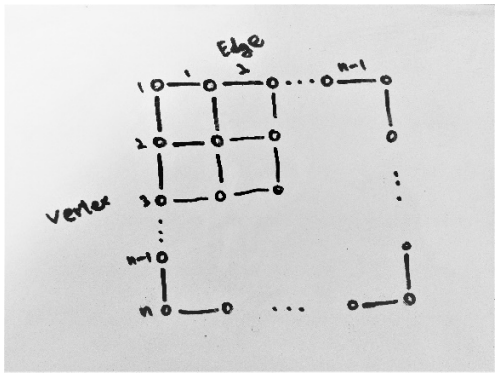

Fig. 5. Edges and vertices on a side

The edges on a face can be divided into horizontal and vertical sets. Each of them has $n(n-1)$ edges. So, the total number of edges on a single face is $2n(n-1)$.

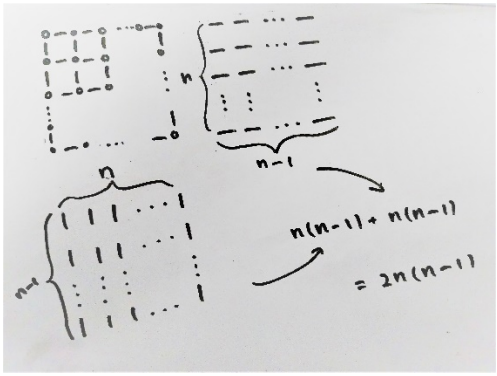

Fig. 6. Edges on a face

Each pair of adjacent faces is connected by n^2 edges.

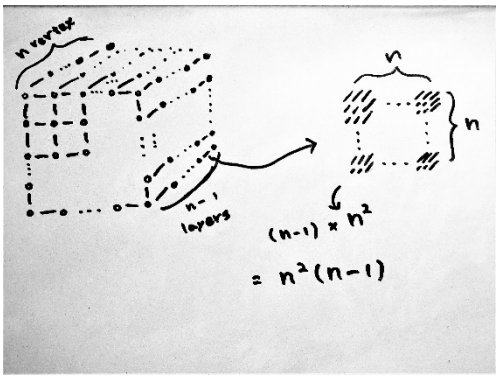

Fig. 7. Edges between layers

In total, there are n faces and $(n-1)$ sets of n^2 connections among them. The total number of edges is therefore

$$2n^2(n-1) + n^2(n-1) = 3(n^3-n^2). \quad (1)$$

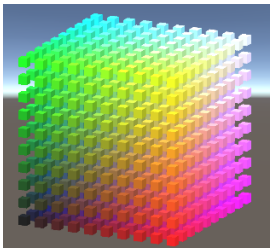

Fig.8. A 10^3 color cube

2.2 Unity Development

This color cube and its animation were developed in Unity. In Unity, a script can be attached to a scene object, which collectively comprise a scene. Attached to each cubie (component of the color cube), is a script that generates another cubie.

2.2.1 "0G" cubie

0G (black) cubie spawns (generates) three 1G cubies along three primary axes: red, green, and blue.

```
1GCubeGenerate(Instantiate(1Gcubie),
num, 1, Vector3.right);
1GCubeGenerate(Instantiate(1Gcubie),
num, 1, Vector3.up);
1GCubeGenerate(Instantiate(1Gcubie),
num, 1, Vector3.forward);
```

2.2.2 "1G" cubie

1G cubies interpolate color between black and each of primary colors. The cubies generate both 2G cubies and 1G cubies. 2G cubies are generated orthogonally and 1G cubies are generated along original direction of travel.

```
if    (isCallOnce    ==    false    &&
transform.position                    ==
targetAxis*2f*cnt)
{
   if (cnt < num - 1) {
   1GCubieGenerate(Instantiate(this.g
ameObject),num,cnt+1,targetAxis);
   }
   if (targetAxis != Vector3.forward)
{
    2GCubeGenerate(Instantiate(2Gcubie
),    num,    1,    transform.position,
Vector3.forward);
   }
   if (targetAxis != Vector3.up) {
    2GCubeGenerate(Instantiate(2Gcubie
),    num,    1,    transform.position,
Vector3.up);
   }
   if (targetAxis != Vector3.right) {
    2GCubeGenerate(Instantiate(2Gcubie
),    num,    1,    transform.position,
Vector3.right);
   }
   isCallOnce = true;
}
```

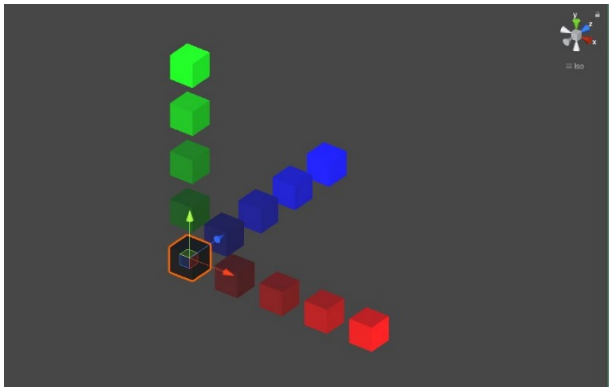

Fig. 9. 0G and 1G cubies

2.2.3 "2G" cubies

Each 2G cubie's color is a blend of two primary colors. The intersection of 2G cubies generates 3G and 2G cubies. 3G cubies are generated orthogonally and 2G cubie is generated along original direction of travel.

```
if    (isCallOnce    ==    false    &&
transform.position    ==    startPosition
+ targetAxis * 2f) {
   if (cnt < num - 1) {
   2GCubeGenerate(Instantiate(this.ga
meObject),    num,    cnt    +    1,
transform.position, targetAxis);
   }
   if(transform.position.x == 0) {

   3GCubeGenerate(Instantiate(3Gcubie
   ),   num,   1,   transform.position,
   Vector3.right);
   }
   if (transform.position.y == 0) {
    3GCubeGenerate(Instantiate(3Gcubie
),    num,    1,    transform.position,
Vector3.up);
   }
   if (transform.position.z == 0) {
    3GCubeGenerate(Instantiate(3Gcubie
),    num,    1,    transform.position,
Vector3.forward);
   }
   isCallOnce = true;
}
```

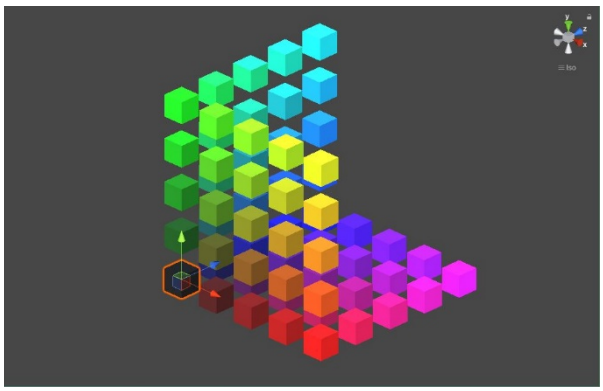

Fig. 10. 0G, 1G, and 2G cubies

2.2.4 "3G" cubies

Each 3G cubie's color is a blend of three primary colors. The cubie generates their copy (instance) along direction of travel.

```
if (isCallOnce == false && transform.position == startPosition + targetAxis * 2f) {
   if (cnt < num - 1) {
   3GCubeGenerate (Instantiate(this.gameObject), num, cnt + 1, transform.position, targetAxis);
   }
    isCallOnce = true;
}
```

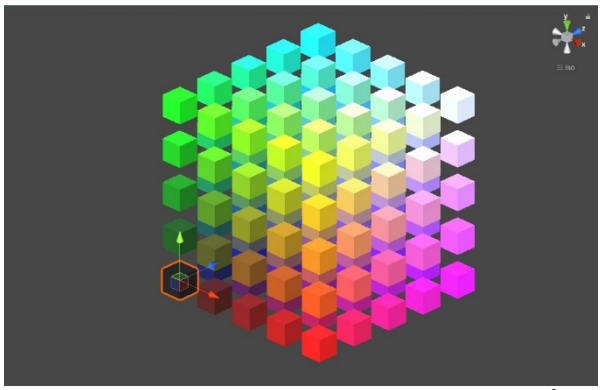

Fig. 11. 0G, 1G, 2G, and 3G cubies comprising a 5^3 color cube

3 Implications

In animation [3], Black cubie generates three "1G" cubies of primary color: red, green, and blue. Each "1G" cubie spawn syounger siblings along same axes and "2G" cubies perpendicularly. Similarly, "2G" cubies spawn themselves along their axes and tertiary cubies "3G" perpendicularly, 3G cubies spawn their sibling along their axes. At the end of the animation, cubies that were spawned from cyan, magenta and yellow are at white. So, there were *1* 0G cubie, *3(n-1)* 1G cubies, $3*2(n-1)^2$ 2G cubies, and $3*2(n-1)^3$ 3G cubies, totally $6n^3-12n^2+9n-2$ cubies more than n^3 of object cube. This is because cubies are generated to move on every internal vertex of the color cube.

Table 1. Number of cubies

n	Object: n^3	Animation: $6n^3-12n^2+9n-2$
1	1	1
2	8	16
3	27	79
5	125	493
10	1000	4888

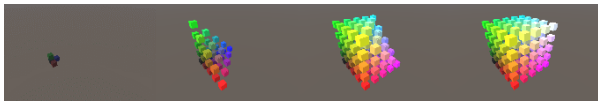

Fig. 12. Sequence of constructing color cube

This animation helps us to sense how colors are determined in the RGB color model.
To enhance the animation, sound was added, displaying an impulse every time a new cubie was spawned. The number of such instantiations depends upon the size of the target cube, as shown in Table 1. There is some redundancy, as multiple simultaneous intersections spawn colocated cubies, as shown in Table 2. The intensity of combined synchronized sound is a cue to multiplicity of instantiation to that level, as shown in Table 3.

Table 2. Number of generated cubies in each sequence in 3^3 color cube

sequence	1	2	3	4	5	6	7
Vertex base	1	3	6	7	6	3	1
Edge base	1	3	9	15	15	9	3
Actual number	1	3	9	18	24	18	6

The sound level based on actual number.

Table 3. Volume of sound in each sequence in 3^3 color cube

sequence	Vertex base	Actual number	Real sound level	Volume of sound
1	1	1	18	6.34726E-5
2	3	3	20	5.712534E-4
3	6	9	25	5.141281E-3
4	7	18	30	2.056512E-2
5	6	24	33	3.656022E-2
6	3	18	30	2.056512E-2
7	1	9	23	2.285013E-3

The color cube makes it easier to understand the RGB color model. The color cube can also show how the color is determined in CMY color model by changing base color and three primary colors. Add some options as future work, for example, when cubie generated cubie make sound and make this animation visible as stereographic.

References

1. L. I. Rui-juan, Study on color space conversion model from RGB to CIEXYZ, Packaging Engineering, **3**, 2009.
2. K. W. Davies, Color cube model, U. S. Patent, **5634795**, Jun. 3,1997.
3. https://drive.google.com/file/d/1g4WBDgfLUFkyGTx1G_uE90iKt_xuxKod/view?usp=sharing