Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Саратовский государственный технический университет

имени Гагарина Ю.А.»

Кафедра «Прикладные информационные технологии»

**Базовые параметры изображений**

**Отчет по курсу**

**«Методы обработки сигналов и изображений**

Выполнил:

Студент группы м2-ИФСТ-11

Маркелов Александр Сергеевич

Проверил:

ст. преподаватель каф. ПИТ

Пиминов Дмитрий Алексеевич

Саратов 2025

**1 Теоретическая часть**

**1.1 Глубина цвета**

Глубина цвета(bit depth или color depth) — это количество бит, используемых для представления цвета каждого пикселя в изображении. Чем больше бит, тем больше возможных цветов может быть представлено. Глубина цвета обычно указывается в битах на канал (канал — это один из компонентов цвета, например, красный, зеленый или синий в модели RGB).

* 8-битная глубина цвета: 256 уровней интенсивности для каждого канала (2^8 = 256). В модели RGB это означает 256 \* 256 \* 256 = 16,7 миллионов цветов.
* 16-битная глубина цвета: 65536 уровней интенсивности для каждого канала (2^16 = 65536). В модели RGB это означает 65536 \* 65536 \* 65536 = 281 трлн цветов.

Глубина цвета влияет на качество изображения: чем выше глубина цвета, тем более детализированные и точные оттенки можно отобразить, что можно так же видеть на рисунке 1, который представлен ниже:



Рисунок 1 - Разница в передачи оттенков при различном глубине цвета

* 1 бит (монохромное изображение): Каждый пиксель может иметь только два состояния (например, черный или белый). Общее количество цветов: 21=2.
* 8 бит (глубина цвета для одного канала): Если используется один канал (например, градации серого), то доступно 28=256 оттенков.
* 24 бита (True Color): Три канала (красный, зеленый, синий) по 8 бит каждый. Общее количество цветов: 224=16,777,216.
* 32 бита: Часто добавляется альфа-канал (прозрачность), что дает возможность управлять прозрачностью каждого пикселя.

Чем выше глубина цвета, тем более детализированным и реалистичным будет изображение, но при этом увеличивается объем данных.

Пример представлен на рисунке 2.

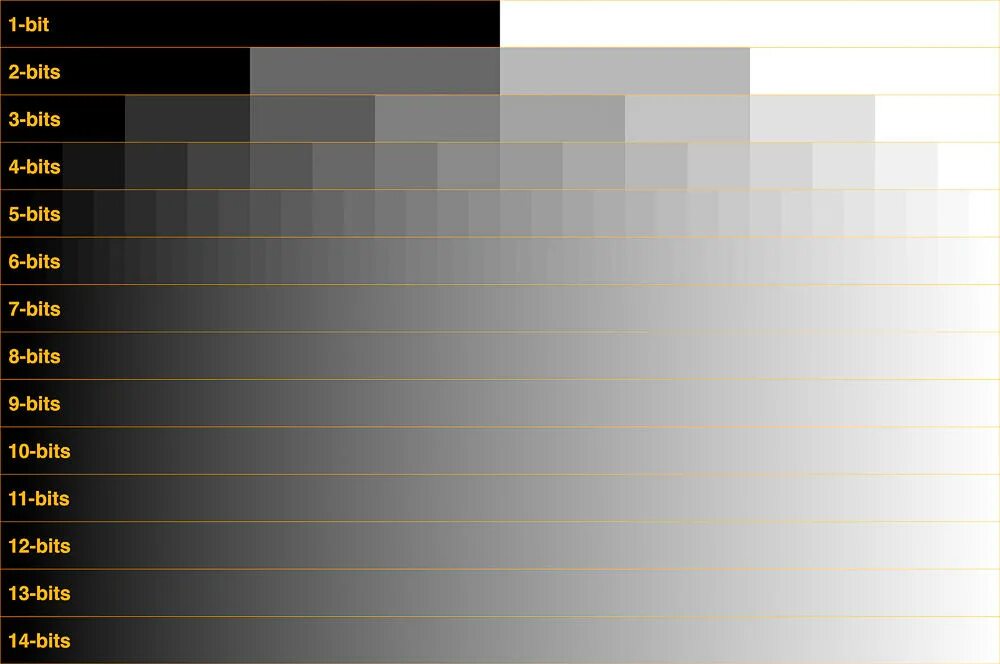


Рисунок 2- Сравнение детальности передачи цветов при различной битности изображения.

**1.2 Цветовые модели**

* + 1. **RGB**

RGB — это аддитивная цветовая модель, которая используется для представления цветов путем комбинирования трех основных (первичных) цветов: красного (Red), зеленого (Green) и синего (Blue). Название модели происходит от первых букв этих цветов на английском языке. Эта модель широко применяется в цифровых технологиях, таких как экраны телевизоров, мониторы компьютеров, смартфоны, камеры и другие устройства, где цвет формируется за счет излучения света.

### Основные принципы RGB

RGB основана на аддитивном смешивании цветов, что означает, что цвета создаются путем добавления света разной интенсивности. В отличие от субтрактивных моделей (например, CMYK, используемой в печати), где цвета получаются за счет поглощения света, в RGB чем больше света вы добавляете, тем ближе результат к белому.

* **Красный (R), зеленый (G)** и **синий (B)** — это базовые цвета, которые человеческий глаз воспринимает через колбочки в сетчатке.
* Каждый из этих цветов имеет интенсивность, обычно измеряемую в диапазоне от 0 до 255 (в 8-битной системе) или от 0 до 1 (в нормализованной форме).
* Комбинируя различные уровни интенсивности красного, зеленого и синего, можно получить практически любой цвет в видимом спектре.

Примеры:

* (0, 0, 0) — полное отсутствие света, черный цвет.
* (255, 0, 0) — чистый красный.
* (0, 255, 0) — чистый зеленый.
* (0, 0, 255) — чистый синий.
* (255, 255, 255) — максимальная интенсивность всех цветов, белый.
* (255, 255, 0) — красный + зеленый = желтый.

### Как работает RGB

Модель RGB имитирует восприятие цвета человеческим глазом. Наши глаза содержат три типа колбочек, каждая из которых чувствительна к определенному диапазону длин волн света: длинным (красный), средним (зеленый) и коротким (синий). Устройства, использующие RGB, такие как экраны, состоят из множества пикселей, каждый из которых содержит субпиксели красного, зеленого и синего цветов. Регулируя яркость этих субпикселей, экран создает иллюзию различных оттенков.

Например:

* Если включить только красный субпиксель на максимум, пиксель будет красным.
* Если включить красный и зеленый на максимум, получится желтый.
* Если все три субпикселя горят на максимальной интенсивности, пиксель выглядит белым.

### Цветовое пространство RGB

RGB определяет так называемое цветовое пространство — набор всех возможных цветов, которые можно получить в рамках этой модели. Однако важно понимать, что RGB не охватывает весь спектр цветов, видимый человеческим глазом. Это зависит от конкретной реализации RGB, например:

* **sRGB** (стандартное RGB) — наиболее распространенное цветовое пространство, используемое в интернете и потребительских устройствах. Оно покрывает лишь часть видимого спектра.
* **Adobe RGB** — более широкое цветовое пространство, используемое в профессиональной фотографии и дизайне.
* **ProPhoto RGB** — еще более широкое пространство, охватывающее почти весь видимый спектр, но редко используемое из-за сложности реализации.

### Применение RGB

1. **Цифровые дисплеи**: Телевизоры, мониторы, смартфоны и проекторы используют RGB для отображения изображений.
2. **Графический дизайн**: Программы вроде Photoshop и Illustrator позволяют задавать цвета в RGB для работы с цифровыми проектами.
3. **Веб-дизайн**: Цвета на сайтах часто задаются в шестнадцатеричном формате (hex), который основан на RGB, например, #FF5733 (где FF — красный, 57 — зеленый, 33 — синий).
4. **Фотография и видео:** Камеры записывают изображения, преобразуя свет в значения RGB.

На рисунке 3 представлен охват RGB цветовой модели :

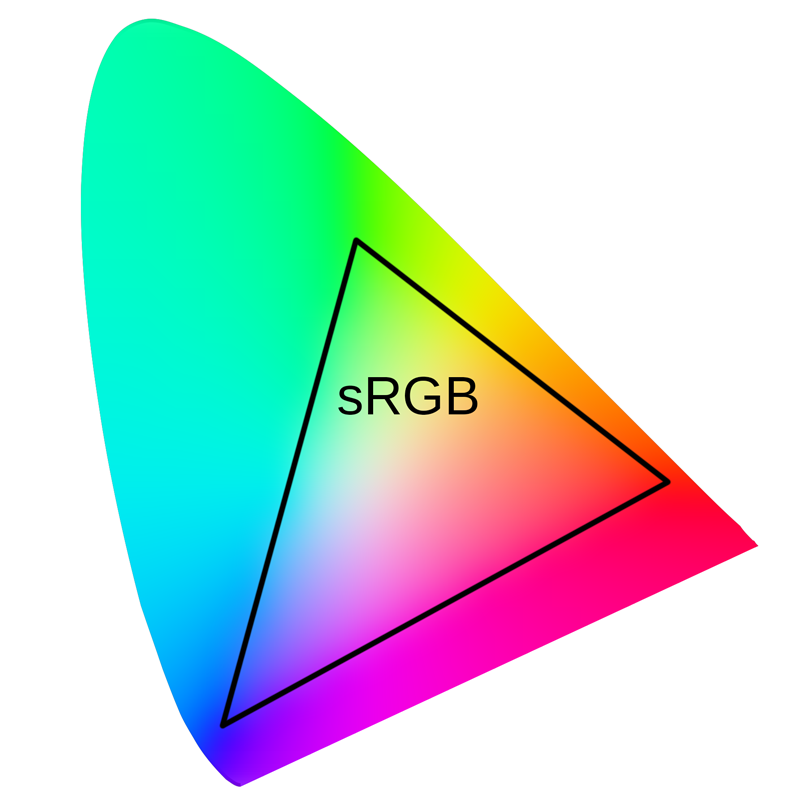


Рисунок 3- охват RGB цветовой модели

### Преимущества и ограничения

**Преимущества:**

* Простота реализации в цифровых устройствах.
* Интуитивно понятна для работы с цветами в аддитивной системе.
* Хорошо соответствует биологии человеческого зрения.

**Ограничения:**

* Ограниченное цветовое покрытие: RGB не может воспроизвести некоторые цвета, видимые человеческим глазом (например, яркие неоновые оттенки).
* Зависимость от устройства: один и тот же RGB-код может выглядеть по-разному на разных экранах из-за калибровки и качества дисплея.
* Не подходит для печати: для печати используется субтрактивная модель CMYK, так как RGB работает со светом, а не с пигментами.
  + 1. **CMYK**

CMYK (схема на рисунке 4) — это субтрактивная цветовая модель, используемая преимущественно в печати. Название модели происходит от английских названий четырех основных цветов: **Cyan (голубой), Magenta (пурпурный), Yellow (желтый)** и **Key/Black (черный)**. В отличие от аддитивной модели RGB, которая работает со светом и добавлением цветов, CMYK основана на вычитании света: цвета создаются за счет поглощения света пигментами или красками на поверхности, обычно бумаге.

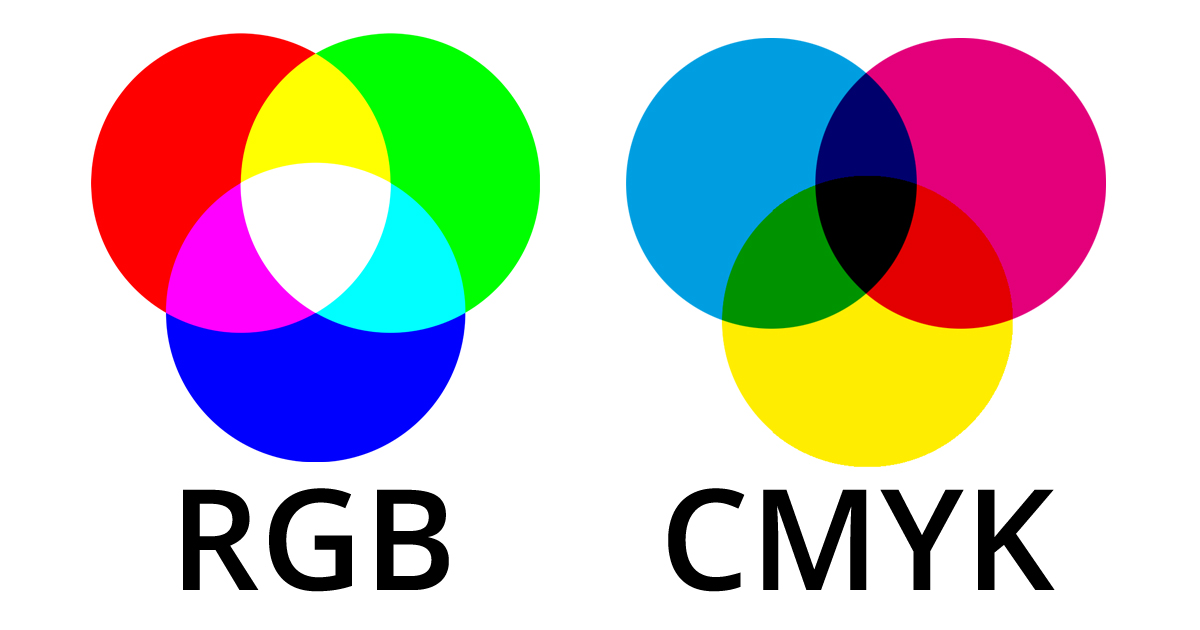


Рисунок 4 - CMYK схема

### Основные принципы CMYK

CMYK работает по принципу субтрактивного смешивания: чем больше краски вы добавляете, тем темнее становится результат, приближаясь к черному. Это противоположно RGB, где добавление света приводит к белому.

* **Голубой (C), пурпурный (M)** и **желтый (Y)** — это основные цвета, которые в теории могут комбинироваться для создания всех остальных оттенков в видимом спектре.
* **Черный (K)** добавлен как отдельный компонент, потому что смешение голубого, пурпурного и желтого в реальности не дает чистого черного цвета (а скорее грязно-коричневый) и требует больше краски, что увеличивает затраты и время высыхания.

Значения в CMYK обычно задаются в процентах (от 0% до 100%) для каждого канала:

* (0%, 0%, 0%, 0%) — отсутствие краски, белый цвет (цвет бумаги).
* (100%, 0%, 0%, 0%) — чистый голубой.
* (0%, 100%, 0%, 0%) — чистый пурпурный.
* (0%, 0%, 100%, 0%) — чистый желтый.
* (0%, 0%, 0%, 100%) — чистый черный.
* (100%, 100%, 100%, 0%) — теоретически черный, но на практике получается темно-серый или коричневатый.

### Как работает CMYK

В печати CMYK-цвета наносятся на бумагу в виде крошечных точек (растр), которые накладываются друг на друга или располагаются рядом. Человеческий глаз смешивает эти точки, создавая иллюзию сплошного цвета. Например:

* Голубой (C) + желтый (Y) = зеленый.
* Пурпурный (M) + желтый (Y) = красный.
* Голубой (C) + пурпурный (M) = синий.

Черный (K) используется для усиления контраста, создания глубоких теней и экономии цветных красок. Без черного канала темные цвета требовали бы полного покрытия бумаги тремя красками (C, M, Y), что не только дорого, но и может привести к проблемам с высыханием и деформацией бумаги.

### Почему "K" вместо "B"?

"B" (Black) не используется в аббревиатуре, чтобы избежать путаницы с синим (Blue) из RGB. "K" означает "Key" (ключевой), так как черный слой в печати часто выступает основой, выравнивающей и усиливающей остальные цвета.

### Цветовое пространство CMYK

CMYK определяет свое цветовое пространство, но оно значительно уже, чем у RGB. Это связано с физическими ограничениями красок:

* Пигменты в CMYK не могут воспроизвести некоторые яркие и насыщенные цвета, которые легко достигаются в RGB (например, яркий неон или чистый голубой экранного света).
* Реальный диапазон воспроизводимых цветов зависит от типа краски, бумаги и технологии печати.

### Применение CMYK

1. **Полиграфия**: Основная модель для печати журналов, книг, плакатов, упаковки и т.д.
2. **Дизайн для печати**: Программы вроде Adobe Illustrator и InDesign используют CMYK для подготовки макетов.
3. **Промышленное производство**: Используется в офсетной, цифровой и трафаретной печати.

### Преимущества и ограничения

**Преимущества:**

* Точное воспроизведение цветов на физических носителях.
* Экономичность благодаря использованию черного канала.
* Соответствие процессу печати, где цвета формируются пигментами.

**Ограничения:**

* Ограниченный цветовой охват (гамма): многие яркие RGB-цвета выходят за пределы возможностей CMYK.
* Зависимость от материалов: цвет может варьироваться в зависимости от типа бумаги (глянцевая, матовая) и качества краски.
* Сложность преобразования из RGB: при переводе из RGB в CMYK часто теряется яркость и насыщенность, что требует ручной корректировки дизайнерами.
  + 1. **HSL**

HSL — это цветовая модель, которая представляет цвета в терминах, более интуитивных для человеческого восприятия, чем RGB или CMYK. Аббревиатура расшифровывается как **Hue (оттенок), Saturation (насыщенность)** и **Lightness (светлота).** Эта модель была разработана для упрощения работы с цветами в дизайне, программировании и обработке изображений, где важно учитывать не только технические характеристики цвета, но и его визуальное восприятие.

### Основные принципы HSL

HSL описывает цвет в цилиндрической системе координат, где:

* **Hue (оттенок)** — это "какой цвет", определяемый углом на цветовом круге.
* **Saturation (насыщенность)** — это "насколько чистый цвет", то есть интенсивность цвета по сравнению с серым.
* **Lightness (светлота)** — это "насколько светлый или темный цвет", от черного до белого.

#### Hue (оттенок)

* Измеряется в градусах от 0° до 360° на цветовом круге.
* 0° (или 360°) — красный, 120° — зеленый, 240° — синий.
* Оттенок определяет базовый цвет без учета его яркости или насыщенности.

Примеры:

* 0° — красный.
* 60° — желтый.
* 180° — голубой (циан).

#### Saturation (насыщенность)

* Измеряется в процентах от 0% до 100%.
* 0% — полностью серый (отсутствие цвета).
* 100% — максимально насыщенный, "чистый" цвет.

Примеры:

* (0°, 100%, 50%) — яркий красный.
* (0°, 0%, 50%) — серый (нет насыщенности).

#### Lightness (светлота)

* Измеряется в процентах от 0% до 100%.
* 0% — черный (полное отсутствие света).
* 50% — "нормальная" светлота, где цвет проявляется наиболее полно.
* 100% — белый (полное присутствие света).

Примеры:

* (0°, 100%, 0%) — черный.
* (0°, 100%, 50%) — красный.
* (0°, 100%, 100%) — белый.

### Как работает HSL

HSL можно представить как цилиндр:

* **Ось цилиндра** — это светлота (от черного внизу до белого сверху).
* **Радиус** — это насыщенность (от серого в центре до чистого цвета на краю).
* **Угол** — это оттенок (полный круг 360°).

Эта модель позволяет легко изменять отдельные аспекты цвета. Например, если вы хотите сделать цвет светлее, вы увеличиваете Lightness, не затрагивая Hue или Saturation.

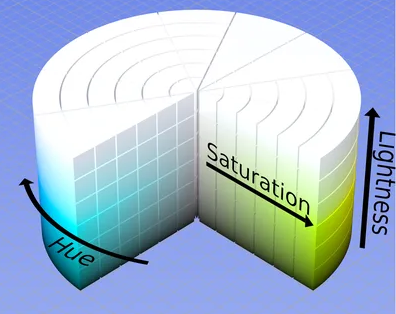


Рисунок 5- HSL модель

### Связь с RGB

HSL — это не независимая модель, а преобразование из RGB. Она основана на тех же трех основных цветах (красный, зеленый, синий), но перестраивает их в более удобный для восприятия формат. Например:

* RGB (255, 0, 0) = HSL (0°, 100%, 50%) — красный.
* RGB (127, 127, 127) = HSL (0°, 0%, 50%) — серый.
* RGB (255, 255, 0) = HSL (60°, 100%, 50%) — желтый.

Преобразование между RGB и HSL нелинейное и требует математических формул, которые учитывают максимальные и минимальные значения RGB-компонентов.

### Применение HSL

1. **Графический дизайн**: HSL удобен для выбора и настройки цветов в программах вроде Photoshop, Illustrator или Figma.
2. **Веб-разработка**: В CSS цвета можно задавать в формате HSL, например, hsl(120, 100%, 50%) для зеленого.
3. **Обработка изображений**: HSL используется для изменения яркости, контраста или цветовых эффектов, так как разделяет оттенок, насыщенность и светлоту.
4. **UI/UX**: Дизайнеры используют HSL для создания гармоничных палитр, изменяя только один параметр (например, оттенок для оттенков одного цвета).

### Преимущества и ограничения

**Преимущества:**

* Интуитивность: Hue, Saturation и Lightness ближе к тому, как люди описывают цвета в реальной жизни.
* Гибкость: Легко менять яркость или насыщенность, не затрагивая базовый оттенок.
* Удобство для создания палитр: Например, можно взять один Hue и варьировать Saturation и Lightness для оттенков.

**Ограничения:**

* Зависимость от RGB: HSL не расширяет цветовое пространство за пределы RGB и наследует его ограничения.
* Неравномерность восприятия: Некоторые изменения в HSL (например, в Lightness) могут выглядеть нелинейно для человеческого глаза по сравнению с более точными моделями, такими как Lab.
* Не подходит для печати: Как и RGB, HSL ориентирован на цифровые устройства, а не на пигменты.
  + 1. **HSV**

HSV — это цветовая модель, которая, как и HSL, ориентирована на более интуитивное представление цвета для человека по сравнению с RGB. Аббревиатура расшифровывается как **Hue (оттенок), Saturation (насыщенность) и Value (значение)**, иногда называемая "яркость". HSV также известна как HSB (Hue, Saturation, Brightness), где "B" — это синоним "Value". Эта модель часто используется в компьютерной графике, дизайне и обработке изображений благодаря своей простоте и удобству для манипуляций с цветом.

### Основные принципы HSV

HSV тоже использует цилиндрическую систему координат, но с немного другим подходом к интерпретации яркости/значения по сравнению с HSL:

* **Hue (оттенок)** — определяет базовый цвет на цветовом круге.
* **Saturation (насыщенность)** — указывает чистоту цвета, то есть насколько он отличается от серого.
* **Value (значение)** — определяет яркость цвета, от черного до максимальной интенсивности цвета.

#### Hue (оттенок)

* Как и в HSL, измеряется в градусах от 0° до 360°.
* 0° — красный, 120° — зеленый, 240° — синий.
* Оттенок задает "какой цвет" без учета его яркости или насыщенности.

Примеры:

* 0° — красный.
* 90° — желто-зеленый.
* 180° — голубой (циан).

#### Saturation (насыщенность)

* Измеряется в процентах от 0% до 100%.
* 0% — полностью серый (или черный/белый, в зависимости от Value).
* 100% — максимально насыщенный, чистый цвет.

Примеры:

* (0°, 100%, 100%) — яркий красный.
* (0°, 0%, 100%) — белый (при максимальном Value).

#### Value (значение)

* Измеряется в процентах от 0% до 100%.
* 0% — черный (полное отсутствие яркости).
* 100% — максимальная яркость цвета, заданного Hue и Saturation.

Примеры:

* (0°, 100%, 0%) — черный.
* (0°, 100%, 50%) — средний красный.
* (0°, 100%, 100%) — яркий красный.

### Как работает HSV

HSV можно представить как конус или цилиндр:

* **Ось** — это Value, где основание (V=100%) — максимальная яркость, а вершина (V=0%) — черный.
* **Радиус** — это Saturation, где центр (S=0%) — серый/белый/черный, а край (S=100%) — чистый цвет.
* **Угол** — это Hue, полный круг в 360°.

В отличие от HSL, где светлота (Lightness) идет от черного через нормальный цвет к белому, в HSV Value отвечает только за интенсивность цвета, а белый достигается при снижении насыщенности (Saturation) при максимальном Value.

### Применение HSV

1. **Графический дизайн**: Удобен для выбора цветов и создания эффектов в программах вроде Photoshop или GIMP.
2. **Обработка изображений**: Часто используется в компьютерном зрении (например, OpenCV) для выделения объектов по цвету, так как Hue позволяет легко отделить цвет от яркости.
3. **Игровая графика**: HSV помогает быстро менять яркость или насыщенность объектов без изменения их базового оттенка.
4. **UI/UX**: Используется для создания динамических палитр или анимаций цвета.

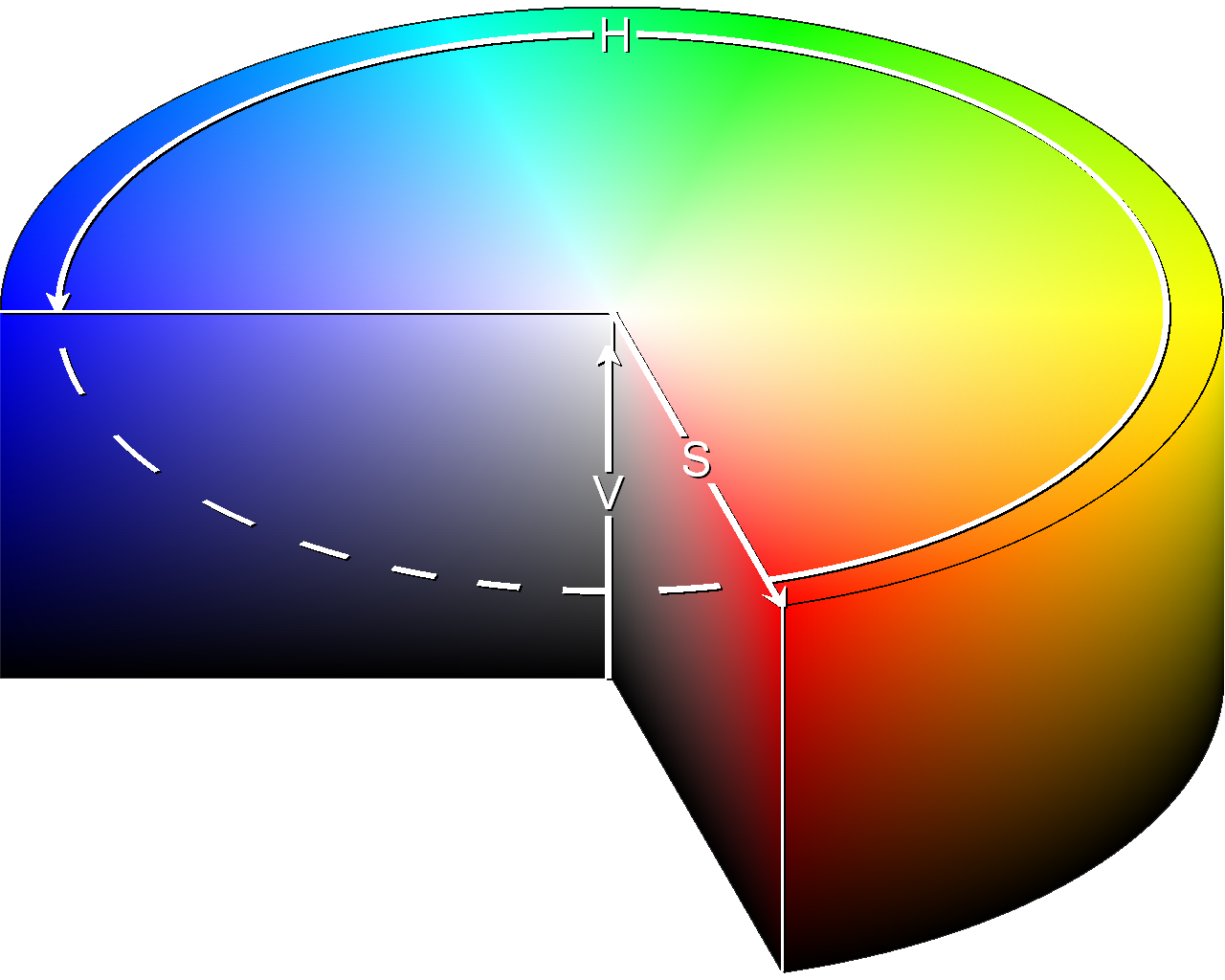


Рисунок 6 - HSV модель

### Преимущества и ограничения

**Преимущества:**

* Интуитивность: Hue, Saturation и Value понятны на уровне восприятия.
* Удобство для анализа: Легко отделить цвет (Hue) от яркости (Value), что полезно в обработке изображений.
* Простота манипуляций: Например, изменение Value позволяет затемнить или осветлить цвет без искажения его характера.

**Ограничения:**

* Зависимость от RGB: HSV не выходит за пределы цветового пространства RGB.
* Неравномерность: Как и HSL, не полностью соответствует восприятию человеческого глаза (в отличие от Lab).
* Ограниченность в печати: HSV ориентирован на цифровые устройства, а не на физические пигменты.
  + 1. **LAB**

CIE Lab (или просто Lab) — это цветовая модель, разработанная Международной комиссией по освещению (CIE) в 1976 году. Она отличается от RGB, CMYK, HSL и HSV тем, что стремится быть **перцептивно равномерной**, то есть изменения в значениях цвета в этой модели более точно соответствуют тому, как человеческий глаз воспринимает различия между цветами. Lab не зависит от устройств (в отличие от RGB или CMYK) и охватывает весь видимый спектр, что делает ее универсальной для научных, промышленных и дизайнерских задач.

### Основные принципы Lab

Lab состоит из трех компонентов:

* L (Lightness)\* — светлота, от черного до белого.
* **a**\* — ось от зеленого к красному (или пурпурному).
* **b**\* — ось от синего к желтому.

#### L\* (светлота)

* Диапазон: от 0 (черный) до 100 (белый).
* Это единственный параметр, который отвечает за яркость, независимо от цвета.

Примеры:

* L\*=0 — черный.
* L\*=50 — средняя светлота.
* L\*=100 — белый.

#### a\* (зеленый-красный)

* Диапазон: примерно от -128 до +127 (в некоторых системах может варьироваться).
* Отрицательные значения — зеленый.
* Положительные значения — красный (или пурпурный).
* a\*=0 — нейтральный (без смещения по этой оси).

Примеры:

* a\*=-50 — насыщенный зеленый (при b\*=0).
* a\*=+50 — насыщенный красный (при b\*=0).

#### b\* (синий-желтый)

* Диапазон: примерно от -128 до +127.
* Отрицательные значения — синий.
* Положительные значения — желтый.
* b\*=0 — нейтральный (без смещения по этой оси).

Примеры:

* b\*=-50 — насыщенный синий (при a\*=0).
* b\*=+50 — насыщенный желтый (при a\*=0).

### Как работает Lab

Lab основана на физиологии человеческого зрения и модели CIE XYZ, которая описывает цвет как комбинацию трех стимулов (X, Y, Z), соответствующих восприятию колбочек в глазу. Lab преобразует эти данные в более удобную форму:

* **L**\* соответствует яркости (Y в XYZ).
* **a**\* и **b**\* представляют цветовые оппонентные оси, основанные на теории цветового восприятия, где зеленый противопоставлен красному, а синий — желтому.

Цвет в Lab задается точкой в трехмерном пространстве (представлено на рисунке 7), где:

* Расстояние между двумя точками (дельта E, ΔE) количественно определяет разницу в цвете, видимую человеком.
* Модель охватывает все цвета, которые может видеть глаз, включая те, что недоступны в RGB или CMYK.

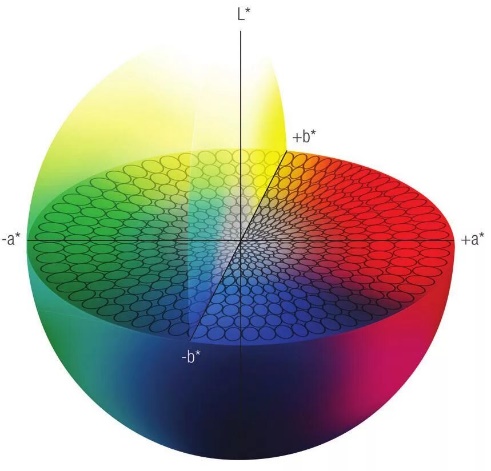


Рисунок 7 - трехмерная модель цветового пространства LAB

### Связь с другими моделями

Lab не зависит от RGB или CMYK напрямую, но может быть преобразована в них через промежуточную модель CIE XYZ. Например:

* RGB (255, 0, 0) ≈ Lab (53, 80, 67) — красный.
* RGB (0, 255, 0) ≈ Lab (88, -86, 83) — зеленый.
* RGB (0, 0, 255) ≈ Lab (32, 79, -108) — синий.

Точные значения зависят от цветового профиля (например, sRGB) и условий освещения.

### Применение Lab

1. **Цветокоррекция**: Используется в Photoshop и других программах для точной настройки цвета, так как разделяет светлоту и цветовые компоненты.
2. **Промышленность**: Применяется в производстве красок, текстиля и пластмасс для контроля качества цвета.
3. **Наука**: Используется в исследованиях восприятия цвета и цветовых различий (например, ΔE).
4. **Калибровка устройств**: Служит промежуточной моделью для преобразования между RGB и CMYK с учетом освещения.

### Преимущества и ограничения

**Преимущества:**

* **Широкий охват:** Включает все видимые цвета, в отличие от RGB или CMYK.
* **Перцептивная равномерность**: Различия в значениях Lab лучше соответствуют тому, как люди видят разницу между цветами.
* **Независимость от устройств:** Не привязана к конкретному дисплею или принтеру, что делает ее универсальной.

**Ограничения:**

* Сложность: Менее интуитивна для человека по сравнению с HSL или HSV.
* Вычислительная нагрузка: Преобразования в Lab и из нее требуют сложных формул.
* Не для прямого использования: Обычно служит промежуточной моделью, а не конечным форматом для экранов или печати.
  + 1. **YCbCr**

YCbCr — это цветовая модель, широко используемая в цифровой обработке видео и изображений, особенно в системах сжатия, таких как JPEG, MPEG и телевизионных стандартах (например, PAL, NTSC). Она разделяет информацию о яркости и цвете, что позволяет эффективно сжимать данные, учитывая особенности человеческого восприятия. Аббревиатура расшифровывается как Y (яркость), Cb (синяя хроминантность) и Cr (красная хроминантность).

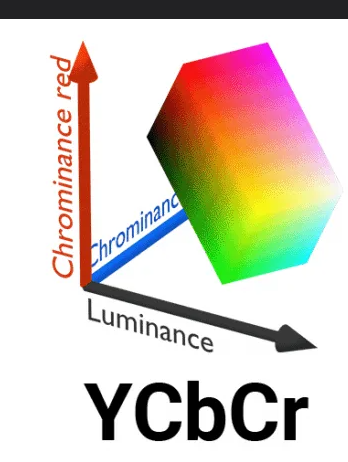


Рисунок 8 - цветовая модель YCbCr

**Основные принципы YCbCr**

YCbCr основана на разделении сигнала на:

* **Y (Luma)** — яркостный компонент, который отвечает за интенсивность света (монохромное изображение).
* **Cb** — синяя цветовая разность (blue chroma), показывающая отклонение цвета от серого в сторону синего.
* **Cr** — красная цветовая разность (red chroma), показывающая отклонение цвета от серого в сторону красного.

Зеленый цвет не представлен отдельно, так как он вычисляется из Y, Cb и Cr, что связано с высокой чувствительностью человеческого глаза к зеленому в яркости.

**Y (яркость)**

* Диапазон: обычно 0–255 (в 8-битной системе).
* 0 — черный, 255 — белый.
* Y содержит только информацию о светлоте, без цветовых данных.

**Cb (синяя хроминантность)**

* Диапазон: 0–255, но центр (нейтральный серый) — 128.
* Меньше 128 — смещение к желтому.
* Больше 128 — смещение к синему.

**Cr (красная хроминантность)**

* Диапазон: 0–255, центр — 128.
* Меньше 128 — смещение к голубому (циану).
* Больше 128 — смещение к красному.

Пример:

* Y=0, Cb=128, Cr=128 — черный (нет яркости, нейтральный цвет).
* Y=255, Cb=128, Cr=128 — белый (максимальная яркость, нейтральный цвет).
* Y=128, Cb=255, Cr=128 — синий (средняя яркость, сильное смещение к синему).

**Как работает YCbCr**

YCbCr — это преобразование из RGB, где:

* **Y** вычисляется как взвешенная сумма R, G и B, учитывающая чувствительность глаза (зеленый вносит больший вклад, чем красный, а синий — наименьший).
* **Cb** = B - Y (разница между синим и яркостью).
* **Cr** = R - Y (разница между красным и яркостью).

Формула преобразования (для ITU-R BT.601, стандартного телевидения):

* Y = 0.299R + 0.587G + 0.114B
* Cb = 0.564(B - Y) + 128
* Cr = 0.713(R - Y) + 128

Обратное преобразование позволяет восстановить RGB из YCbCr.

**Почему разделение яркости и цвета?**

Человеческий глаз более чувствителен к изменениям яркости (Y), чем к изменениям цвета (Cb, Cr). Это используется в сжатии:

* **Субдискретизация**: Цветовые компоненты (Cb и Cr) часто уменьшаются в разрешении (например, 4:2:2 или 4:2:0), тогда как Y сохраняет полное разрешение. Это снижает объем данных без заметной потери качества.

Примеры субдискретизации:

* **4:4:4** — полное разрешение для Y, Cb, Cr.
* **4:2:2** — Cb и Cr имеют половину горизонтального разрешения.
* **4:2:0** — Cb и Cr имеют половину разрешения по горизонтали и вертикали (стандарт JPEG/MPEG).

**Связь с RGB**

YCbCr напрямую зависит от RGB и является его производной. Например:

* RGB (255, 0, 0) ≈ YCbCr (76, 85, 255) — красный.
* RGB (0, 255, 0) ≈ YCbCr (150, 64, 21) — зеленый.
* RGB (0, 0, 255) ≈ YCbCr (29, 255, 107) — синий.

Точные значения зависят от стандарта (BT.601, BT.709 и т.д.), который определяет коэффициенты преобразования.

**Применение YCbCr**

1. Видео и телевидение: Используется в цифровых видеостандартах (MPEG, H.264) и аналоговых системах (NTSC, PAL).
2. Сжатие изображений: JPEG сохраняет данные в YCbCr, применяя субдискретизацию для уменьшения размера файла.
3. Обработка сигналов: Удобна для фильтрации или коррекции яркости/цвета отдельно.
4. Трансляция: YCbCr лежит в основе компонентного видео (YUV), где Y — яркость, а U и V — аналоги Cb и Cr.

**Преимущества и ограничения**

**Преимущества:**

* Эффективность сжатия: Разделение яркости и цвета позволяет экономить пропускную способность или память.
* Соответствие восприятию: Учитывает, что глаз лучше различает яркость, чем мелкие цветовые детали.
* Универсальность: Используется как в цифровых, так и в аналоговых системах.

**Ограничения:**

* Зависимость от RGB: Не расширяет цветовое пространство за пределы RGB.
* Сложность преобразования: Требует вычислений для перехода между RGB и YCbCr.
* Ограниченная интуитивность: Менее понятна для дизайна по сравнению с HSL/HSV.
  1. **Формулы перевода между цветовыми моделями:**

**1.3.1 Формулы перевода из RGB**

* 1. **Формулы перевода из RGB в CMYK:**
* C = 1 – (R/255)
* M = 1 – (G/255)
* Y = 1 – (B/255)
* K= min (C,M,Y)
  1. **Формулы перевода из RGB в HSL:**

Определение максимального и минимального значения среди R, G, B:

*R* '= *R* / 255

*G* '= *G* / 255

*B* '= *B* / 255

*Cmax* = max ( *R* ', *G* ', *B* ')

*Cmin* = min ( *R* ', *G* ', *B* ')

Δ = *Cmax* - *Cmin*

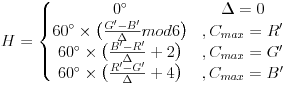
Вычисление яркости **L**:

*L* = ( *Cmax + Cmin* ) / 2

Вычисление насыщенности **S**:



Вычисление оттенка **H** (зависит от разницы между max и min значениями):



* 1. Формулы перевода из RGB в HSV

Определение максимального и минимального значения среди R, G, B:

*R* '= *R* / 255

*G* '= *G* / 255

*B* '= *B* / 255

*Cmax* = max ( *R* ', *G* ', *B* ')

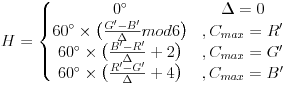
*Cmin* = min ( *R* ', *G* ', *B* ')

Δ = *Cmax* - *Cmin*

Вычисление насыщенности **S**:

**Если Cmax = 0, то S = 0, иначе**

Вычисление оттенка **H** (зависит от разницы между max и min значениями):



Значение V:

V = Cmax

* 1. Формулы перевода из RGB в LAB:

L (Lightness) – яркость;

A – компонент красно-зеленой оси;

B – компонент сине-жёлтой оси.

Перевод RGB в LAB требует промежуточного преобразования через XYZ:

*R* '= *R* / 255

*G* '= *G* / 255

*B* '= *B* / 255

Где C -Это R,G,B.

*X*=0.4124564⋅ Rlinear +0.3575761⋅ Glinear +0.1804375⋅ Blinear   
*Y*=0.2126729⋅ Rlinear +0.7151522⋅ Glinear +0.0721750⋅ Blinear   
*Z*=0.0193339⋅ Rlinear +0.1191920⋅ Glinear +0.9503041⋅ Blinear

Затем используются нелинейные преобразования для получения L, A и B.

* 1. Формулы перевода из RGB в YCbCr

Y=0.299R+0.587G+0.114B

Cb=128−0.168736R−0.331264G+0.5B

Cr=128+0.5R−0.418688G−0.081312B

**1.3.2 Формулы перевода из CMYK**

2 Преобразования из CMYK

1. Формулы перевода из CMYK в RGB

* R=255(1−C)(1−K),
* G=255(1−M)(1−K),
* B=255(1−Y)(1−K)

1. Формулы перевода из CMYK в HSL

Cmax = max (((1-C)(1-K)), ((1-M)(1-K)),((1-C)(1-K)))

Cmin = min (((1-C)(1-K)), ((1-M)(1-K)),((1-C)(1-K)))

= Cmax – Cmin = max (((1-C)(1-K)), ((1-M)(1-K)),((1-C)(1-K))) - min (((1-C)(1-K)), ((1-M)(1-K)),((1-C)(1-K)))

Яркость:

Насыщенность:

Если

max (((1-C)(1-K)), ((1-M)(1-K)),((1-C)(1-K))) - min (((1-C)(1-K)), ((1-M)(1-K)),((1-C)(1-K))) = 0, то H=S=0, иначе:

Оттенок:

Если , то H = 0, иначе:

1. Формулы преобразования из CMYK в HSV

Cmax = max (((1-C)(1-K)), ((1-M)(1-K)),((1-C)(1-K)))

Cmin = min (((1-C)(1-K)), ((1-M)(1-K)),((1-C)(1-K)))

= Cmax – Cmin = max (((1-C)(1-K)), ((1-M)(1-K)),((1-C)(1-K))) - min (((1-C)(1-K)), ((1-M)(1-K)),((1-C)(1-K)))

V = Cmax = max (((1-C)(1-K)), ((1-M)(1-K)),((1-C)(1-K)))

Если Cmax = 0, то

Елси , то H = 0, иначе:

1. Формулы переводов из CMYK в LAB

X = 0.4124 \* + 0.3576 \* + 0.1805 \*

Y = 0.2126 \* + 0.7152 \* + 0.0722 \*

Z = 0.0193 \* + 0.1192 \* + 0.9505 \*

**L = 116 \* Fy - 16**

**A = 500 \* (Fx - Fy)**

**B = 200 \* (Fy - Fz)**

1. Формулы перевода из CMYK в YCbCr

Y=0.299\*255(1-C)(1-K)+0.587\*255(1-M)(1-K)+0.114\*255(1-Y)(1-K) = 76.245(1-C)(1-K)+149.685(1-M)(1-K)+29.07(1-Y)(1-K) = **255 - 255K - 76.245C - 149.685M - 29.07Y + 76.245CK + 149.685MK + 29.07YK**

Cb=128−0.168736\*255(1-C)(1-K)−0.331264\*255(1-M)(1-K)+0.5\*255(1-Y)(1-K) = **128 + 43.02768C + 84.47232M − 127.5Y − 43.02768CK − 84.47232MK + 127.5YK**

Cr=128+0.5\*255(1-C)(1-K)−0.418688\*255(1-M)(1-K)−0.081312\*255(1-Y)(1-K) = **128 − 127.5C + 106.76544M + 20.73456Y + 127.5CK − 106.76544MK − 20.73456YK**

**1.3.3 Формулы перевода из HSL**

1. Формулы перевода из HSL в RGB

Когда 0 ≤ H <360, 0 ≤ S ≤ 1 и 0 ≤ L ≤ 1:

С = (1 - | 2L - 1 |) × S

X = C × (1 - | ( H / 60 °) mod 2 - 1 |)

м = L - C / 2

R, G, B =

(C + m, X + m, m) если 0 ≤ H < 60

(X + m, С+ m, m) если 60 ≤ H < 120

(m, С+ m, X + m) если 120 ≤ H < 180

(m, X + m, С+ m) если 180 ≤ H < 240

(X + m, m, С+ m) если 240 ≤ H < 300

(С+ m, m, X + m) если 300 ≤ H < 360

1. Формулы перевода из HSL в CMYK

Когда 0 ≤ H <360, 0 ≤ S ≤ 1 и 0 ≤ L ≤ 1:

С = (1 - | 2L - 1 |) × S

X = C × (1 - | ( H / 60 °) mod 2 - 1 |)

м = L - C / 2

R, G, B =

(C + m, X + m, m) если 0 ≤ H < 60

(X + m, С+ m, m) если 60 ≤ H < 120

(m, С+ m, X + m) если 120 ≤ H < 180

(m, X + m, С+ m) если 180 ≤ H < 240

(X + m, m, С+ m) если 240 ≤ H < 300

(С+ m, m, X + m) если 300 ≤ H < 360

***K = 1 - max(R, G, B)***

***C = (1 - R - K) / (1 - K) если K < 1, иначе C = 0***

***M = (1 - G - K) / (1 - K) если K < 1, иначе M = 0***

***Y = (1 - B - K) / (1 - K) если K < 1, иначе Y = 0***

1. **Формулы перевода из HSL в HSV**

**H\_HSV = H\_HSL**

Chroma = S\_HSL \* min(L, 1-L)

**V = L + Chroma / 2**

**S\_HSV = Chroma / V если V > 0**

**S\_HSV = 0 если V = 0**

1. **Формулы перевода из HSL в LAB**

Когда 0 ≤ H <360, 0 ≤ S ≤ 1 и 0 ≤ L ≤ 1:

С = (1 - | 2L - 1 |) × S

X = C × (1 - | ( H / 60 °) mod 2 - 1 |)

м = L - C / 2

R, G, B =

(C + m, X + m, m) если 0 ≤ H < 60

(X + m, С+ m, m) если 60 ≤ H < 120

(m, С+ m, X + m) если 120 ≤ H < 180

(m, X + m, С+ m) если 180 ≤ H < 240

(X + m, m, С+ m) если 240 ≤ H < 300

(С+ m, m, X + m) если 300 ≤ H < 360

**L = 116 \* Fy - 16**

**A = 500 \* (Fx - Fy)**

**B = 200 \* (Fy - Fz)**

1. **Формулы перевода из HSL в YCbCr**

**Chroma = S \* min(L, 1-L)**

**Chroma = S \* min(L, 1-L)**

**H' = H / 360**

**Y = L + Chroma \* (0.299 \* cos(2π \* H') + 0.587 \* cos(2π \* (H' - 1/3)) + 0.114 \* cos(2π \* (H' - 2/3)))**

**Cb = 0.5 + Chroma \* (-0.168736 \* cos(2π \* H') - 0.331264 \* cos(2π \* (H' - 1/3)) + 0.5 \* cos(2π \* (H' - 2/3)))**

**Cr = 0.5 + Chroma \* (0.5 \* cos(2π \* H') - 0.418688 \* cos(2π \* (H' - 1/3)) - 0.081312 \* cos(2π \* (H' - 2/3)))**

**1.3.4 Формулы перевода из HSV:**

1. **Формулы перевода из HSV в RGB:**

Когда 0 ≤ H <360, 0 ≤ S ≤ 1 и 0 ≤ V ≤ 1:

С = (1 - | 2V - 1 |) × S

X = C × (1 - | ( H / 60 °) mod 2 - 1 |)

м = V - C / 2

R, G, B =

(C + m, X + m, m) если 0 ≤ H < 60

(X + m, С+ m, m) если 60 ≤ H < 120

(m, С+ m, X + m) если 120 ≤ H < 180

(m, X + m, С+ m) если 180 ≤ H < 240

(X + m, m, С+ m) если 240 ≤ H < 300

(С+ m, m, X + m) если 300 ≤ H < 360

1. **Формулы перевода из HSV в CMYK**

Когда 0 ≤ H <360, 0 ≤ S ≤ 1 и 0 ≤ V ≤ 1:

С = (1 - | 2V - 1 |) × S

X = C × (1 - | ( H / 60 °) mod 2 - 1 |)

м = V - C / 2

R, G, B =

(C + m, X + m, m) если 0 ≤ H < 60

(X + m, С+ m, m) если 60 ≤ H < 120

(m, С+ m, X + m) если 120 ≤ H < 180

(m, X + m, С+ m) если 180 ≤ H < 240

(X + m, m, С+ m) если 240 ≤ H < 300

(С+ m, m, X + m) если 300 ≤ H < 360

***K = 1 - max(R, G, B)***

***C = (1 - R - K) / (1 - K) если K < 1, иначе C = 0***

***M = (1 - G - K) / (1 - K) если K < 1, иначе M = 0***

***Y = (1 - B - K) / (1 - K) если K < 1, иначе Y = 0***

1. **Формулы перевода из HSV в HSL**

**H' = H**

**S' = S**

**V' = V**

1. **Формулы перевода из HSV в LAB**

Когда 0 ≤ H <360, 0 ≤ S ≤ 1 и 0 ≤ V ≤ 1:

С = (1 - | 2V - 1 |) × S

X = C × (1 - | ( H / 60 °) mod 2 - 1 |)

м = V - C / 2

R, G, B =

(C + m, X + m, m) если 0 ≤ H < 60

(X + m, С+ m, m) если 60 ≤ H < 120

(m, С+ m, X + m) если 120 ≤ H < 180

(m, X + m, С+ m) если 180 ≤ H < 240

(X + m, m, С+ m) если 240 ≤ H < 300

(С+ m, m, X + m) если 300 ≤ H < 360

**X = 0.4124 \* R + 0.3576 \* G + 0.1805 \* B**

**Y = 0.2126 \* R + 0.7152 \* G + 0.0722 \* B**

**Z = 0.0193 \* R + 0.1192 \* G + 0.9505 \* B**

**L = 116 \* Fy - 16**

**A = 500 \* (Fx - Fy)**

**B = 200 \* (Fy - Fz)**

1. **Формулы перевода из HSV в YCbCr**

Когда 0 ≤ H <360, 0 ≤ S ≤ 1 и 0 ≤ V ≤ 1:

С = (1 - | 2V - 1 |) × S

X = C × (1 - | ( H / 60 °) mod 2 - 1 |)

м = V - C / 2

R, G, B =

(C + m, X + m, m) если 0 ≤ H < 60

(X + m, С+ m, m) если 60 ≤ H < 120

(m, С+ m, X + m) если 120 ≤ H < 180

(m, X + m, С+ m) если 180 ≤ H < 240

(X + m, m, С+ m) если 240 ≤ H < 300

(С+ m, m, X + m) если 300 ≤ H < 360

**Y = 0.299 \* R + 0.587 \* G + 0.114 \* B**

**Cb = 0.5 - 0.168736 \* R - 0.331264 \* G + 0.5 \* B**

**Cr = 0.5 + 0.5 \* R - 0.418688 \* G - 0.081312 \* B**

**1.3.5 Формулы перевода из LAB**

1. **Формулы перевода из LAB в RGB**

fy = (L + 16) / 116

fx = fy + A / 500

fz = fy - B / 200

1. Формулы перевода из LAB в CMYK

fy = (L + 16) / 116

fx = fy + A / 500

fz = fy - B / 200

1. Формулы перевода из LAB в HSL

fy = (L + 16) / 116

fx = fy + A / 500

fz = fy - B / 200

Cmin = min ()

Cmax=max ()

**Δ = Cmax – Cmin =** max ()- min()

Если

Елси , то H = 0, иначе:

**1.3.4 Формулы перевода из LAB в HSV**

fy = (L + 16) / 116

fx = fy + A / 500

fz = fy - B / 200

Cmin = min ()

Cmax=max ()

**Δ = Cmax – Cmin =** max ()- min()

V = Cmax = max

()

S = 0, если , иначе:

Елси , то H = 0, иначе:

1. Формулы перевода из LAB в YCbCr

fy = (L + 16) / 116

fx = fy + A / 500

fz = fy - B / 200

Y = 0.299 \* + 0.587 \* + 0.114 \*

Cb = 0.5 - 0.168736 \* - 0.331264 \* + 0.5 \*

Cr = 0.5 + 0.5 \* - 0.418688 \* - 0.081312 \*

**1.3.6 Формулы перевода из YCbCr**

1. **Формулы перевода из YCbCr в RGB**

**R = Y + 1.402 \* (Cr - 0.5)**

**G = Y - 0.344136 \* (Cb - 0.5) - 0.714136 \* (Cr - 0.5)**

**B = Y + 1.772 \* (Cb - 0.5)**

1. **Формулы перевода из YCbCr в CMYK**

**R = Y + 1.402 \* (Cr - 0.5)**

**G = Y - 0.344136 \* (Cb - 0.5) - 0.714136 \* (Cr - 0.5)**

**B = Y + 1.772 \* (Cb - 0.5)**

K = 1 - max(R, G, B) = max (((1 – (Y + 1.402 \* (Cr - 0.5))- max (Y + 1.402 \* (Cr - 0.5), Y - 0.344136 \* (Cb - 0.5)) – (0.714136 \* (Cr - 0.5), Y + 1.772 \* (Cb - 0.5)))

C = (1 - R - K) / (1 - K) = (((1 – (Y + 1.402 \* (Cr - 0.5))- max (Y + 1.402 \* (Cr - 0.5), Y - 0.344136 \* (Cb - 0.5)) – (0.714136 \* (Cr - 0.5), Y + 1.772 \* (Cb - 0.5)))

если K < 1,

иначе C = 0

M = (1 - G - K) / (1 - K) = (1-G-K)/ (((1 – (Y + 1.402 \* (Cr - 0.5))- max (Y + 1.402 \* (Cr - 0.5), Y - 0.344136 \* (Cb - 0.5)) – (0.714136 \* (Cr - 0.5), Y + 1.772 \* (Cb - 0.5))) если K < 1,

иначе M = 0

Y = (1 - B - K) / (1 - K) = (1-B-K)/ (((1 – (Y + 1.402 \* (Cr - 0.5))- max (Y + 1.402 \* (Cr - 0.5), Y - 0.344136 \* (Cb - 0.5)) – (0.714136 \* (Cr - 0.5), Y + 1.772 \* (Cb - 0.5)))если K < 1,

иначе Y = 0

1. **Формулы перевода из YCbCr в HSL**

**R =** **Y + 1.402 \* (Cr - 0.5)**

**G = Y - 0.344136 \* (Cb - 0.5) - 0.714136 \* (Cr - 0.5)**

**B =** **Y + 1.772 \* (Cb - 0.5)**

Cmax = max (R,G,B) = max ( **Y + 1.402 \* (Cr - 0.5) , Y - 0.344136 \* (Cb - 0.5) - 0.714136 \* (Cr - 0.5), Y + 1.772 \* (Cb - 0.5))**

Cmin = min (R,G,B) = min ( ( **Y + 1.402 \* (Cr - 0.5) , Y - 0.344136 \* (Cb - 0.5) - 0.714136 \* (Cr - 0.5), Y + 1.772 \* (Cb - 0.5))**

**Δ = Сmax – Cmin =** max ( **Y + 1.402 \* (Cr - 0.5) , Y - 0.344136 \* (Cb - 0.5) - 0.714136 \* (Cr - 0.5), Y + 1.772 \* (Cb - 0.5)) -** max ( **Y + 1.402 \* (Cr - 0.5) , Y - 0.344136 \* (Cb - 0.5) - 0.714136 \* (Cr - 0.5), Y + 1.772 \* (Cb - 0.5))**

**Яркость:**

**Насыщенность:**

Если

= 0, то H=S=0, иначе:

**Оттенок:**

Елси , то H = 0, иначе:

1. **Формулы перевода из YCbCr в HSV**

**R = Y + 1.402 \* (Cr - 0.5)**

**G = Y - 0.344136 \* (Cb - 0.5) - 0.714136 \* (Cr - 0.5)**

**B = Y + 1.772 \* (Cb - 0.5)**

Cmax = max (R,G,B) = max ( **Y + 1.402 \* (Cr - 0.5) , Y - 0.344136 \* (Cb - 0.5) - 0.714136 \* (Cr - 0.5), Y + 1.772 \* (Cb - 0.5))**

Cmin = min (R,G,B) = min ( ( **Y + 1.402 \* (Cr - 0.5) , Y - 0.344136 \* (Cb - 0.5) - 0.714136 \* (Cr - 0.5), Y + 1.772 \* (Cb - 0.5))**

**Δ = Сmax – Cmin =** max ( **Y + 1.402 \* (Cr - 0.5) , Y - 0.344136 \* (Cb - 0.5) - 0.714136 \* (Cr - 0.5), Y + 1.772 \* (Cb - 0.5)) -** max ( **Y + 1.402 \* (Cr - 0.5) , Y - 0.344136 \* (Cb - 0.5) - 0.714136 \* (Cr - 0.5), Y + 1.772 \* (Cb - 0.5))**

**Оттенок:**

Елси , то H = 0, иначе:

1. **Формулы перевода из YCbCr в LAB**

**R = Y + 1.402 \* (Cr - 0.5)**

**G = Y - 0.344136 \* (Cb - 0.5) - 0.714136 \* (Cr - 0.5)**

**B = Y + 1.772 \* (Cb - 0.5)**

**Перевод в XYZ:**

X ***=*** 0.4124 \* ***R*** + 0.3576 \* ***G*** + 0.1805 \* ***B*** ***=*** 0.4124 \* Y + 1.402 \* (Cr - 0.5)+ 0.3576 \* Y - 0.344136 \* (Cb - 0.5) - 0.714136 \* (Cr - 0.5)+ 0.1805 \* Y + 1.772 \* (Cb - 0.5) ***=*** **0.9505⋅Y+1.427864⋅Cb+0.687864⋅Cr−1.057864**

Y ***=*** 0.2126 \* ***R*** + 0.7152 \* ***G*** + 0.0722 \* ***B*** ***=*** 0.2126 \* Y + 1.402 \* (Cr - 0.5)+ 0.7152 \* Y - 0.344136 \* (Cb - 0.5) - 0.714136 \* (Cr - 0.5)+ 0.0722 \* Y + 1.772 \* (Cb - 0.5) **= Y+1.427864⋅Cb+0.687864⋅Cr−1.057864**

Z **=** 0.0193 \* ***R*** + 0.1192 \* ***G*** + 0.9505 \* ***B*** **=** 0.0193 \* Y + 1.402 \* (Cr - 0.5)+ 0.1192 \* Y - 0.344136 \* (Cb - 0.5) - 0.714136 \* (Cr - 0.5)+ 0.9505 \* Y + 1.772 \* (Cb - 0.5) **= 1.089⋅Y+1.427864⋅Cb+0.687864⋅Cr−1.057864**

**Перевод в LAB**

**2 Практическая часть**

* 1. **Примеры работы ПО**

В ходе практической части было разработано программное обеспечение с графическим интерфейсом (GUI) для анализа изображений и преобразования цветов между различными моделями. Приложение реализовано на Java с использованием библиотеки Swing для интерфейса и Java 2D API для обработки изображений. Основная цель ПО — предоставить пользователю удобный инструмент для загрузки изображений, получения их характеристик и анализа цветовых значений в различных системах представления цвета.

**Функциональные возможности программы**

Программное обеспечение обладает следующими основными функциями:

1. **Загрузка и отображение изображения**  
   Пользователь может загрузить изображение в одном из поддерживаемых форматов (JPEG, PNG, BMP, GIF, WBMP) через диалоговое окно выбора файла. После загрузки изображение отображается в окне программы с возможностью масштабирования и прокрутки. Приложение имеет автоматическую подстройку под размер окна с сохранением пропорций.
2. **Масштабирование изображения (зум)**  
   Реализована функция интерактивного увеличения и уменьшения изображения с помощью колеса мыши. Зум центрируется относительно положения курсора, что позволяет пользователю детально рассмотреть интересующие области изображения. Для изображений, превышающих размер видимой области, добавлены полосы прокрутки.
3. **Вывод информации об изображении**  
   После загрузки изображения программа предоставляет следующие данные:
   * Размер файла (в КБ)
   * Разрешение (ширина × высота в пикселях)
   * Глубина цвета (бит на пиксель)
   * Формат файла (например, PNG)
4. **Анализ цветовых характеристик пикселя**  
   Пользователь может выбрать любой пиксель изображения с помощью клика левой кнопкой мыши. Программа отображает координаты выбранного пикселя (x, y) и его цветовые значения в шести цветовых моделях:
   * RGB (0–255, например: (255, 0, 0))
   * HSV (H: 0–360°, S/V: 0–100%, например: H:0°, S:100%, V:100%)
   * HSL (H: 0–360°, S/L: 0–100%, например: H:0°, S:100%, L:50%)
   * YCbCr (Y: 0–255, Cb/Cr: -128–127, например: Y:76, Cb:85, Cr:255)
   * CMYK (0–100%, например: (0%, 100%, 100%, 0%))
   * LAB (L: 0–100, a/b: ±128, например: L:54, a:81, b:70)

Примеры работы ПО представлены на рисунках 9 и 10

1. **Обработка ошибок**  
   Программа включает механизмы обработки исключений, таких как защита от кликов вне изображения (выводится сообщение), проверка на корректность файла (поддержка формата, отсутствие повреждений), обработка исключений при вычислении цветовых моделей.

**Технические особенности**

Технические особенности программного обеспечения основаны на модульной архитектуре MVC (Model-View-Controller), обеспечивающей четкое разделение компонентов и упрощающей дальнейшую модернизацию системы. Основной класс App выполняет роль координатора, управляя взаимодействием между моделью (ImageModel), отвечающей за хранение и обработку данных изображения, представлением (ImageView), реализующим графический интерфейс с использованием библиотеки Swing, и контроллером (ImageController), который обрабатывает пользовательские действия и бизнес-логику. Для работы с изображениями используется Java 2D API, обеспечивающий высокую производительность при операциях загрузки, масштабирования и анализа пикселей. Преобразование между цветовыми моделями (RGB, CMYK, HSL, HSV, LAB, YCbCr) реализовано с применением точных математических формул, что гарантирует корректность вычислений. Механизм масштабирования изображения учитывает динамическое изменение размеров с плавной прокруткой и центрированием относительно положения курсора, что обеспечивает интуитивно понятное взаимодействие с пользователем. Обработка исключений и проверка входных данных реализованы на всех уровнях приложения, что повышает его отказоустойчивость. Кроссплатформенность обеспечивается за счет использования стандартных библиотек Java, позволяющих запускать приложение на различных операционных системах без модификации исходного кода.

**Пример использования**

Пользователь запускает приложение, нажимает кнопку «Загрузить изображение» и выбирает файл. После загрузки изображение отображается в окне с возможностью масштабирования. Клик по изображению вызывает отображение координат пикселя и его цветовых значений во всех поддерживаемых моделях. Информация об изображении обновляется автоматически при загрузке нового файла.

Изображение выглядит как примат, млекопитающее, обезьяна, снимок экрана

Контент, сгенерированный ИИ, может содержать ошибки.  
Рисунок 9- пример работы ПО(Формат BMP)Изображение выглядит как млекопитающее, примат, Земное животное, живая природа

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 10 – Пример работы ПО (Формат PNG. Демонстрация обработки нажатия за область)

* 1. **Код ПО**

https://github.com/Aleksandr1209/ImageApp.git